

Sadur Buku

Linux System Admin Guide

(Panduan Administrator Sistem Linux)

Versi 0.9

Penulis

Lars Wirzenius, Joanna Oja, Stephen Stafford, Alex Minggu

Penyadur

Haris Sandra, S.Kom

Pengantar administrasi sistem sistem Linux untuk pemula. Hak Cipta

1993—1998 Lars Wirzenius.

Hak Cipta 1998—2001 Joanna Oja. Hak

Cipta 2001—2003 Stephen Stafford.

Hak Cipta 2003—2004 Stephen Stafford & Alex Weeks.

Hak Cipta 2004—Present Alex Weeks.

Merek dagang dimiliki oleh pemiliknya.

Izin diberikan untuk menyalin, mendistribusikan dan/atau memodifikasi dokumen ini di bawah persyaratan Lisensi Dokumentasi Gratis GNU, Versi 1.2 atau versi yang lebih baru yang diterbitkan oleh Free Software Foundation; tanpa Bagian Invarian, tanpa Teks Sampul Depan, dan tanpa Teks Sampul Belakang. Salinan lisensi disertakan dalam bagian yang berjudul "Lisensi Dokumentasi Bebas GNU".

Tentang Buku Ini

"Hanya dua hal yang tak terbatas, alam semesta dan kebodohan manusia, dan saya tidak yakin tentang yang pertama." Albert Einstein

1. Ucapan Terima Kasih

1.1. Ucapan terima kasih Joanna

Banyak orang telah membantu saya dengan buku ini, secara langsung maupun tidak langsung. Saya ingin berterima kasih terutama kepada Matt Welsh atas inspirasi dan kepemimpinan LDP, Andy Oram karena membuat saya bekerja lagi dengan umpan balik yang sangat berharga, Olaf Kirch karena menunjukkan kepada saya bahwa itu bisa dilakukan, dan Adam Richter di Yggdrasil dan yang lainnya karena menunjukkan itu kepada saya. orang lain dapat mengangganya menarik juga.

Stephen Tweedie, H. Peter Anvin, Remy Card, Theodore Ts'o, dan Stephen Tweedie telah mengizinkan saya meminjam karya mereka (dan dengan demikian membuat buku terlihat lebih tebal dan lebih mengesankan): perbandingan antara sistem file xia dan ext2, perangkat list dan deskripsi sistem file ext2. Ini bukan bagian dari buku lagi. Saya sangat berterima kasih untuk ini, dan sangat menyesal untuk versi sebelumnya yang terkadang tidak memiliki atribusi yang tepat.

Selain itu, saya ingin berterima kasih kepada Mark Komarinski yang telah mengirimkan materinya pada tahun 1993 dan banyak kolom administrasi sistem di Jurnal Linux. Mereka cukup informatif dan inspiratif.

Banyak komentar bermanfaat telah dikirim oleh banyak orang. Lubang hitam mini arsip saya tidak memungkinkan saya menemukan semua nama mereka, tetapi beberapa di antaranya, dalam urutan abjad: Paul Caprioli, Ales Cepek, Marie-France Declerfayt, Dave Dobson, Olaf Flebbe, Helmut Geyer, Larry Greenfield dan ayahnya, Stephen Harris, Jyrki Havia, Jim Haynes, York Lam, Timothy Andrew Lister, Jim Lynch, Michael J. Micek, Jacob Navia, Dan Poirier, Daniel Quinlan, Jouni K Seppänen, Philippe Steindl, GB Stotte. Saya minta maaf kepada siapa pun
Saya lupa.

2. Konvensi Tipografi

Sepanjang buku ini, saya telah mencoba menggunakan konvensi tipografi yang seragam. Mudah-mudahan mereka membantu keterbacaan. Jika Anda dapat menyarankan perbaikan, silakan hubungi saya.

Nama file dinyatakan sebagai:

`/usr/share/doc/foo`. Nama perintah dinyatakan

sebagai: `fsck`

Alamat email dinyatakan sebagai: `<pengguna@domain.com>`

URL dinyatakan sebagai: `http://www.tldp.org`

Saya akan menambahkan ke bagian ini ketika hal-hal muncul saat mengedit. Jika Anda melihat sesuatu yang harus ditambahkan, beri tahu saya.

Bab 1

1. Pendahuluan

"Pada awalnya, file itu tanpa bentuk, dan kosong; dan kekosongan ada di wajah bit. Dan Jari-jari Penulis bergerak di atas permukaan keyboard. Dan Penulis berkata, Biarlah ada kata-kata, dan ada kata-kata."

Panduan Administrator Sistem Linux, menjelaskan aspek administrasi sistem dalam menggunakan Linux. Ini ditujukan untuk orang-orang yang hampir tidak tahu apa-apa tentang administrasi sistem (mereka yang mengatakan ``apa itu?"), tetapi yang telah menguasai setidaknya dasar-dasar penggunaan normal. Manual ini tidak memberi tahu Anda cara menginstal Linux; yang dijelaskan dalam dokumen Instalasi dan Persiapan. Lihat di bawah untuk informasi lebih lanjut tentang manual Linux.

Administrasi sistem mencakup semua hal yang harus Anda lakukan untuk menjaga agar sistem komputer tetap dapat digunakan. Ini mencakup hal-hal seperti mencadangkan file (dan memulihkannya jika perlu), menginstal program baru, membuat akun untuk pengguna (dan menghapusnya saat tidak lagi diperlukan), memastikan bahwa sistem file tidak rusak, dan seterusnya. Jika komputer adalah, katakanlah, sebuah rumah, administrasi sistem akan disebut pemeliharaan, dan akan mencakup pembersihan, perbaikan jendela yang pecah, dan hal-hal lain semacam itu.

Struktur manual ini sedemikian rupa sehingga banyak bab harus dapat digunakan secara independen, jadi jika Anda memerlukan informasi tentang pencadangan, misalnya, Anda dapat membaca bab itu saja. Namun, manual ini pertama dan terutama merupakan tutorial dan dapat dibaca secara berurutan atau secara keseluruhan.

Manual ini tidak dimaksudkan untuk digunakan sepenuhnya secara mandiri. Banyak dokumentasi Linux lainnya juga penting untuk administrator sistem. Bagaimanapun, administrator sistem hanyalah pengguna dengan hak dan tugas khusus. Sumber daya yang sangat berguna adalah halaman manual, yang harus selalu dikonsultasikan saat Anda tidak terbiasa dengan suatu perintah. Jika Anda tidak tahu perintah mana yang Anda butuhkan, maka perintah apropos dapat digunakan. Lihat halaman manualnya untuk lebih jelasnya.

Meskipun manual ini ditujukan untuk Linux, prinsip umumnya adalah bahwa manual ini juga harus berguna dengan sistem operasi berbasis UNIX lainnya. Sayangnya, karena ada begitu banyak perbedaan antara versi UNIX yang berbeda secara umum, dan dalam administrasi sistem pada khususnya, hanya ada sedikit harapan untuk mencakup semua varian. Bahkan mencakup semua kemungkinan untuk Linux itu sulit, karena sifat perkembangannya.

Tidak ada satu distribusi Linux resmi, sehingga orang yang berbeda memiliki pengaturan yang berbeda dan banyak orang memiliki pengaturan yang mereka buat sendiri. Buku ini tidak ditargetkan pada satu distribusi. Distribusi dapat dan memang sangat bervariasi. Bila memungkinkan, perbedaan telah dicatat dan alternatif diberikan. Untuk daftar distribusi dan beberapa perbedaannya lihat http://en.wikipedia.org/wiki/Comparison_of_Linux_distributions.

Dalam mencoba menjelaskan cara kerja sesuatu, daripada hanya mendaftar ``lima langkah mudah" untuk setiap tugas, ada banyak informasi di sini yang tidak diperlukan untuk semua orang, tetapi bagian tersebut ditandai seperti itu dan dapat dilewati jika Anda menggunakan sistem yang telah dikonfigurasi sebelumnya. Membaca semuanya akan, secara alami, meningkatkan pemahaman Anda tentang sistem dan seharusnya membuat penggunaan dan pengelolannya lebih produktif.

Pemahaman adalah kunci sukses dengan Linux. Buku ini hanya bisa memberikan resep, tapi apa yang akan Anda lakukan ketika dihadapkan pada masalah yang buku ini tidak punya resepnya? Jika buku dapat memberikan pemahaman, maka resep tidak diperlukan. Jawabannya akan terbukti dengan sendirinya.

Panduan Administrator Sistem Linux

Seperti semua pengembangan terkait Linux lainnya, pekerjaan untuk menulis manual ini dilakukan secara sukarela: Saya melakukannya karena saya pikir itu mungkin menyenangkan dan karena saya merasa itu harus dilakukan. Namun, seperti semua pekerjaan sukarela, ada batasan berapa banyak waktu, pengetahuan, dan pengalaman yang dimiliki orang. Ini berarti bahwa manual tersebut belum tentu sebagus jika seorang penyihir dibayar mahal untuk menulisnya dan telah menghabiskan ribuan tahun untuk menyempurnakannya. Diperingatkan.

Satu titik tertentu di mana sudut telah dipotong adalah bahwa banyak hal yang sudah didokumentasikan dengan baik dalam manual lain yang tersedia secara bebas tidak selalu tercakup di sini. Ini berlaku terutama untuk dokumentasi khusus program, seperti semua detail penggunaan mkfs. Hanya tujuan program dan penggunaannya yang diperlukan untuk tujuan manual ini yang dijelaskan. Untuk informasi lebih lanjut, lihat manual lain ini. Biasanya, semua dokumentasi yang dirujuk adalah bagian dari set dokumentasi Linux lengkap.

1.1. Linux atau GNU/Linux, itulah pertanyaannya.

Banyak orang merasa bahwa Linux harus benar-benar disebut GNU/Linux. Ini karena Linux hanyalah kernel, bukan aplikasi yang berjalan di atasnya. Sebagian besar utilitas baris perintah dasar ditulis oleh Free Software Foundation saat mengembangkan sistem operasi GNU mereka. Di antara utilitas tersebut adalah beberapa perintah paling dasar seperti cp, mv, ls, dan dd.

Singkatnya, yang terjadi adalah, FSF mulai mengembangkan GNU dengan menulis hal-hal seperti kompiler, pustaka C, dan utilitas baris perintah dasar sebelum kernel. Linus Torvalds, memulai Linux dengan menulis kernel Linux terlebih dahulu dan menggunakan aplikasi yang ditulis untuk GNU.

Saya tidak merasa bahwa ini adalah forum yang tepat untuk memperdebatkan nama apa yang harus digunakan orang ketika merujuk ke Linux. Saya menyebutkannya di sini, karena saya merasa penting untuk memahami hubungan antara GNU dan Linux, dan juga menjelaskan mengapa beberapa Linux kadang-kadang disebut sebagai GNU/Linux. Dokumen tersebut hanya akan menyebutnya sebagai Linux.

Sisi GNU dari masalah ini dibahas di situs web mereka:

hubungan <http://www.gnu.org/gnu/linux-and-gnu.html>

Mengapa Linux harus GNU/Linux <http://www.gnu.org/gnu/why-gnu-linux.html>

FAQ GNU/Linux <http://www.gnu.org/gnu/gnu-linux-faq.html>

Berikut adalah beberapa tampilan

alternatif: <http://librenix.com/?inode=2312>

<http://www.topology.org/linux/lingl.html>

<http://atulchitnis.net/writings/gnulinix.php>

1.2. Merek Dagang

Microsoft, Windows, Windows NT, Windows 2000, dan Windows XP adalah merek dagang dan/atau merek dagang terdaftar dari Microsoft Corporation.

Panduan Administrator Sistem Linux

Red Hat adalah merek dagang Red Hat, Inc., di Amerika Serikat dan negara lain.

SuSE adalah merek dagang dari Novell.

Linux adalah merek dagang terdaftar dari Linus Torvalds.

UNIX adalah merek dagang terdaftar di Amerika Serikat dan negara lain, dilisensikan secara eksklusif melalui X/Open
Perusahaan Ltd.

GNU adalah merek dagang terdaftar dari Free Software Foundation.

Nama produk lain yang disebutkan di sini mungkin merupakan merek dagang dan/atau merek dagang terdaftar dari masing-masing perusahaan.

Bab 2.

2. Ikhtisar Sistem Linux

"Tuhan melihat segala sesuatu yang telah dia buat, dan melihat bahwa itu sangat baik." Bible King James

Versi: kapan. Kejadian 1:31

Bab ini memberikan gambaran umum tentang sistem Linux. Pertama, layanan utama yang disediakan oleh sistem operasi dijelaskan. Kemudian, program-program yang mengimplementasikan layanan ini dijelaskan dengan sangat kurang detail. Tujuan dari bab ini adalah untuk memberikan pemahaman tentang sistem secara keseluruhan, sehingga setiap bagian dijelaskan secara rinci di tempat lain.

2.1. Berbagai bagian dari sistem operasi

UNIX dan sistem operasi 'seperti UNIX' (seperti Linux) terdiri dari kernel dan beberapa program sistem. Ada juga beberapa program aplikasi untuk melakukan pekerjaan. Kernel adalah jantung dari sistem operasi. Bahkan, sering keliru dianggap sebagai sistem operasi itu sendiri, padahal tidak. Sebuah sistem operasi menyediakan lebih banyak layanan daripada kernel biasa.

Itu melacak file pada disk, memulai program dan menjalankannya secara bersamaan, memberikan memori dan sumber daya lainnya ke berbagai proses, menerima paket dari dan mengirim paket ke jaringan, dan seterusnya. Kernel melakukan sangat sedikit dengan sendirinya, tetapi menyediakan alat yang dengannya semua layanan dapat dibangun. Ini juga mencegah siapa pun mengakses perangkat keras secara langsung, memaksa semua orang untuk menggunakan alat yang disediakan. Dengan cara ini kernel memberikan perlindungan bagi pengguna satu sama lain. Alat yang disediakan oleh kernel digunakan melalui panggilan sistem. Lihat halaman manual bagian 2 untuk informasi lebih lanjut tentang ini.

Program sistem menggunakan alat yang disediakan oleh kernel untuk mengimplementasikan berbagai layanan yang diperlukan dari sistem operasi. Program sistem, dan semua program lainnya, dijalankan `di atas kernel', dalam apa yang disebut mode pengguna. Perbedaan antara sistem dan program aplikasi adalah salah satu tujuannya: aplikasi dimaksudkan untuk menyelesaikan hal-hal yang berguna (atau untuk bermain, jika itu adalah game), sedangkan program sistem adalah diperlukan untuk membuat sistem bekerja. Pengolah kata adalah sebuah aplikasi; mount adalah program sistem. Perbedaannya sering agak kabur, bagaimanapun, dan penting hanya untuk pengkategorian kompulsif.

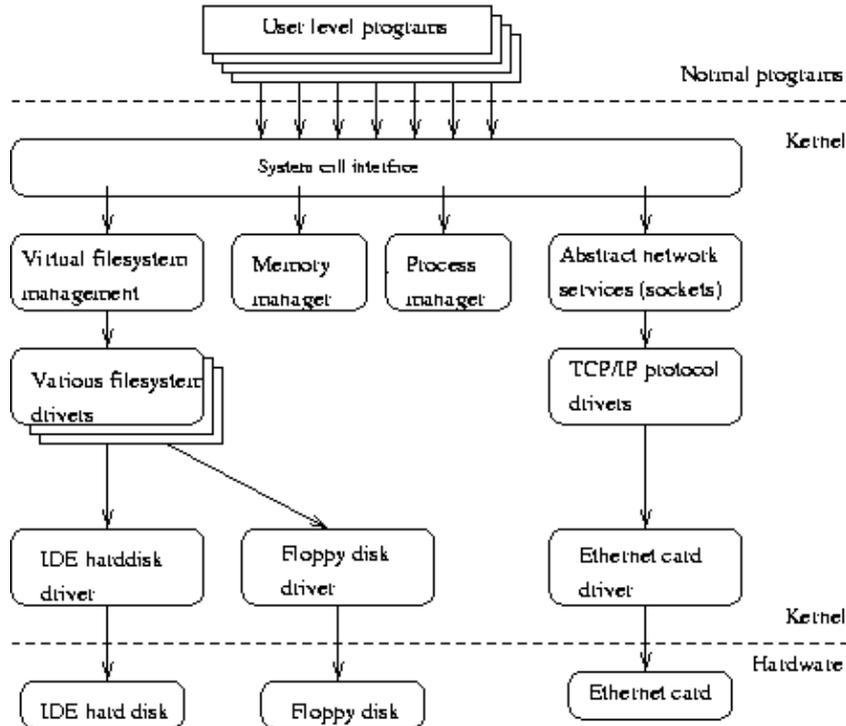
Sistem operasi juga dapat berisi kompiler dan pustaka yang sesuai (GCC dan pustaka C khususnya di Linux), meskipun tidak semua bahasa pemrograman harus menjadi bagian dari sistem operasi. Dokumentasi, dan terkadang bahkan game, juga bisa menjadi bagian darinya. Secara tradisional, sistem operasi telah ditentukan oleh isi pita atau disk instalasi; dengan Linux tidak begitu jelas karena tersebar di seluruh situs FTP dunia.

2.2. Bagian penting dari kernel

Kernel Linux terdiri dari beberapa bagian penting: manajemen proses, manajemen memori, driver perangkat keras, driver sistem file, manajemen jaringan, dan berbagai bagian lainnya. Gambar [2-1](#) menunjukkan beberapa di antaranya.

Gambar 2-1. Beberapa bagian yang lebih penting dari kernel Linux

Panduan Administrator Sistem Linux



Mungkin bagian terpenting dari kernel (tidak ada yang berfungsi tanpanya) adalah manajemen memori dan manajemen proses. Manajemen memori menangani penetapan area memori dan area ruang swap ke proses, bagian dari kernel, dan untuk cache buffer. Manajemen proses menciptakan proses, dan mengimplementasikan multitasking dengan mengalihkan proses aktif pada prosesor.

Pada level terendah, kernel berisi driver perangkat keras untuk setiap jenis perangkat keras yang didukungnya. Karena dunia ini penuh dengan berbagai jenis perangkat keras, jumlah driver perangkat keras sangat banyak. Seringkali ada banyak perangkat keras serupa yang berbeda dalam cara mereka dikendalikan oleh perangkat lunak. Itu

kesamaan memungkinkan untuk memiliki kelas umum driver yang mendukung operasi serupa; setiap anggota kelas memiliki antarmuka yang sama dengan kernel lainnya tetapi berbeda dalam hal apa yang perlu dilakukan untuk mengimplementasikannya. Sebagai contoh, semua driver disk terlihat sama dengan kernel lainnya, yaitu, mereka semua memiliki operasi seperti `inisialisasi drive`, `read sector N`, dan `write sector N`.

Beberapa layanan perangkat lunak yang disediakan oleh kernel itu sendiri memiliki sifat yang serupa, dan oleh karena itu dapat diabstraksikan ke dalam kelas-kelas. Misalnya, berbagai protokol jaringan telah diabstraksikan ke dalam satu antarmuka pemrograman, perpustakaan socket BSD. Contoh lain adalah lapisan sistem file virtual (VFS) yang mengabstraksikan operasi sistem file dari implementasinya. Setiap jenis sistem file menyediakan implementasi dari setiap operasi sistem file. Ketika beberapa entitas mencoba menggunakan sistem file, permintaan berjalan melalui VFS, yang mengarahkan permintaan ke driver sistem file yang tepat.

Diskusi yang lebih mendalam tentang internal kernel dapat ditemukan di <http://www.tldp.org/LDP/lki/index.html>. Dokumen ini ditulis untuk kernel 2.4. Ketika saya menemukan satu untuk kernel 2.6, saya akan mencantulkannya di sini.

2.3. Layanan utama dalam sistem UNIX

Panduan Administrator Sistem Linux

Bagian ini menjelaskan beberapa layanan UNIX yang lebih penting, tetapi tanpa banyak detail. Mereka dijelaskan lebih teliti dalam bab-bab selanjutnya.

2.3.1. **init**

Layanan yang paling penting dalam sistem UNIX disediakan oleh `init` yang dimulai sebagai proses pertama dari setiap sistem UNIX, sebagai hal terakhir yang dilakukan kernel saat boot. Ketika `init` dimulai, ia melanjutkan proses boot dengan melakukan berbagai tugas startup (memeriksa dan memasang sistem file, memulai daemon, dll).

Daftar pasti dari hal-hal yang dilakukan `init` bergantung pada rasa apa itu; ada beberapa untuk dipilih. `init` biasanya menyediakan konsep mode pengguna tunggal, di mana tidak ada yang bisa masuk dan root menggunakan shell di konsol; mode biasa disebut mode multiuser. Beberapa rasa menggeneralisasi ini sebagai run level; mode single dan multiuser dianggap sebagai dua run level, dan bisa juga ada tambahan, misalnya, untuk menjalankan X di konsol.

Linux memungkinkan hingga 10 runlevel, 0–9, tetapi biasanya hanya beberapa yang ditentukan secara default. Runlevel 0 didefinisikan sebagai "system berhenti". Runlevel 1 didefinisikan sebagai "mode pengguna tunggal". Runlevel 3 didefinisikan sebagai "multi-pengguna" karena merupakan runlevel tempat sistem melakukan booting dalam kondisi normal sehari-hari. Runlevel 5 biasanya sama dengan 3 kecuali bahwa GUI juga dimulai. Runlevel 6 didefinisikan sebagai "reboot sistem". Runlevel lain bergantung pada bagaimana distribusi tertentu Anda mendefinisikannya, dan mereka sangat bervariasi antar distribusi. Melihat isi dari `/etc/inittab` biasanya akan memberikan beberapa petunjuk tentang runlevel yang telah ditentukan sebelumnya dan definisinya sebagai apa.

Dalam operasi normal, `init` memastikan `getty` berfungsi (untuk mengizinkan pengguna masuk) dan untuk mengadopsi proses yatim (proses yang induknya telah mati; di UNIX semua proses harus berada dalam satu pohon, jadi anak yatim harus diadopsi).

Ketika sistem dimatikan, `init` yang bertanggung jawab untuk mematikan semua proses lain, melepas semua sistem file dan menghentikan prosesor, bersama dengan hal lain yang telah dikonfigurasi untuk dilakukan.

2.3.2. **Login dari terminal**

Login dari terminal (melalui jalur serial) dan konsol (bila tidak menjalankan X) disediakan oleh program `getty`. `init` memulai instance `getty` terpisah untuk setiap terminal tempat login diizinkan. `getty` membaca nama pengguna dan menjalankan program login, yang membaca kata sandi. Jika nama pengguna dan kata sandi benar, login menjalankan shell. Ketika shell berakhir, yaitu, pengguna logout, atau saat login dihentikan karena nama pengguna dan kata sandi tidak cocok, `init` memperhatikan hal ini dan memulai instance `getty` baru. Kernel tidak memiliki gagasan tentang login, ini semua ditangani oleh program sistem.

2.3.3. **syslog**

Kernel dan banyak program sistem menghasilkan kesalahan, peringatan, dan pesan lainnya. Seringkali penting bahwa pesan-pesan ini dapat dilihat kemudian, bahkan jauh di kemudian hari, sehingga mereka harus ditulis ke sebuah file. Program yang melakukan ini adalah `syslog`. Itu dapat dikonfigurasi untuk mengurutkan pesan ke file yang berbeda menurut penulis atau tingkat kepentingannya. Misalnya, pesan kernel sering diarahkan ke file terpisah dari yang lain, karena pesan kernel seringkali lebih penting dan perlu dibaca secara teratur untuk menemukan masalah.

Bab 15 akan memberikan lebih banyak tentang ini.

2.3.4. **Eksekusi perintah berkala: cron dan at**

Panduan Administrator Sistem Linux

Baik pengguna dan administrator sistem sering kali perlu menjalankan perintah secara berkala. Misalnya, administrator sistem mungkin ingin menjalankan perintah untuk membersihkan folder dengan file sementara (`/tmp` dan `/var/tmp`) dari file lama, agar disk tidak terisi, karena tidak semua program membersihkan sendiri benar.

Layanan cron diatur untuk melakukan ini. Setiap pengguna dapat memiliki `crontab` file, di mana dia mencantumkan perintah yang ingin dia jalankan dan waktu mereka harus dieksekusi. Daemon cron menanggapi memulai perintah saat ditentukan.

Layanan `at` mirip dengan cron, tetapi hanya sekali: perintah dieksekusi pada waktu tertentu, tetapi tidak diulang.

Kami akan membahas ini lebih lanjut nanti. Lihat halaman manual `cron(1)`, `crontab(1)`, `crontab(5)`, `at(1)` dan `atd(8)` untuk informasi lebih mendalam.

Bab 13 akan membahas ini.

2.3.5. Antarmuka pengguna grafis

UNIX dan Linux tidak memasukkan antarmuka pengguna ke dalam kernel; sebaliknya, mereka membiarkannya diimplementasikan oleh program tingkat pengguna. Ini berlaku untuk mode teks dan lingkungan grafis.

Pengaturan ini membuat sistem lebih fleksibel, tetapi memiliki kelemahan yaitu sederhana untuk mengimplementasikan antarmuka pengguna yang berbeda untuk setiap program, membuat sistem lebih sulit untuk dipelajari.

Lingkungan grafis yang terutama digunakan dengan Linux disebut X Window System (disingkat X). X juga tidak mengimplementasikan antarmuka pengguna; itu hanya mengimplementasikan sistem jendela, yaitu, alat yang dengannya antarmuka pengguna grafis dapat diimplementasikan. Beberapa pengelola jendela yang populer adalah: `fvwm`, `icewm`, `blackbox`, dan `windowmaker`. Ada juga dua manajer desktop populer, KDE dan GNOME.

2.3.6. Jaringan

Jaringan adalah tindakan menghubungkan dua atau lebih komputer sehingga mereka dapat berkomunikasi satu sama lain. Metode sebenarnya untuk menghubungkan dan berkomunikasi sedikit rumit, tetapi hasil akhirnya sangat berguna.

Sistem operasi UNIX memiliki banyak fitur jaringan. Sebagian besar layanan dasar (sistem file, pencetakan, pencadangan, dll) dapat dilakukan melalui jaringan. Hal ini dapat membuat administrasi sistem lebih mudah, karena memungkinkan administrasi terpusat, sambil tetap menuai manfaat komputasi mikro dan komputasi terdistribusi, seperti biaya yang lebih rendah dan toleransi kesalahan yang lebih baik.

Namun, buku ini hanya melirik jaringan; lihat Panduan Administrator Jaringan Linux <http://www.tldp.org/LDP/nag2/index.html> untuk informasi lebih lanjut, termasuk deskripsi dasar tentang bagaimana jaringan beroperasi.

2.3.7. Login jaringan

Login jaringan bekerja sedikit berbeda dari login normal. Untuk setiap orang yang masuk melalui jaringan ada koneksi jaringan virtual yang terpisah, dan jumlahnya bisa berapa saja tergantung pada bandwidth yang tersedia. Oleh karena itu, tidak mungkin menjalankan `getty` terpisah untuk setiap kemungkinan koneksi

Panduan Administrator Sistem Linux

virtual. Ada juga beberapa cara berbeda untuk masuk melalui jaringan, telnet dan ssh menjadi yang utama di jaringan TCP/IP.

Saat ini banyak administrator sistem Linux menganggap telnet dan rlogin tidak aman dan lebih memilih ssh, "secure shell", yang mengenkripsi lalu lintas yang melewati jaringan, sehingga memperkecil kemungkinan bahwa si jahat dapat "mengendus" koneksi Anda dan mendapatkan data sensitif seperti nama pengguna dan kata sandi. Sangat disarankan Anda menggunakan ssh daripada telnet atau rlogin.

Login jaringan memiliki, alih-alih sekumpulan getty, satu daemon per cara login (telnet dan ssh memiliki daemon terpisah) yang mendengarkan semua upaya login yang masuk. Ketika ia menyadarinya, ia memulai contoh baru dari dirinya sendiri untuk menangani upaya tunggal itu; instance asli terus mendengarkan upaya lain. Instance baru bekerja mirip dengan getty.

2.3.8. Sistem file jaringan

Salah satu hal bermanfaat yang dapat dilakukan dengan layanan jaringan adalah berbagi file melalui sistem file jaringan. Tergantung pada jaringan Anda, ini dapat dilakukan melalui Network File System (NFS), atau melalui Common Internet File System (CIFS). NFS biasanya merupakan layanan berbasis 'UNIX'. Di Linux, NFS didukung oleh kernel. Namun CIFS tidak. Di Linux, CIFS didukung oleh Samba <http://www.samba.org>.

Dengan sistem file jaringan, setiap operasi file yang dilakukan oleh program pada satu mesin dikirim melalui jaringan ke komputer lain. Ini membodohi program untuk berpikir bahwa semua file di komputer lain sebenarnya ada di komputer tempat program berjalan. Hal ini membuat berbagi informasi menjadi sangat sederhana, karena tidak memerlukan modifikasi pada program.

Ini akan dibahas secara lebih rinci dalam [Bagian 5.4](#).

2.3.9. Surat

Surat elektronik adalah metode yang paling populer digunakan untuk berkomunikasi melalui komputer. Surat elektronik disimpan dalam file menggunakan format khusus, dan program surat khusus digunakan untuk mengirim dan membaca surat.

Setiap pengguna memiliki kotak surat masuk (file dalam format khusus), tempat semua surat baru disimpan. Ketika seseorang mengirim surat, program surat menempatkan kotak surat penerima dan menambahkan surat itu ke file kotak surat. Jika kotak surat penerima berada di mesin lain, surat itu dikirim ke mesin lain, yang mengirimkannya ke kotak surat yang dianggap paling sesuai.

Sistem surat terdiri dari banyak program. Pengiriman surat ke kotak surat lokal atau remote dilakukan oleh satu program (mail transfer agent (MTA) , misalnya sendmail atau postfix), sedangkan program yang digunakan pengguna banyak dan beragam (mail user agent (MUA) , misalnya pinus , atau evolusi Kotak surat biasanya disimpan di

`/var/spool/mail` sampai MUA pengguna mengambilnya.

Untuk informasi lebih lanjut tentang mengatur dan menjalankan layanan email, Anda dapat membaca HOWTO Administrator Surat di <http://www.tldp.org/HOWTO/Mail-Administrator-HOWTO.html>, atau kunjungi situs web sendmail atau postfix. <http://www.sendmail.org/>, atau <http://www.postfix.org/> .

2.3.10. Pencetakan

Hanya satu orang yang dapat menggunakan printer pada satu waktu, tetapi tidak ekonomis untuk tidak berbagi printer di antara pengguna. Oleh karena itu, printer dikelola oleh perangkat lunak yang menerapkan antrian cetak: semua pekerjaan cetak dimasukkan ke dalam antrian dan setiap kali printer selesai dengan satu pekerjaan, pekerjaan berikutnya dikirim ke sana secara otomatis. Ini membebaskan pengguna dari mengatur

Panduan Administrator Sistem Linux

antrian cetak dan memperebutkan kendali printer. Sebaliknya, mereka membentuk antrian baru di printer, menunggu hasil cetakan mereka, karena sepertinya tidak ada yang pernah bisa mendapatkan perangkat lunak antrian untuk mengetahui secara pasti kapan hasil cetakan seseorang benar-benar selesai. Ini adalah dorongan besar untuk hubungan sosial intra-kantor.

Perangkat lunak antrian cetak juga menggulung hasil cetak pada disk, yaitu, teks disimpan dalam file saat pekerjaan dalam antrian. Hal ini memungkinkan program aplikasi untuk mengeluarkan pekerjaan cetak dengan cepat ke perangkat lunak antrian cetak; aplikasi tidak harus menunggu sampai pekerjaan benar-benar dicetak untuk melanjutkan. Ini sangat nyaman, karena memungkinkan seseorang untuk mencetak satu versi, dan tidak perlu menunggu untuk dicetak sebelum dapat membuat versi baru yang sepenuhnya direvisi.

Anda dapat merujuk ke Pencetakan-HOWTO yang terletak di <http://www.tldp.org/HOWTO/Printing-HOWTO/index.html> untuk bantuan lebih lanjut dalam menyiapkan printer.

2.3.11. Tata letak sistem file

Sistem file dibagi menjadi banyak bagian; biasanya di sepanjang baris sistem file root dengan `/bin` , `/lib` , `/etc` , `/dev` , dan beberapa lainnya; sebuah `/usr` sistem file dengan program dan data yang tidak berubah; `/var` berkas sistem dengan mengubah data (seperti file log); dan `/home` untuk file pribadi semua orang. Tergantung pada perangkat kerasnya konfigurasi dan keputusan administrator sistem, divisi dapat berbeda; bahkan bisa all in satu sistem file.

Bab 3 menjelaskan tata letak sistem file dalam beberapa detail kecil; Standar Hirarki Sistem File . mencakupnya secara lebih rinci. Ini dapat ditemukan di web di: <http://www.pathname.com/fhs/>

Bab 3.

3. Ikhtisar Pohon Folder

"Dua hari kemudian, ada Pooh, duduk di dahan, menjuntai kakinya, dan di sana, di sampingnya, ada empat pot madu..." (AA Milne)

Bab ini menjelaskan bagian-bagian penting dari pohon folder Linux standar, berdasarkan Standar Hirarki Sistem File. Ini menguraikan cara normal memecah pohon folder menjadi sistem file terpisah dengan tujuan berbeda dan memberikan motivasi di balik pemisahan khusus ini. Tidak semua distribusi Linux mengikuti standar ini dengan rendah hati, tetapi cukup umum untuk memberi Anda gambaran umum.

3.1. Latar belakang

Bab ini secara longgar didasarkan pada Filesystems Hierarchy Standard (FHS), versi 2.1, yang mencoba menetapkan standar tentang bagaimana pohon folder di sistem Linux diatur. Standar seperti itu memiliki keuntungan bahwa akan lebih mudah untuk menulis atau port perangkat lunak untuk Linux, dan untuk mengelola mesin Linux, karena semuanya harus

berada di tempat standar. Tidak ada otoritas di balik standar yang memaksa siapa pun untuk mematuhi, tetapi telah mendapat dukungan dari banyak distribusi Linux. Bukan ide yang baik untuk memutuskan hubungan dengan FHS tanpa alasan yang sangat kuat. FHS mencoba untuk mengikuti tradisi Unix dan tren saat ini, membuat sistem Linux akrab bagi mereka yang berpengalaman dengan sistem Unix lainnya, dan sebaliknya.

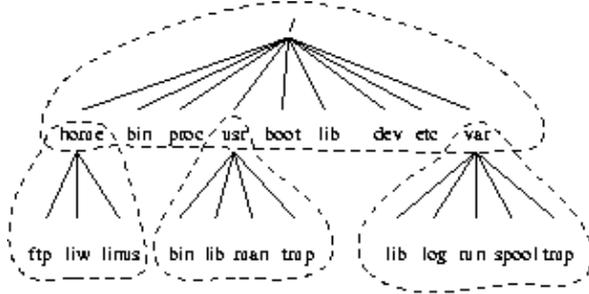
Bab ini tidak sedetail FHS. Seorang administrator sistem juga harus membaca FHS lengkap untuk pemahaman yang lengkap.

Bab ini tidak menjelaskan semua file secara rinci. Tujuannya bukan untuk menggambarkan setiap file, tetapi untuk memberikan gambaran umum tentang sistem dari sudut pandang sistem file. Informasi lebih lanjut tentang setiap file tersedia di tempat lain dalam manual ini atau di halaman manual Linux.

Pohon folder lengkap dimaksudkan untuk dapat dipecah menjadi bagian-bagian yang lebih kecil, masing-masing mampu berada di disk atau partisinya sendiri, untuk mengakomodasi batas ukuran disk dan untuk memudahkan pencadangan dan tugas administrasi sistem lainnya. Bagian utama adalah akar (`/`), `/usr`, `/var`, dan `/home` sistem file (lihat Gambar 3-1). Setiap bagian memiliki tujuan yang berbeda. Pohon folder telah dirancang sedemikian rupa sehingga bekerja dengan baik di jaringan mesin Linux yang mungkin berbagi beberapa bagian dari sistem file melalui perangkat read-only (misalnya, CD-ROM), atau melalui jaringan dengan NFS.

Gambar 3-1. Bagian dari pohon folder Unix. Garis putus-putus menunjukkan batas partisi.

Panduan Administrator Sistem Linux



Peran dari bagian yang berbeda dari pohon folder dijelaskan di bawah ini.

- Sistem file root khusus untuk setiap mesin (biasanya disimpan di disk lokal, meskipun bisa juga ramdisk atau drive jaringan) dan berisi file yang diperlukan untuk mem-boot sistem, dan membawanya ke keadaan sedemikian rupa sehingga sistem file lain dapat di-mount. Oleh karena itu, isi dari sistem file root akan cukup untuk status pengguna tunggal. Ini juga akan berisi alat untuk memperbaiki sistem yang rusak, dan untuk memulihkan file yang hilang dari cadangan.
- /usr filesystem berisi semua perintah, perpustakaan, halaman manual, dan file lain yang tidak berubah yang diperlukan selama operasi normal. Tidak ada file di /usr harus spesifik untuk mesin tertentu, juga tidak boleh dimodifikasi selama penggunaan normal. Hal ini memungkinkan file untuk dibagikan melalui jaringan, yang dapat menghemat biaya karena menghemat ruang disk (dapat dengan mudah ratusan megabyte, semakin banyak gigabyte dalam /usr). Itu dapat membuat administrasi lebih mudah (hanya master /usr perlu diubah saat memperbarui aplikasi, tidak setiap mesin secara terpisah) untuk memiliki jaringan /usr dipasang. Bahkan jika sistem file berada di disk lokal, itu dapat dipasang hanya-baca, untuk mengurangi kemungkinan kerusakan sistem file selama crash.
- /var filesystem berisi file yang berubah, seperti folder spool (untuk surat, berita, printer, dll), file log, halaman manual yang diformat, dan file sementara. Secara tradisional segala sesuatu di /var telah berada di suatu tempat di bawah /usr, tapi itu membuatnya tidak mungkin untuk dipasang /usr baca saja.
- /home filesystem berisi folder home pengguna, yaitu, semua data nyata pada sistem. Memisahkan folder home ke pohon folder atau sistem file mereka sendiri membuat pencadangan lebih mudah; yang lain bagian sering tidak harus dicadangkan, atau setidaknya tidak sesering mereka jarang berubah. Besar /home mungkin harus dipecah di beberapa sistem file, yang memerlukan penambahan level penamaan tambahan di bawah /home, Misalnya /home /siswadan /home /staf.

Meskipun bagian-bagian yang berbeda telah disebut sistem file di atas, tidak ada persyaratan bahwa mereka benar-benar berada di sistem file yang terpisah. Mereka dapat dengan mudah disimpan dalam satu sistem jika sistemnya adalah sistem pengguna tunggal yang kecil dan pengguna ingin menjaga semuanya tetap sederhana. Pohon folder juga dapat dibagi menjadi sistem file secara berbeda, tergantung pada seberapa besar disk, dan bagaimana ruang dialokasikan untuk berbagai tujuan. Namun, bagian yang penting adalah bahwa semua nama standar berfungsi; bahkan jika, katakanlah, /var dan /usr sebenarnya berada di partisi yang sama, namanya /usr/lib/libc.adan /var/log/pesan harus bekerja, misalnya dengan memindahkan file di bawah /var ke dalam /usr/var, dan membuat /var sebuah symlink ke /usr/var.

Struktur sistem file Unix mengelompokkan file menurut tujuannya, yaitu, semua perintah ada di satu tempat, semua file data di tempat lain, dokumentasi di tempat ketiga, dan seterusnya. Alternatifnya adalah mengelompokkan file file menurut programnya, yaitu, semua file Emacs akan berada di satu folder, semua TeX di folder lain, dan seterusnya. Masalah dengan pendekatan yang terakhir adalah menyulitkan untuk berbagi file (folder program sering berisi file statis dan dapat dibagikan dan berubah dan tidak dapat dibagikan), dan kadang-kadang bahkan untuk menemukan file (misalnya, halaman manual dalam jumlah besar tempat, dan membuat program halaman manual menemukan semuanya adalah mimpi buruk pemeliharaan).

3.2. Sistem file root

Sistem file root umumnya harus kecil, karena berisi file yang sangat penting dan sistem file kecil yang jarang dimodifikasi memiliki peluang lebih baik untuk tidak rusak. Sistem berkas root yang rusak biasanya berarti bahwa sistem menjadi tidak dapat di-boot kecuali dengan tindakan khusus (misalnya, dari floppy), jadi Anda tidak ingin mengambil risiko.

Folder root umumnya tidak berisi file apa pun, kecuali mungkin pada sistem lama di mana image boot standar untuk sistem, biasanya disebut `/vmlinuz` disimpan di sana. (Sebagian besar distribusi telah memindahkan file-file itu ke `/boot` folder. Jika tidak, semua file disimpan di subfolder di bawah sistem file root:

/bin

Perintah yang diperlukan selama boot yang mungkin digunakan oleh pengguna biasa (mungkin setelah `/sbin` boot).

Suka `/bin`, tetapi perintah tersebut tidak ditujukan untuk pengguna biasa, meskipun mereka dapat menggunakannya jika perlu dan diizinkan. `/sbin` biasanya tidak berada di jalur default pengguna normal, tetapi akan berada di jalur default root.

/akar File konfigurasi khusus untuk mesin.

/lib Folder home untuk root pengguna. Ini biasanya tidak dapat diakses oleh pengguna lain di sistem

Pustaka bersama yang dibutuhkan oleh program pada sistem file root.

/lib/modul

Modul kernel yang dapat dimuat, terutama yang diperlukan untuk mem-boot sistem saat memulihkan dari bencana (misalnya, driver jaringan dan sistem file).

/dev

File perangkat. Ini adalah file khusus yang membantu antarmuka pengguna dengan berbagai perangkat

/tmp

di sistem. Berkas sementara. Seperti namanya, program yang berjalan sering menyimpan file-file

/boot

sementara di sini.

File yang digunakan oleh bootstrap loader, misalnya LILO atau GRUB. Gambar kernel sering disimpan di sini alih-alih di folder root. Jika ada banyak gambar kernel, folder dapat dengan mudah bertambah besar, dan mungkin lebih baik menyimpannya di sistem file yang terpisah. Alasan lain adalah untuk memastikan gambar kernel berada dalam 1024 silinder pertama dari disk IDE. Batas 1024 silinder ini tidak lagi berlaku dalam banyak kasus. Dengan BIOS modern dan versi LILO (Linux LOader) yang lebih baru, batas silinder 1024 dapat dilewati dengan pengalamatan blok logis (LBA). Lihat halaman manual lilo untuk lebih jelasnya.

/mnt

Titik pemasangan untuk pemasangan sementara oleh administrator sistem. Program tidak seharusnya dipasang/mntsecara otomatis./mntdapat dibagi menjadi subfolder (misalnya,/mnt/dosamungkin floppy drive menggunakan sistem file MS-DOS, dan/mnt/tambahanmungkin sama dengan sistem file ext2).

/proc, /usr, /var, /home

Mount poin untuk sistem file lainnya. Meskipun `/proc` tidak berada pada disk apa pun pada kenyataannya itu masih disebutkan di sini. Lihat bagian tentang `/proc` nanti di bab.

3.3. /etc folder

`/etc` menyimpan banyak file. Beberapa di antaranya dijelaskan di bawah ini. Untuk yang lain, Anda harus menentukan program mana yang mereka ikuti dan membaca halaman manual untuk program itu. Banyak file konfigurasi jaringan adalah di `/etc` juga, dan dijelaskan dalam Panduan Administrator Jaringan.

/etc/rcatau /etc/rc.d atau /etc/rc?.d

Skrip atau folder skrip untuk dijalankan saat startup atau saat mengubah level run. [Lihat Bagian 2.3.1](#) untuk informasi lebih lanjut.

/etc/passwd

Basis data pengguna, dengan bidang yang memberikan nama pengguna, nama asli, folder home, dan informasi lainnya tentang setiap pengguna. Formatnya didokumentasikan di halaman manual `passwd`.

/etc/bayangan

`/etc/bayangan` adalah file terenkripsi yang menyimpan kata sandi pengguna.

/etc/fdprm

Tabel parameter disket. Menjelaskan seperti apa format floppy disk yang berbeda. Digunakan oleh `setfdprm`. Lihat halaman manual `setfdprm` untuk informasi lebih lanjut.

/etc/fstab

Mencantumkan sistem file yang dipasang secara otomatis saat startup dengan perintah `mount a` (dalam `/etc/rcatau` file startup yang setara). Di Linux, juga berisi informasi tentang area swap yang digunakan secara otomatis oleh `swapon a`. [Lihat Bagian 5.10.7](#) dan halaman panduan pemasangan untuk informasi lebih lanjut. Juga `fstab` biasanya memiliki halaman manual sendiri di bagian 5.

/etc/grup

Mirip dengan `/etc/passwd`, tetapi menjelaskan grup, bukan pengguna. Lihat [kelompok](#) halaman manual di bagian 5 untuk informasi lebih lanjut.

/etc/inittab

File konfigurasi untuk `init`.

/etc/isu

Output oleh `getty` sebelum prompt login. Biasanya berisi deskripsi singkat atau pesan sambutan ke sistem. Isinya terserah administrator sistem.

/etc/magic

File konfigurasi untuk file. Berisi deskripsi berbagai format file berdasarkan file mana menebak jenis file. Lihat [sihir](#) dan file halaman manual untuk informasi lebih lanjut.

/etc/motd

Pesan hari ini, secara otomatis keluar setelah login berhasil. Konten terserah sistem administrator. Sering digunakan untuk mendapatkan informasi kepada setiap pengguna, seperti peringatan tentang waktu henti yang direncanakan.

/etc/mtab

Daftar sistem file yang saat ini dipasang. Awalnya diatur oleh skrip boot, dan diperbarui secara otomatis dengan perintah `mount`. Digunakan ketika daftar sistem file yang di-mount diperlukan, misalnya, dengan perintah `df`.

/etc/login.defs

Panduan Administrator Sistem Linux

File konfigurasi untuk perintah login. `login.defs` file biasanya memiliki halaman manual di bagian 5.

/etc/printcap

Suka `/etc/termcap` `/etc/printcap`, tetapi ditujukan untuk printer. Namun menggunakan berbeda

sintaxis. `Itucap` cetak memiliki halaman manual di bagian 5.

/etc/profil, /etc/bash.rc, /etc/csh.cshrc

File dieksekusi saat login atau waktu startup oleh shell Bourne, BASH, atau C. Ini memungkinkan sistem

administrator untuk mengatur default global untuk semua pengguna. Pengguna juga dapat membuat salinan individual ini di folder home mereka untuk mempersonalisasi lingkungan mereka. Lihat halaman manual untuk masing-masing cangkang.

/etc/securetty

Mengidentifikasi terminal aman, yaitu terminal dari mana root diizinkan untuk masuk. Biasanya hanya konsol virtual terdaftar, sehingga menjadi tidak mungkin (atau setidaknya lebih sulit) untuk mendapatkan hak pengguna super dengan membobol sistem melalui modem atau jaringan. Jangan izinkan login root melalui jaringan. Lebih suka masuk sebagai pengguna yang tidak memiliki hak istimewa dan gunakan `su` atau `sudo` untuk mendapatkan hak akses root.

/etc/shells

Mencantumkan cangkang tepercaya. Perintah `chsh` memungkinkan pengguna untuk mengubah shell login mereka hanya ke shell yang terdaftar di

file ini. `ftpd`, adalah proses server yang menyediakan layanan FTP untuk mesin, akan memeriksa apakah shell pengguna terdaftar di `/etc/shells` dan tidak akan membiarkan orang masuk kecuali Shell terdaftar di sana.

/etc/termcap

Basis data kemampuan terminal. Menjelaskan dengan "urutan pelarian" berbagai terminal dapat dikendalikan. Program ditulis sehingga alih-alih secara langsung mengeluarkan urutan pelarian yang hanya berfungsi pada merek terminal tertentu, mereka mencari urutan yang benar untuk melakukan apa pun yang ingin mereka lakukan. Akibatnya sebagian besar program bekerja dengan sebagian besar jenis terminal. Lihat `termcap`, `curs_termcap`, dan istilah halaman manual untuk informasi lebih lanjut.

3.4. /dev folder

`/dev` folder berisi file perangkat khusus untuk semua perangkat. File perangkat dibuat selama instalasi, dan kemudian dengan skrip `/dev/MAKEDEV`. `/dev/MAKEDEV.local` adalah skrip yang ditulis oleh administrator sistem yang membuat file atau tautan perangkat lokal saja (yaitu yang bukan bagian dari `MAKEDEV` standar, seperti file perangkat untuk beberapa driver perangkat non-standar).

Daftar berikut ini sama sekali tidak lengkap atau sedetail mungkin. Banyak dari file perangkat ini akan memerlukan dukungan yang dikompilasi ke dalam kernel Anda untuk perangkat keras. Baca dokumentasi kernel untuk menemukan detail perangkat tertentu.

Jika menurut Anda ada perangkat lain yang harus disertakan di sini tetapi tidak, beri tahu saya. Saya akan mencoba memasukkannya ke dalam revisi berikutnya.

/dev/dsp

Prosesor Sinyal Digital. Pada dasarnya ini membentuk antarmuka antara perangkat lunak yang menghasilkan suara

dan kartu suara Anda. Ini adalah perangkat karakter pada node utama 14 dan minor 3.

/dev/fd0

Floppy drive pertama. Jika Anda cukup beruntung memiliki beberapa drive maka mereka akan diberi nomor

berurutan. Ini adalah perangkat karakter pada node utama 2 dan minor 0.

/dev/fb0

Panduan Administrator Sistem Linux

Perangkat framebuffer pertama. Framebuffer adalah lapisan abstraksi antara perangkat lunak dan grafik perangkat keras. Ini berarti bahwa aplikasi tidak perlu tahu tentang jenis perangkat keras yang Anda miliki tetapi hanya bagaimana berkomunikasi dengan API driver framebuffer (Application Programming Interface) yang didefinisikan dan distandarisasi dengan baik. Framebuffer adalah perangkat karakter dan berada di node utama 29 dan minor 0.

/dev/hda

/dev/hda adalah drive IDE utama pada pengontrol IDE utama. /dev/hdb drive budak di pengontrol utama. /dev/hdc, dan /dev/hdd adalah perangkat master dan slave di pengontrol sekunder masing-masing. Setiap disk dibagi menjadi beberapa partisi. Partisi 1–4 adalah partisi primer dan partisi 5 dan di atasnya adalah partisi logis di dalam partisi extended. Oleh karena itu file perangkat yang mereferensikan setiap partisi terdiri dari beberapa bagian. Sebagai contoh /dev/hdc9 referensi partisi 9 (partisi logis di dalam tipe partisi yang diperluas) pada drive IDE master pada pengontrol IDE sekunder. Nomor simpul mayor dan minor agak rumit. Untuk pengontrol IDE pertama, semua partisi adalah perangkat blok pada node utama 3. Drive master hda berada di minor 0 dan drive budak hdb berada di minor 64. Untuk setiap partisi di dalam drive, tambahkan nomor partisi ke nomor node minor untuk drive. Sebagai contoh /dev/hdb5 adalah mayor 3, minor 69 (64 + 5 = 69). Drive pada antarmuka sekunder ditangani dengan cara yang sama, tetapi dengan node utama 22.

/dev/ht0

Tape drive IDE pertama. Drive berikutnya diberi nomor ht1 dll. Mereka adalah perangkat karakter pada simpul utama 37 dan mulai dari simpul kecil 0 untuk ht0 1 untuk ht1 dll.

/dev/js0

Joystick analog pertama. Joystick berikutnya diberi nomor js1, js2 dll. Joystick digital adalah ditelepon djs0, djs1 dan seterusnya. Mereka adalah perangkat karakter pada node utama 15. Joystick analog mulai dari minor node 0 dan naik ke 127 (lebih dari cukup bahkan untuk gamer yang paling fanatik). Digital joystick mulai dari simpul kecil 128.

/dev/lp0

Perangkat printer paralel pertama. Printer berikutnya diberi nomor lp1, lp2 dll. Mereka adalah perangkat pada mode utama 6 dan node kecil mulai dari 0 dan diberi nomor secara berurutan.

/dev/loop0

Perangkat loopback pertama. Perangkat loopback digunakan untuk memasang sistem file yang tidak terletak di perangkat blok lain seperti disk. Misalnya jika Anda ingin memasang image CD ROM iso9660 tanpa membakarnya ke CD, maka Anda perlu menggunakan perangkat loopback untuk melakukannya. Ini biasanya transparan bagi pengguna dan ditangani oleh perintah mount. Lihat halaman manual untuk mount dan losetup. Perangkat loopback adalah perangkat blok pada node utama 7 dan dengan node kecil mulai dari 0 dan diberi nomor secara berurutan.

/dev/md0

Grup metadisk pertama. Metadisk terkait dengan RAID (Redundant Array of Independent Disks) perangkat. Silakan merujuk ke RAID HOWTO terbaru di LDP untuk detail lebih lanjut. Ini dapat ditemukan di <http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html>. Perangkat metadisk adalah perangkat blok pada node utama 9 dengan node kecil dimulai dari 0 dan diberi nomor secara berurutan.

/dev/pencampur

Ini adalah bagian dari driver OSS (Open Sound System). Lihat dokumentasi OSS di <http://www.opensound.com> untuk lebih jelasnya. Ini adalah perangkat karakter pada node utama 14, node kecil 0.

/dev/null

Ember kecil. Lubang hitam tempat Anda dapat mengirim data agar tidak pernah terlihat lagi. Apa saja yang dikirim ke

Panduan Administrator Sistem Linux

`/dev/null` akan hilang. Ini dapat berguna jika, misalnya, Anda ingin menjalankan perintah tetapi tidak ada umpan balik yang muncul di terminal. Ini adalah perangkat karakter pada node utama 1 dan node kecil 3.

`/dev/psaux`

Port mouse PS/2. Ini adalah perangkat karakter pada simpul utama 10, simpul kecil 1.

`/dev/pda`

Disk IDE port paralel. Ini dinamai mirip dengan disk pada pengontrol IDE internal (`/dev/hd*`). Mereka adalah perangkat blok pada node utama 45. Node kecil memerlukan sedikit penjelasan lebih lanjut di sini. Perangkat pertama adalah `/dev/pdada` pada node minor 0. Partisi pada perangkat ini ditemukan dengan menambahkan nomor partisi ke nomor minor perangkat. Setiap perangkat dibatasi masing-masing 15 partisi daripada 63 (batas untuk disk IDE internal). `/dev/pdb` node kecil mulai dari 16, `/dev/pdc` pada 32 dan `/dev/pddd` 48. Jadi misalnya nomor simpul minor untuk `/dev/pdc6` akan menjadi 38 ($32 + 6 = 38$). Skema ini membatasi Anda hingga 4 disk paralel dengan masing-masing 15 partisi.

`/dev/pcd0`

Drive CD ROM port paralel. Ini diberi nomor dari 0 dan seterusnya. Semua adalah perangkat blok di mayor simpul 46. `/dev/pcd0` ada di node minor 0 dengan drive berikutnya berada di node minor 1, 2, 3 dst.

`/dev/pt0`

Perangkat pita port paralel. Kaset tidak memiliki partisi sehingga hanya diberi nomor secara berurutan. Mereka adalah perangkat karakter pada simpul utama 96. Nomor simpul kecil dimulai dari 0 untuk `/dev/pt0`, 1 untuk `/dev/pt1`, dan seterusnya.

`/dev/parport0`

Port paralel mentah. Sebagian besar perangkat yang terpasang ke port paralel memiliki driver sendiri. Ini adalah perangkat untuk mengakses port secara langsung. Ini adalah perangkat karakter pada simpul utama 99 dengan simpul kecil 0. Perangkat berikutnya setelah yang pertama diberi nomor secara berurutan untuk menambah simpul kecil.

`/dev/acak` atau `/dev/urandom`

Ini adalah generator nomor acak kernel. `/dev/acak` adalah generator non-deterministik yang artinya nilai bilangan berikutnya tidak dapat ditebak dari bilangan sebelumnya. Ini menggunakan entropi perangkat keras sistem untuk menghasilkan angka. Ketika tidak memiliki entropi lagi untuk digunakan maka ia harus menunggu sampai ia telah mengumpulkan lebih banyak sebelum memungkinkan lebih banyak angka untuk dibaca darinya.

`/dev/urandom` bekerja sama. Awalnya ia juga menggunakan entropi perangkat keras sistem, tetapi ketika tidak ada lagi entropi yang digunakan, ia akan terus mengembalikan angka menggunakan rumus penghasil angka acak semu. Ini dianggap kurang aman untuk tujuan vital seperti pembuatan pasangan kunci kriptografi. Jika keamanan menjadi perhatian utama Anda, maka gunakan `/dev/acak`, jika kecepatan lebih penting maka `/dev/urandom` berfungsi dengan baik. Mereka adalah perangkat karakter pada node utama 1 dengan node minor 8 untuk `/dev/acak` dan 9 untuk `/dev/urandom`.

`/dev/sda`

Drive SCSI pertama pada bus SCSI pertama. Drive berikut diberi nama yang mirip dengan drive IDE. `/dev/sdb` adalah drive SCSI kedua, `/dev/sdc` adalah drive SCSI ketiga, dan seterusnya.

`/dev/ttyS0`

Port serial pertama. Sering kali ini adalah port yang digunakan untuk menghubungkan modem eksternal ke sistem Anda.

`/dev/n01`

Ini adalah cara sederhana untuk mendapatkan banyak 0. Setiap kali Anda membaca dari perangkat ini, itu akan mengembalikan 0. Ini kadang-kadang bisa berguna, misalnya ketika Anda menginginkan file dengan panjang tetap tetapi tidak terlalu peduli apa isinya. Ini adalah perangkat karakter pada node utama 1 dan node kecil 5.

3.5. /usr berkas sistem.

/usr sistem file sering kali berukuran besar, karena semua program diinstal di sana. Semua file di/usr biasanya datang dari distribusi Linux; program yang diinstal secara lokal dan hal-hal lain berada di bawah/usr/local. Ini membuatnya memungkinkan untuk memperbarui sistem dari versi distribusi baru, atau bahkan distribusi yang sama sekali baru, tanpa harus menginstall semua program lagi. Beberapa subfolder dari/usr tercantum di bawah ini (beberapa folder yang kurang penting telah dihapus; lihat FSSTND untuk informasi lebih lanjut).

/usr/X11R6.

Sistem X Window, semua file. Untuk menyederhanakan pengembangan dan penginstalan X, file X memiliki belum terintegrasi ke dalam sistem lainnya. Ada pohon folder di bawah ini/usr/X11R6mirip dengan yang di bawah ini/usr diri.

/usr/bin.

Hampir semua perintah pengguna. Beberapa perintah ada di/bin atau di/usr/local/bin.

/usr/sbin

Perintah administrasi sistem yang tidak diperlukan pada sistem file root, misalnya, sebagian besar server program.

/usr/bagikan/man, /usr/bagikan/info, /usr/bagikan/doc

Halaman manual, dokumen Info GNU, dan berbagai file dokumentasi lainnya, masing-masing.

/usr/termasuk

File header untuk bahasa pemrograman C. Ini sebenarnya harus di bawah/usr/libuntuk konsistensi, tetapi tradisi sangat mendukung nama ini.

/usr/lib

File data yang tidak berubah untuk program dan subsistem, termasuk beberapa file konfigurasi seluruh situs.

Nama lib berasal dari perpustakaan; awalnya perpustakaan subrutin pemrograman disimpan di /usr/lib.

/usr/local

Tempat untuk perangkat lunak yang diinstal secara lokal dan file lainnya. Distribusi mungkin tidak menginstal apa pun di sini.

Ini dicadangkan hanya untuk penggunaan administrator lokal. Dengan cara ini dia dapat benar-benar yakin bahwa tidak ada pembaruan atau peningkatan pada distribusinya yang akan menimpa perangkat lunak tambahan apa pun yang telah dia instal secara lokal.

3.6. /var berkas sistem

/var berisi data yang diubah saat sistem berjalan normal. Ini khusus untuk setiap sistem, yaitu, tidak dibagikan melalui jaringan dengan komputer lain.

/var/cache/man

Cache untuk halaman manual yang diformat sesuai permintaan. Sumber untuk halaman manual biasanya disimpan

di /usr/share/pria/pria?/ (di mana ? adalah bagian manual. Lihat halaman manual untuk man di bagian 7); beberapa halaman manual mungkin datang dengan versi pra-format, yang mungkin disimpan di/usr/share/pria/kucing* . Halaman manual lainnya perlu diformat saat pertama kali dilihat; versi yang diformat kemudian disimpan di/var/cache/man sehingga orang berikutnya yang melihat halaman yang sama tidak perlu menunggu untuk diformat.

/var/permainan

Data variabel apa pun yang termasuk dalam game di /usr harus ditempatkan di sini. Ini jika /usr sudah terpasang

Panduan Administrator Sistem Linux

baca saja.

/var/lib

File yang berubah saat sistem berjalan normal.

/var/local

Data variabel untuk program yang diinstal di `/usr/local` (yaitu, program yang telah diinstal oleh administrator sistem). Perhatikan bahwa bahkan program yang diinstal secara lokal harus menggunakan yang lain/`var` folder jika sesuai, misalnya, `/var/kunci`.

/var/kunci

Kunci file. Banyak program mengikuti konvensi untuk membuat file kunci di `/var/kunci` untuk menunjukkan bahwa mereka menggunakan perangkat atau file tertentu. Program lain akan melihat file kunci dan tidak akan mencoba untuk menggunakan perangkat atau file.

/var/log

File log dari berbagai program, terutama login (`/var/log/wtmp`, yang mencatat semua login dan logout ke sistem) dan `syslog` (`/var/log/pesan`, di mana semua kernel dan sistem program pesan biasanya disimpan). File dalam `/var/log` sering dapat tumbuh tanpa batas, dan mungkin memerlukan pembersihan secara berkala.

/var/mail

Ini adalah lokasi yang disetujui FHS untuk file kotak surat pengguna. Bergantung pada seberapa jauh distribusi Anda menuju kepatuhan FHS, file-file ini mungkin masih disimpan di `/var/spool/mail`.

/var/run

File yang berisi informasi tentang sistem yang valid hingga sistem di-boot berikutnya. Untuk contoh, `/var/run/utmp` berisi informasi tentang orang-orang yang sedang login.

/var/spool

Folder untuk berita, antrian printer, dan pekerjaan antrian lainnya. Setiap gulungan yang berbeda memilikinya sendiri subfolder di bawah `/var/spool`, misalnya, spool berita ada di `/var/spool/berita`. Perhatikan bahwa beberapa penginstalan yang tidak sepenuhnya sesuai dengan versi terbaru FHS mungkin memiliki kotak surat pengguna di bawah `/var/spool/mail`.

/var/tmp

File sementara yang besar atau yang perlu ada untuk waktu yang lebih lama dari yang diizinkan `/tmp`. (Meskipun administrator sistem mungkin tidak mengizinkan file yang sangat lama masuk `/var/tmp` salah satu.)

3.7. /proc berkas sistem

`/proc` filesystem berisi sistem file ilusi. Itu tidak ada pada disk. Sebagai gantinya, kernel membuatnya di memori. Ini digunakan untuk memberikan informasi tentang sistem (awalnya tentang proses, maka namanya). Beberapa file dan folder yang lebih penting dijelaskan di bawah ini. `/proc` sistem file dijelaskan secara lebih rinci di `prochalaman` manual.

/proc/1

Folder dengan informasi tentang proses nomor 1. Setiap proses memiliki folder di bawah ini/`proc` dengan nama menjadi nomor identifikasi prosesnya.

/proc/cpuinfo

Informasi tentang prosesor, seperti jenis, merek, model, dan kinerjanya.

/proc/perangkat

Daftar driver perangkat yang dikonfigurasi ke dalam kernel yang sedang berjalan.

/proc/dma

Menunjukkan saluran DMA mana yang sedang digunakan saat ini.

/proc/filesystem

Panduan Administrator Sistem Linux

Sistem file dikonfigurasi ke dalam kernel.

/proc/interrupts

Menunjukkan interupsi mana yang digunakan, dan berapa banyak interupsi yang telah ada.

/proc/ioprots

Port I/O mana yang sedang digunakan saat ini.

/proc/kcore

Gambar memori fisik sistem. Ini ukurannya persis sama dengan fisikmu memori, tetapi tidak terlalu memakan banyak memori; itu dihasilkan dengan cepat saat program mengaksesnya. (Ingat: kecuali Anda menyalinnya di tempat lain, tidak ada di bawah `/proc` memakan ruang disk sama sekali.)

/proc/kmsg

Keluaran pesan oleh kernel. Ini juga diarahkan ke `syslog`.

/proc/ksyms

Tabel simbol untuk kernel.

/proc/loadavg

`Rata-rata beban' sistem; tiga indikator tidak berarti tentang seberapa banyak pekerjaan yang harus dilakukan sistem lakukan saat ini.

/proc/meminfo

Informasi tentang penggunaan memori, baik fisik maupun swap.

/proc/modul

Modul kernel mana yang dimuat saat ini.

/proc/net

Informasi status tentang protokol jaringan.

/proc/self

Tautan simbolis ke folder proses dari program yang sedang dilihat `/proc`. Ketika dua proses melihat `/proc`, mereka mendapatkan tautan yang berbeda. Ini terutama kenyamanan untuk memudahkan program untuk mendapatkan di folder proses mereka.

/proc/stat

Berbagai statistik tentang sistem, seperti jumlah kesalahan halaman sejak sistem di-boot.

/proc/waktu aktif

Waktu sistem telah habis.

/proc/versi

Versi kernelnya.

Perhatikan bahwa meskipun file di atas cenderung merupakan file teks yang mudah dibaca, terkadang file tersebut dapat diformat dengan cara yang tidak mudah dicerna. Ada banyak perintah yang tidak lebih dari membaca file di atas dan memformatnya agar lebih mudah dipahami. Misalnya, program bebas berbunyi `/proc/meminfo` mengonversi jumlah yang diberikan dalam byte ke kilobyte (dan juga menambahkan sedikit lebih banyak informasi).

Bab 4. Perangkat Keras, Perangkat, dan Alat

"Pengetahuan berbicara, tetapi kebijaksanaan mendengarkan." Jimi Hendrix

Bab ini memberikan gambaran umum tentang apa itu file perangkat, dan cara membuatnya. Daftar kanonik file perangkat adalah `/usr/src/linux/Documentation/devices.txt` jika Anda telah menginstal kode sumber kernel Linux di sistem Anda. Perangkat yang tercantum di sini sudah benar pada kernel versi 2.6.8.

4.1. Utilitas Perangkat Keras

4.1.1. Skrip MAKEDEV

Sebagian besar file perangkat sudah dibuat dan siap digunakan setelah Anda menginstal sistem Linux Anda. Jika kebetulan Anda perlu membuat yang tidak disediakan, maka Anda harus mencoba menggunakan skrip MAKEDEV terlebih dahulu. Script ini biasanya terletak di `/dev/MAKEDEV` tetapi mungkin juga memiliki salinan (atau tautan simbolik) di `/sbin/MAKEDEV`. Jika ternyata tidak berada di jalur Anda, maka Anda perlu menentukan jalurnya secara eksplisit.

Secara umum perintah tersebut digunakan sebagai:

```
# /dev/MAKEDEV v ttyS0
buat ttyS0 c 4 64 root: dialout 0660
```

Ini akan membuat file perangkat `/dev/ttyS0` dengan node utama 4 dan node kecil 64 sebagai perangkat karakter dengan izin akses 0660 dengan root pemilik dan dialout grup.

`ttyS0` adalah port serial. Nomor node mayor dan minor adalah angka yang dipahami oleh kernel. Kernel mengacu pada perangkat keras sebagai angka, ini akan sangat sulit untuk kita ingat, jadi kita menggunakan nama file. Izin akses 0660 berarti izin baca dan tulis untuk pemilik (dalam hal ini root) dan izin baca dan tulis untuk anggota grup (dalam hal ini dialout) tanpa akses untuk orang lain.

4.1.2. Perintah mknod

MAKEDEV adalah cara yang disukai untuk membuat file perangkat yang tidak ada. Namun terkadang skrip MAKEDEV tidak mengetahui file perangkat yang ingin Anda buat. Di sinilah perintah `mknod` masuk. Untuk menggunakan `mknod` Anda perlu mengetahui nomor node mayor dan minor untuk perangkat yang Anda gunakan.

ingin membuat. Itu `devices.txt` file dalam dokumentasi sumber kernel adalah sumber kanonik dari informasi ini.

Sebagai contoh, mari kita anggap bahwa versi skrip MAKEDEV kita tidak tahu cara membuat `/dev/ttyS0` berkas perangkat. Kita perlu menggunakan `mknod` untuk membuatnya. Kita tahu dari melihat `devices.txt` bahwa itu harus menjadi perangkat karakter dengan nomor utama 4 dan nomor kecil 64. Jadi sekarang kita tahu semua yang kita butuhkan untuk membuat file.

```
# mknod /dev/ttyS0 c 4 64
# chown root.dialout /dev/ttyS0
# chmod 0644 /dev/ttyS0
# ls -l /dev/ttyS0
crw-rw---- 1 root dialout 4, 64 23 Okt 18:23 /dev/ttyS0
```

Seperti yang Anda lihat, lebih banyak langkah diperlukan untuk membuat file. Dalam contoh ini Anda dapat melihat proses yang diperlukan. Sangat tidak mungkin bahwa file `ttyS0` tidak akan disediakan oleh skrip `MAKEDEV`, tetapi cukup untuk menggambarkan maksudnya.

4.1.3. Perintah `lspci`

`lspci`

UNTUK DITAMBAHKAN

4.1.4. Perintah `lsdev`

`lsdev`

UNTUK DITAMBAHKAN

4.1.5. Perintah `lsusb`

`lsusb`

UNTUK DITAMBAHKAN

4.1.6. Perintah `lsraid`

`lsraid`

UNTUK DITAMBAHKAN

4.1.7. Perintah `hdparm`

`hdparm`

UNTUK DITAMBAHKAN

4.1.8. Lebih Banyak Sumber Daya Perangkat Keras

Informasi lebih lanjut tentang sumber daya perangkat keras apa yang digunakan kernel dapat ditemukan di `/proc` folder. Lihat Bagian 3.7 di bab 3.

4.2. Modul Kernel

Bagian ini akan membahas modul kernel.

UNTUK DITAMBAHKAN

4.2.1. lsmod

lsmod

UNTUK DITAMBAHKAN

4.2.2. insmod

insmod

UNTUK DITAMBAHKAN

4.2.3. demmod

demmod

UNTUK DITAMBAHKAN

4.2.4. rmmod

rmmod

UNTUK DITAMBAHKAN

4.2.5. modprobe

modprobe

UNTUK

DITAMBAHKAN

Bab 5. Menggunakan Disk dan Media Penyimpanan Lainnya

"Pada disk yang jelas Anda dapat mencari selamanya."

Saat Anda menginstal atau memutakhirkan sistem Anda, Anda perlu melakukan cukup banyak pekerjaan pada disk Anda. Anda harus membuat sistem file pada disk Anda sehingga file dapat disimpan di dalamnya dan menyediakan ruang untuk berbagai bagian sistem Anda.

Bab ini menjelaskan semua kegiatan awal tersebut. Biasanya, setelah Anda mengatur sistem Anda, Anda tidak perlu melakukan pekerjaan itu lagi, kecuali menggunakan disket. Anda harus kembali ke bab ini jika Anda menambahkan disk baru atau ingin menyempurnakan penggunaan disk Anda.

Tugas dasar dalam mengelola disk adalah:

- Format disk Anda. Ini melakukan berbagai hal untuk mempersiapkannya untuk digunakan, seperti memeriksa bad sector. (Pemformatan saat ini tidak diperlukan untuk sebagian besar hard disk.)
- Mempartisi hard disk, jika ingin digunakan untuk beberapa aktivitas yang tidak boleh saling mengganggu. Salah satu alasan untuk mempartisi adalah untuk menyimpan sistem operasi yang berbeda pada disk yang sama. Alasan lainnya adalah untuk menjaga file pengguna terpisah dari file sistem, yang menyederhanakan pencadangan dan membantu melindungi file sistem dari kerusakan.
- Buat sistem file (dengan tipe yang sesuai) pada setiap disk atau partisi. Disk tidak berarti apa-apa bagi Linux sampai Anda membuat sistem file; kemudian file dapat dibuat dan diakses di atasnya.
- Pasang sistem file yang berbeda untuk membentuk struktur pohon tunggal, baik secara otomatis, atau manual sesuai kebutuhan. (Sistem file yang dipasang secara manual biasanya juga perlu dilepas secara manual.)

Bab 6 berisi informasi tentang memori virtual dan caching disk, yang juga perlu Anda ketahui saat menggunakan disk.

5.1. Dua jenis perangkat

UNIX, dan oleh karena itu Linux, mengenali dua jenis perangkat yang berbeda: perangkat blok akses acak (seperti disk), dan perangkat karakter (seperti kaset dan saluran serial), beberapa di antaranya mungkin serial, dan beberapa akses acak. Setiap perangkat yang didukung diwakili dalam sistem file sebagai file perangkat. Saat Anda membaca atau menulis file perangkat, data berasal dari atau masuk ke perangkat yang diwakilinya. Dengan cara ini tidak ada program khusus (dan tidak ada metodologi pemrograman aplikasi khusus, seperti menangkap interupsi atau polling port serial) yang diperlukan untuk mengakses perangkat; misalnya, untuk mengirim file ke printer, orang bisa mengatakan

```
$ cat nama file > /dev/lp1
$
```

dan isi file dicetak (file harus, tentu saja, dalam bentuk yang dimengerti printer). Namun, karena bukanlah ide yang baik untuk meminta beberapa orang memasukkan file mereka ke printer pada saat yang bersamaan, seseorang biasanya menggunakan program khusus untuk mengirim file yang akan dicetak (biasanya `lpr`).

Panduan Administrator Sistem Linux

Program ini memastikan bahwa hanya satu file yang sedang dicetak pada satu waktu, dan secara otomatis akan mengirim file ke printer segera setelah selesai dengan file sebelumnya. Sesuatu yang serupa diperlukan untuk sebagian besar perangkat. Faktanya, seseorang jarang perlu khawatir tentang file perangkat sama sekali.

Karena perangkat muncul sebagai file dalam sistem file (dalam `/dev` folder), mudah untuk melihat file perangkat apa yang ada, menggunakan `ls` atau perintah lain yang sesuai. Dalam output `ls l`, kolom pertama berisi jenis file

dan izinnya. Misalnya, memeriksa perangkat serial mungkin memberikan

```
$ ls -l /dev/ttyS0
crw-rw-r-- 1 root dialout 4, 64 19 Agustus 18:56 /dev/ttyS0
$
```

Karakter pertama di kolom pertama, yaitu `c` di `crw-rw-rw-` di atas, memberi tahu pengguna yang diinformasikan jenis file, dalam hal ini perangkat karakter. Untuk file biasa, karakter pertama adalah `-`, untuk folder adalah `D`, dan untuk memblokir perangkat `B`; lihat halaman manual `ls` untuk informasi lebih lanjut.

Perhatikan bahwa biasanya semua file perangkat ada meskipun perangkat itu sendiri mungkin tidak diinstal. Jadi hanya karena Anda punya file `/dev/sda`, itu tidak berarti bahwa Anda benar-benar memiliki hard disk SCSI. Memiliki semua file perangkat membuat program penginstalan lebih sederhana, dan membuatnya lebih mudah untuk menambahkan perangkat keras baru (tidak perlu mengetahui parameter yang benar untuk dan membuat file perangkat untuk perangkat baru).

5.2. Hard disk

Subbagian ini memperkenalkan terminologi yang terkait dengan hard disk. Jika Anda sudah mengetahui istilah dan konsepnya, Anda dapat melewati subbagian ini.

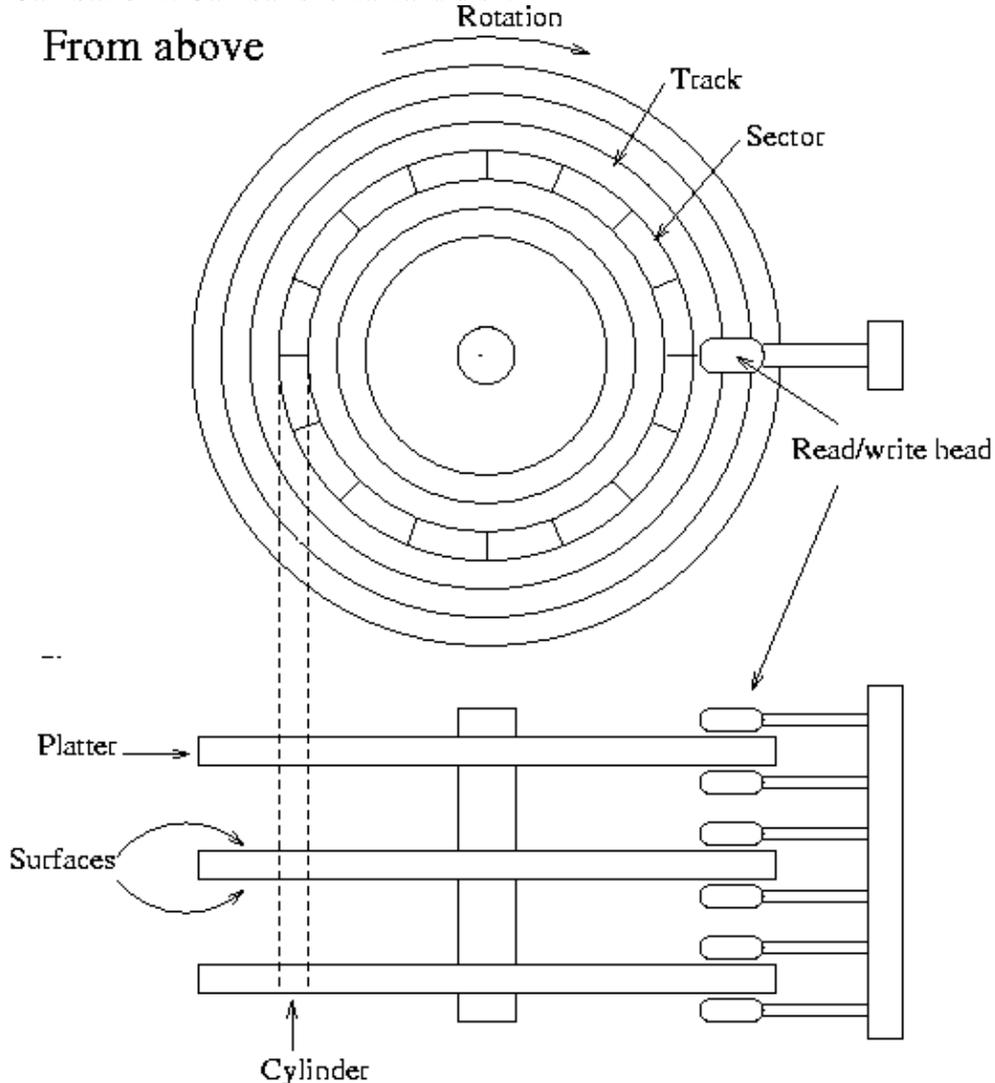
Lihat [Gambar 5-1](#) untuk gambaran skema bagian-bagian penting dalam hard disk. Sebuah hard disk terdiri dari satu atau lebih piringan aluminium melingkar, yang salah satu atau kedua permukaannya dilapisi dengan bahan magnetik yang digunakan untuk merekam data. Untuk setiap permukaan, ada kepala baca-tulis yang memeriksa atau mengubah data yang direkam. Piring-piring berputar pada sumbu yang sama; kecepatan rotasi tipikal adalah 5400 atau 7200 rotasi per menit, meskipun hard disk berkinerja tinggi memiliki kecepatan lebih tinggi dan disk lama mungkin memiliki kecepatan lebih rendah. Kepala bergerak di sepanjang jari-jari piringan; gerakan ini dikombinasikan dengan rotasi piringan memungkinkan kepala untuk mengakses semua bagian permukaan.

Prosesor (CPU) dan disk sebenarnya berkomunikasi melalui pengontrol disk. Hal ini membuat komputer lain tidak mengetahui cara menggunakan drive, karena pengontrol untuk berbagai jenis disk dapat dibuat menggunakan antarmuka yang sama ke bagian komputer lainnya. Oleh karena itu, komputer dapat mengatakan hanya "hei disk, berikan apa yang saya inginkan", alih-alih serangkaian sinyal listrik yang panjang dan rumit untuk menggerakkan kepala ke lokasi yang tepat dan menunggu posisi yang benar untuk datang di bawah kepala dan melakukan semua hal tidak menyenangkan lainnya yang diperlukan. (Pada kenyataannya, antarmuka ke pengontrol masih kompleks, tetapi jauh lebih sedikit daripada yang seharusnya.) Pengontrol juga dapat melakukan hal-hal lain, seperti caching, atau penggantian bad sector otomatis.

Di atas biasanya semua yang perlu dipahami tentang perangkat keras. Ada juga hal-hal lain, seperti motor yang memutar piringan dan menggerakkan kepala, dan elektronik yang mengontrol pengoperasian bagian mekanis, tetapi sebagian besar tidak relevan untuk memahami prinsip kerja hard disk.

Permukaan biasanya dibagi menjadi cincin konsentris, yang disebut trek, dan ini pada gilirannya dibagi menjadi sektor-sektor. Pembagian ini digunakan untuk menentukan lokasi pada hard disk dan mengalokasikan ruang disk ke file. Untuk menemukan tempat tertentu pada hard disk, dapat dikatakan ``surface 3, track 5, sector 7''. Biasanya jumlah sektor sama untuk semua trek, tetapi beberapa hard disk menempatkan lebih banyak sektor di trek luar (semua sektor memiliki ukuran fisik yang sama, jadi lebih banyak yang muat di trek luar yang lebih panjang). Biasanya, sebuah sektor akan menampung 512 byte data. Disk itu sendiri tidak dapat menangani jumlah data yang lebih kecil dari satu sektor.

Gambar 5-1. Gambar skema hard disk.



Setiap permukaan dibagi menjadi trek (dan sektor) dengan cara yang sama. Ini berarti bahwa ketika kepala untuk satu permukaan berada di trek, kepala untuk permukaan lainnya juga berada di trek yang sesuai. Semua trek yang sesuai yang diambil bersama disebut silinder. Memindahkan head dari satu track (silinder) ke track yang lain membutuhkan waktu, sehingga dengan menempatkan data yang sering diakses secara bersamaan (misalnya file) sehingga berada dalam satu silinder, tidak perlu memindahkan head untuk membaca semua itu. Ini meningkatkan kinerja. Tidak selalu mungkin untuk menempatkan file seperti ini; file yang disimpan di beberapa tempat pada disk disebut terfragmentasi.

Jumlah permukaan (atau kepala, yang merupakan hal yang sama), silinder, dan sektor sangat bervariasi; spesifikasi jumlah masing-masing disebut geometri hard disk. Geometri biasanya disimpan di lokasi memori

khusus bertenaga baterai yang disebut RAM CMOS, dari mana sistem operasi dapat mengambilnya selama bootup atau inisialisasi driver.

Sayangnya, BIOS memiliki batasan desain, yang membuatnya tidak mungkin untuk menentukan nomor trek yang lebih besar dari 1024 di RAM CMOS, yang terlalu kecil untuk hard disk besar. Untuk mengatasinya, pengontrol hard disk berbohong tentang geometri, dan menerjemahkan alamat yang diberikan oleh komputer menjadi sesuatu yang sesuai dengan kenyataan. Misalnya, hard disk mungkin memiliki 8 kepala, 2048 trek, dan 35 sektor per trek. Pengontrolnya dapat berbohong ke komputer dan mengklaim bahwa ia memiliki 16 kepala, 1024 trek, dan 35 sektor per trek, sehingga tidak melebihi batas trek, dan menerjemahkan alamat yang diberikan komputer dengan membagi dua nomor kepala, dan menggandakan nomor pelacak. Matematika bisa lebih rumit dalam kenyataan, karena angkanya tidak sebaik di sini (tetapi sekali lagi, detailnya tidak relevan untuk memahami prinsipnya). Terjemahan ini mendistorsi pandangan sistem operasi tentang bagaimana disk diatur, sehingga membuatnya tidak praktis untuk menggunakan trik semua-data-pada-satu-silinder untuk meningkatkan kinerja.

Terjemahan hanya masalah untuk disk IDE. Disk SCSI menggunakan nomor sektor berurutan (yaitu, pengontrol menerjemahkan nomor sektor berurutan ke kepala, silinder, dan triplet sektor), dan metode yang sama sekali berbeda untuk CPU untuk berbicara dengan pengontrol, sehingga mereka terisolasi dari masalah. Namun, perhatikan bahwa komputer mungkin juga tidak mengetahui geometri sebenarnya dari disk SCSI.

Karena Linux sering kali tidak mengetahui geometri sebenarnya dari sebuah disk, sistem filenya bahkan tidak mencoba untuk menyimpan file dalam satu silinder. Sebagai gantinya, ia mencoba untuk menetapkan sektor bernomor urut ke file, yang hampir selalu memberikan kinerja yang serupa. Masalah ini semakin diperumit oleh cache pengontrol, dan pengambilan otomatis dilakukan oleh pengontrol.

Setiap hard disk diwakili oleh file perangkat terpisah. Mungkin (biasanya) hanya ada dua atau empat hard disk IDE. Ini dikenal sebagai `/dev/hda`, `/dev/hdb`, `/dev/hdc`, dan `/dev/hdd`, masing-masing. Hard disk SCSI dikenal sebagai `/dev/sda`, `/dev/sdb`, dan seterusnya. Konvensi penamaan serupa ada untuk jenis hard disk lainnya; lihat Bab 4 untuk informasi lebih lanjut. Perhatikan bahwa file perangkat untuk hard disk memberikan akses ke seluruh disk, tanpa memperhatikan partisi (yang akan dibahas di bawah), dan sangat mudah untuk mengacaukan partisi atau data di dalamnya jika Anda tidak hati-hati. File perangkat disk biasanya hanya digunakan untuk mendapatkan akses ke master boot record (yang juga akan dibahas di bawah).

5.3. Jaringan Area Penyimpanan Konsep

SAN adalah jaringan penyimpanan khusus yang menyediakan akses tingkat blok ke LUN. LUN, atau nomor unit logis, adalah disk virtual yang disediakan oleh SAN. Administrator sistem memiliki akses dan hak yang sama ke LUN seolah-olah itu adalah disk yang langsung terpasang padanya. Administrator dapat mempartisi, dan memformat disk dengan cara apa pun yang dia pilih.

Dua protokol jaringan yang umum digunakan dalam SAN adalah saluran serat dan iSCSI. Jaringan saluran serat sangat cepat dan tidak terbebani oleh lalu lintas jaringan lain di LAN perusahaan. Namun, itu sangat mahal. Kartu Fibre channel masing-masing berharga sekitar \$1000.00 USD. Mereka juga memerlukan sakelar saluran serat khusus.

iSCSI adalah teknologi baru yang mengirimkan perintah SCSI melalui jaringan TCP/IP. Meskipun metode ini mungkin tidak secepat jaringan Fibre Channel, metode ini menghemat uang dengan menggunakan perangkat keras jaringan yang lebih murah.

Lebih Banyak Yang Akan Ditambahkan

Bab 5. Menggunakan Disk dan Media

5.4. Penyimpanan Terlampir Jaringan Draft

NAS menggunakan jaringan Ethernet perusahaan Anda yang ada untuk memungkinkan akses ke disk bersama. Ini adalah akses tingkat sistem file. Administrator sistem tidak memiliki kemampuan untuk mempartisi atau memformat disk karena berpotensi digunakan bersama oleh banyak komputer. Teknologi ini biasanya digunakan untuk menyediakan beberapa workstation akses ke data yang sama.

Mirip dengan SAN, NAS perlu menggunakan protokol untuk memungkinkan akses ke disknya. Dengan NAS ini juga

CIFS/Samba, atau NFS. Secara tradisional CIFS digunakan dengan jaringan Microsoft Windows, dan NFS digunakan dengan jaringan UNIX & Linux. Namun, dengan Samba, mesin Linux juga dapat menggunakan bagian CIFS.

Apakah ini berarti bahwa server Windows 2003 atau kotak Linux Anda adalah server NAS karena mereka menyediakan akses ke drive bersama melalui jaringan Anda? Ya begitulah. Anda juga dapat membeli perangkat NAS dari sejumlah produsen. Perangkat ini dirancang khusus untuk menyediakan akses data berkecepatan tinggi.

Lebih Banyak Yang Akan Ditambahkan

5.4.1. NFS

UNTUK DITAMBAHKAN

5.4.2. CIFS

UNTUK DITAMBAHKAN

5.5. disket

Floppy disk terdiri dari membran fleksibel yang ditutupi pada satu atau kedua sisinya dengan bahan magnetik yang mirip dengan hard disk. Floppy disk itu sendiri tidak memiliki kepala baca-tulis, yang disertakan dalam drive. Sebuah floppy sesuai dengan satu platter dalam hard disk, tetapi dapat dilepas dan satu drive dapat digunakan untuk mengakses berbagai floppy, dan floppy yang sama dapat dibaca oleh banyak drive, sedangkan hard disk adalah satu unit yang tak terpisahkan.

Seperti hard disk, floppy dibagi menjadi trek dan sektor (dan dua trek yang sesuai di kedua sisi floppy membentuk silinder), tetapi jumlahnya jauh lebih sedikit daripada di hard disk.

Sebuah floppy drive biasanya dapat menggunakan beberapa jenis disk yang berbeda; misalnya, drive 3,5 inci dapat menggunakan keduanya 720

Disk KB dan 1,44 MB. Karena drive harus beroperasi sedikit berbeda dan sistem operasi harus mengetahui seberapa besar disk, ada banyak file perangkat untuk floppy drive, satu per kombinasi drive dan jenis disk. Karena itu, `/dev/fd0` adalah floppy drive (fd0) pertama, yang harus drive 3,5 inci, menggunakan disk kepadatan tinggi (H) 3,5 inci dengan ukuran 1440 KB (1440), yaitu, floppy HD normal 3,5 inci.

Namun, nama untuk floppy drive rumit, dan oleh karena itu Linux memiliki jenis perangkat floppy khusus yang secara otomatis mendeteksi jenis disk di dalam drive. Ia bekerja dengan mencoba membaca sektor pertama dari floppy yang baru dimasukkan menggunakan jenis floppy yang berbeda sampai menemukan

yang benar. Ini secara alami mengharuskan floppy diformat terlebih dahulu. Perangkat otomatis disebut `/dev/fd0`, `/dev/fd1`, dan seterusnya.

Parameter yang digunakan perangkat otomatis untuk mengakses disk juga dapat diatur menggunakan program `setfdprm`. Ini dapat berguna jika Anda perlu menggunakan disk yang tidak mengikuti ukuran disket biasa, misalnya, jika mereka memiliki jumlah sektor yang tidak biasa, atau jika deteksi otomatis karena alasan tertentu gagal dan file perangkat yang sesuai hilang.

Linux dapat menangani banyak format floppy disk yang tidak standar selain semua yang standar. Beberapa di antaranya memerlukan penggunaan program pemformatan khusus. Kami akan melewatkan jenis disk ini untuk saat ini, tetapi sementara itu Anda dapat memeriksa `/etc/fdprm` mengajukan. Ini menentukan pengaturan yang dikenali `setfdprm`.

Sistem operasi harus mengetahui kapan disk telah diubah di floppy drive, misalnya, untuk menghindari penggunaan data cache dari disk sebelumnya. Sayangnya, jalur sinyal yang digunakan untuk ini terkadang putus, dan lebih buruk lagi, ini tidak akan selalu terlihat saat menggunakan drive dari dalam MS-DOS. Jika Anda mengalami masalah aneh menggunakan disket, ini mungkin alasannya. Satu-satunya cara untuk memperbaikinya adalah dengan memperbaiki floppy drive.

5.6. CD-ROM

Drive CD-ROM menggunakan disk berlapis plastik yang dapat dibaca secara optik. Informasi direkam pada permukaan disk dalam 'lubang' kecil yang disejajarkan di sepanjang spiral dari pusat ke tepi. Drive mengarahkan sinar laser di sepanjang spiral untuk membaca disk. Ketika laser mengenai sebuah lubang, laser dipantulkan dalam satu cara; ketika menyentuh permukaan halus, itu dipantulkan dengan cara lain. Ini memudahkan pengkodean bit, dan karenanya informasi. Selebihnya mudah, mekanik saja.

Drive CD-ROM lambat dibandingkan dengan hard disk. Sedangkan hard disk biasa akan memiliki waktu pencarian rata-rata kurang dari 15 milidetik, drive CD-ROM cepat dapat menggunakan sepersepuluh detik untuk pencarian. Kecepatan transfer data sebenarnya cukup tinggi pada ratusan kilobyte per detik. Kelambatan berarti bahwa drive CD-ROM tidak menyenangkan untuk digunakan seperti hard disk (beberapa distribusi Linux menyediakan sistem file 'live' pada CD-ROM, sehingga tidak perlu menyalin file ke hard disk, membuat instalasi lebih mudah dan menghemat banyak ruang hard disk), meskipun masih memungkinkan. Untuk menginstal perangkat lunak baru, CD-ROM sangat baik, karena kecepatan maksimum tidak penting selama instalasi.

Ada beberapa cara untuk mengatur data pada CD-ROM. Yang paling populer ditentukan oleh standar internasional ISO 9660. Standar ini menetapkan sistem file yang sangat minimal, yang bahkan lebih kasar daripada yang digunakan MS-DOS. Di sisi lain, sangat minim sehingga setiap sistem operasi harus dapat memetakannya ke sistem aslinya.

Untuk penggunaan UNIX normal, sistem file ISO 9660 tidak dapat digunakan, jadi ekstensi ke standar telah dikembangkan, yang disebut ekstensi Rock Ridge. Rock Ridge memungkinkan nama file yang lebih panjang, tautan simbolik, dan banyak barang lainnya, membuat CD-ROM terlihat kurang lebih seperti sistem file UNIX kontemporer. Lebih baik lagi, sistem file Rock Ridge masih merupakan sistem file ISO 9660 yang valid, sehingga dapat digunakan oleh sistem non-UNIX juga. Linux mendukung ekstensi ISO 9660 dan Rock Ridge; ekstensi dikenali dan digunakan secara otomatis.

Namun, sistem file hanya setengah dari pertempuran. Sebagian besar CD-ROM berisi data yang memerlukan program khusus untuk mengaksesnya, dan sebagian besar program ini tidak berjalan di Linux (kecuali, mungkin, di bawah `dosex`, emulator Linux MS-DOS, atau `wine`, emulator Windows).

Ironisnya mungkin, wine sebenarnya adalah singkatan dari ``Wine Is Not an Emulator''. Wine, lebih tepatnya, adalah pengganti API (Application Program Interface). Silakan lihat dokumentasi anggur di <http://www.winehq.com> untuk informasi lebih lanjut.

Ada juga VMWare, produk komersial, yang mengemulasi seluruh mesin x86 dalam perangkat lunak. Lihat situs web VMWare, <http://www.vmware.com> untuk informasi lebih lanjut.

Drive CD-ROM diakses melalui file perangkat yang sesuai. Ada beberapa cara untuk menghubungkan drive CD-ROM ke komputer: melalui SCSI, melalui kartu suara, atau melalui EIDE. Peretasan perangkat keras yang diperlukan untuk melakukan ini berada di luar cakupan buku ini, tetapi jenis koneksi menentukan file perangkat.

5.7. kaset

Sebuah tape drive menggunakan tape, mirip dengan kaset yang digunakan untuk musik. Kaset bersifat serial, yang berarti bahwa untuk mencapai bagian tertentu darinya, Anda harus terlebih dahulu melewati semua bagian di antaranya. Sebuah disk dapat diakses secara acak, yaitu, Anda dapat melompat langsung ke sembarang tempat pada disk. Akses serial kaset membuat mereka lambat.

Di sisi lain, kaset relatif murah untuk dibuat, karena tidak perlu cepat. Mereka juga dapat dengan mudah dibuat cukup panjang, dan karena itu dapat berisi sejumlah besar data. Hal ini membuat kaset sangat cocok untuk hal-hal seperti pengarsipan dan pencadangan, yang tidak memerlukan kecepatan besar, tetapi mendapat manfaat dari biaya rendah dan kapasitas penyimpanan yang besar.

5.8. Memformat

Memformat adalah proses penulisan tanda pada media magnetik yang digunakan untuk menandai trek dan sektor. Sebelum disk diformat, permukaan magnetiknya adalah sinyal magnetik yang berantakan. Ketika diformat, beberapa urutan dibawa ke dalam kekacauan dengan menggambar garis pada dasarnya di mana trek pergi, dan di mana mereka dibagi menjadi beberapa sektor. Detail sebenarnya tidak persis seperti ini, tapi itu tidak relevan. Yang penting adalah bahwa disk tidak dapat digunakan kecuali telah diformat.

Terminologinya agak membingungkan di sini: di MS-DOS dan MS Windows, kata pemformatan digunakan untuk mencakup juga proses pembuatan sistem file (yang akan dibahas di bawah). Di sana, kedua proses sering digabungkan, terutama untuk disket. Ketika perbedaan perlu dibuat, pemformatan sebenarnya disebut pemformatan tingkat rendah, sedangkan pembuatan sistem file disebut pemformatan tingkat tinggi. Di lingkungan UNIX, keduanya disebut memformat dan membuat sistem file, jadi itu juga yang digunakan dalam buku ini.

Untuk IDE dan beberapa disk SCSI, pemformatan sebenarnya dilakukan di pabrik dan tidak perlu diulang; maka kebanyakan orang jarang perlu khawatir tentang hal itu. Faktanya, memformat hard disk dapat menyebabkannya bekerja kurang baik, misalnya karena disk mungkin perlu diformat dengan cara yang sangat khusus untuk memungkinkan penggantian bad sector otomatis berfungsi.

Disk yang perlu atau dapat diformat seringkali memerlukan program khusus, karena antarmuka ke logika pemformatan di dalam drive berbeda dari drive ke drive. Program pemformatan sering ada di BIOS pengontrol, atau disediakan sebagai program MS-DOS; tak satu pun dari ini dapat dengan mudah digunakan dari dalam Linux.

Selama memformat, seseorang mungkin menemukan titik buruk pada disk, yang disebut blok buruk atau sektor buruk. Ini kadang-kadang ditangani oleh drive itu sendiri, tetapi meskipun demikian, jika lebih banyak

Panduan Administrator Sistem Linux

yang berkembang, sesuatu perlu dilakukan untuk menghindari penggunaan bagian-bagian disk tersebut. Logika untuk melakukan ini dibangun ke dalam sistem file; cara menambahkan informasi ke dalam sistem file dijelaskan di bawah ini. Atau, seseorang dapat membuat partisi kecil yang hanya menutupi bagian buruk dari disk; pendekatan ini mungkin merupakan ide yang baik jika titik buruknya sangat besar, karena sistem file terkadang dapat mengalami masalah dengan area buruk yang sangat besar.

Floppy diformat dengan `fdformat`. File perangkat floppy yang akan digunakan diberikan sebagai parameter. Misalnya, perintah berikut akan memformat floppy 3,5 inci kepadatan tinggi di floppy drive pertama:

```
$fdformat /dev/fd0H1440
Dua sisi, 80 trek, 18 dtk/trek. Kapasitas total
    1440kB.
Memformat      ...
selesai
Memverifikasi ... selesai
$
```

Perhatikan bahwa jika Anda ingin menggunakan perangkat pendeteksi otomatis (mis. `/dev/fd0`), Anda harus mengatur parameter perangkat dengan `setfdprm` terlebih dahulu. Untuk mencapai efek yang sama seperti di atas, seseorang harus melakukan hal berikut:

```
$ setfdprm /dev/fd0 1440/1440
$fdformat /dev/fd0
Dua sisi, 80 trek, 18 dtk/trek. Kapasitas total
    1440KB.
Memformat      ...
selesai
Memverifikasi ...
selesai
$
```

Biasanya lebih mudah untuk memilih file perangkat yang benar yang cocok dengan jenis floppy. Perhatikan bahwa tidak bijaksana untuk memformat disket untuk memuat lebih banyak informasi daripada yang dirancang.

format fd juga memvalidasi floppy, yaitu, memeriksa blok yang buruk. Ini akan mencoba blok buruk beberapa kali (Anda biasanya dapat mendengar ini, suara drive berubah secara dramatis). Jika floppy hanya sedikit buruk (karena kotoran pada kepala baca/tulis, beberapa kesalahan adalah sinyal palsu), `fdformat` tidak akan mengeluh, tetapi kesalahan nyata akan membatalkan proses validasi. Kernel akan mencetak pesan log untuk setiap kesalahan I/O yang ditemukannya; ini akan pergi ke konsol atau, jika `syslog` sedang digunakan, ke `file/var/log/pesan`. `fdformat` sendiri tidak akan memberi tahu di mana kesalahannya (biasanya orang tidak peduli, disket cukup murah sehingga yang buruk secara otomatis dibuang).

```
$fdformat /dev/fd0H1440
Dua sisi, 80 trek, 18 dtk/trek. Kapasitas total
    1440KB.
Memformat      ...
selesai
Memverifikasi ... baca: Kesalahan tidak diketahui
$
```

Perintah `badblocks` dapat digunakan untuk mencari disk atau partisi apa pun untuk blok buruk (termasuk floppy). Itu tidak memformat disk, sehingga dapat digunakan untuk memeriksa bahkan sistem file yang ada. Contoh di bawah ini memeriksa floppy 3,5 inci dengan dua blok buruk.

```
$ badblock /dev/fd0H1440 1440
718
719
$
```

badblock menampilkan nomor blok dari blok buruk yang ditemukannya. Sebagian besar sistem file dapat menghindari blok buruk seperti itu. Mereka memelihara daftar blok buruk yang diketahui, yang diinisialisasi ketika sistem file dibuat, dan dapat dimodifikasi nanti. Pencarian awal untuk blok buruk dapat dilakukan dengan perintah `mkfs` (yang menginisialisasi sistem file), tetapi pemeriksaan selanjutnya harus dilakukan dengan `badblock` dan blok baru harus ditambahkan dengan `fsck`. Kami akan menjelaskan `mkfs` dan `fsck` nanti.

Banyak disk modern secara otomatis melihat blok yang buruk, dan mencoba untuk memperbaikinya dengan menggunakan blok khusus yang baik sebagai gantinya. Ini tidak terlihat oleh sistem operasi. Fitur ini harus didokumentasikan dalam manual disk, jika Anda ingin tahu apakah itu terjadi. Bahkan disk semacam itu bisa gagal, jika jumlah blok buruk bertambah terlalu besar, meskipun kemungkinan besar disk tersebut akan sangat ~~busuk sehingga tidak dapat digunakan.~~

5.9. Partisi

Sebuah hard disk dapat dibagi menjadi beberapa partisi. Setiap partisi berfungsi seolah-olah itu adalah hard disk yang terpisah. Idenya adalah jika Anda memiliki satu hard disk, dan ingin memiliki, katakanlah, dua sistem operasi di dalamnya, Anda dapat membagi disk menjadi dua partisi. Setiap sistem operasi menggunakan partisinya sesuai keinginan dan tidak menyentuh yang lain. Dengan cara ini kedua sistem operasi dapat hidup berdampingan secara damai pada hard disk yang sama. Tanpa partisi, seseorang harus membeli hard disk untuk setiap sistem operasi.

Floppy biasanya tidak dipartisi. Tidak ada alasan teknis yang menentang hal ini, tetapi karena ukurannya sangat kecil, partisi akan sangat jarang berguna. CD-ROM biasanya juga tidak dipartisi, karena lebih mudah digunakan sebagai satu disk besar, dan jarang diperlukan beberapa sistem operasi dalam satu disk.

5.9.1. MBR, sektor boot, dan tabel partisi

Informasi tentang bagaimana hard disk telah dipartisi disimpan di sektor pertamanya (yaitu, sektor pertama dari trek pertama pada permukaan disk pertama). Sektor pertama adalah master boot record (MBR) dari disk; ini adalah sektor tempat BIOS membaca dan memulai saat mesin pertama kali di-boot. Catatan boot master berisi program kecil yang membaca tabel partisi, memeriksa partisi mana yang aktif (yaitu, ditandai dapat di-boot), dan membaca sektor pertama dari partisi itu, sektor boot partisi (MBR juga merupakan sektor boot, tetapi itu memiliki status khusus dan karena itu nama khusus). Sektor boot ini berisi program kecil lain yang membaca bagian pertama dari sistem operasi yang disimpan di partisi itu (dengan asumsi itu dapat di-boot), dan kemudian memulainya.

Skema partisi tidak dibangun ke dalam perangkat keras, atau bahkan ke dalam BIOS. Ini hanya konvensi yang diikuti oleh banyak sistem operasi. Tidak semua sistem operasi mengikutinya, tetapi mereka adalah pengecualian. Beberapa sistem operasi mendukung partisi, tetapi mereka menempati satu partisi pada hard disk, dan menggunakan metode partisi internal di dalam partisi tersebut. Jenis yang terakhir ada secara damai dengan sistem operasi lain (termasuk Linux), dan tidak memerlukan tindakan khusus, tetapi sistem operasi yang tidak mendukung partisi tidak dapat hidup berdampingan pada disk yang sama dengan sistem operasi lain.

Sebagai tindakan pencegahan keamanan, adalah ide yang baik untuk menuliskan tabel partisi pada selembar kertas, sehingga jika pernah rusak Anda tidak perlu kehilangan semua file Anda. (Tabel partisi yang buruk dapat diperbaiki dengan `fdisk`). Informasi yang relevan diberikan oleh perintah `fdisk l`:

```
$ fdisk l /dev/hda
```

```
Disk /dev/hda: 15 kepala, 57 sektor, 790 silinder
Unit = silinder 855 * 512 byte
```

```

Perang sepa  Mulai   Awal    Akh  Blok    In  Sistem
/dev/hda1    1             1       24   10231+  82  pertukaran Linux
/dev/hda2    25            25      48   10260   83  Linux asli
/dev/hda3    49            49     408  153900  83  Linux asli
/dev/hda4    409           409     790  163305   5  Diperpanjang
/dev/hda5    409           409     744  143611+ 83  Linux asli
/dev/hda6    745           745     790  19636+  83  Linux asli
$

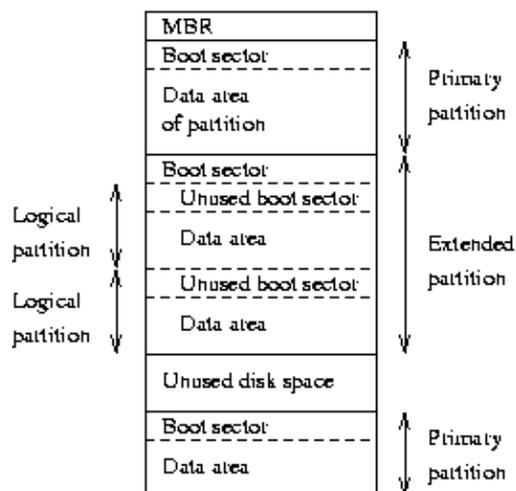
```

5.9.2. Partisi yang diperluas dan logis

Skema partisi asli untuk hard disk PC hanya mengizinkan empat partisi. Ini dengan cepat ternyata terlalu sedikit dalam kehidupan nyata, sebagian karena beberapa orang menginginkan lebih dari empat sistem operasi (Linux, MS-DOS, OS/2, Minix, FreeBSD, NetBSD, atau Windows/NT, untuk beberapa nama), tetapi terutama karena terkadang ada baiknya memiliki beberapa partisi untuk satu sistem operasi. Misalnya, ruang swap biasanya paling baik diletakkan di partisinya sendiri untuk Linux daripada di partisi Linux utama karena alasan kecepatan (lihat di bawah). Untuk mengatasi masalah desain ini, partisi extended diciptakan. Trik ini memungkinkan mempartisi partisi primer menjadi sub-partisi. Partisi utama yang dibagi lagi adalah partisi yang diperluas; sub-partisi adalah partisi logis. Mereka berperilaku seperti partisi primer, tetapi dibuat secara berbeda. Tidak ada perbedaan kecepatan di antara mereka. Dengan menggunakan partisi extended, Anda sekarang dapat memiliki hingga 15 partisi per disk.

Struktur partisi hard disk mungkin terlihat seperti pada Gambar 5-2. Disk dibagi menjadi tiga partisi utama, yang kedua dibagi menjadi dua partisi logis. Bagian dari disk tidak dipartisi sama sekali. Disk secara keseluruhan dan setiap partisi utama memiliki sektor boot.

Gambar 5-2. Contoh partisi harddisk.



5.9.3. Jenis partisi

Tabel partisi (yang ada di MBR, dan yang untuk partisi extended) berisi satu byte per partisi yang mengidentifikasi jenis partisi itu. Ini mencoba mengidentifikasi sistem operasi yang menggunakan partisi, atau untuk apa partisi itu digunakan. Tujuannya adalah untuk menghindari dua sistem operasi yang secara tidak sengaja menggunakan partisi yang sama. Namun, pada kenyataannya, sistem operasi tidak terlalu

Panduan Administrator Sistem Linux

peduli dengan byte tipe partisi; misalnya, Linux sama sekali tidak peduli apa itu. Lebih buruk lagi, beberapa dari mereka menggunakannya secara tidak benar; misalnya, setidaknya beberapa versi DR-DOS mengabaikan bit paling signifikan dari byte, sementara yang lain tidak.

Tidak ada badan standarisasi untuk menentukan apa arti setiap nilai byte, tetapi sejauh menyangkut Linux, berikut adalah daftar tipe partisi sesuai program fdisk.

```
0 Empty 1c Hidden Win95 FA 70 DiskSecure Mult bb Boot Wizard hid
1 FAT12 1e Tersembunyi Win95 FA 75 PC/IX menjadi boot Solaris
2 XENIX root 24 NEC DOS 80 Old Minix c1 DRDOS/dtk (FAT-
3 XENIX usr 39 Paket 9 81 Minix / lama Lin c4 DRDOS/dtk (FAT-
4 FAT16 <32M 3c PartitionMagic 82 Linux menukar c6 DRDOS/dtk (FAT-
5 Diperpanjang 40 Venix 80286 83 Linux c7 Syrinx
6 FAT16 41 PPC PREP Boot 84 OS/2 tersembunyi C: da Non-FS data
7 HPFS/NTFS 42 SFS 85 Linux diperpanjang db CP/M / CTO / .
8 AIX 4d QNX4.x 86 NTFS volume diatur dari Dell Utility
9 AIX yang dapat di-boot 4e QNX4.x bagian ke-2 87 set volume NTFS df BootIt
a OS/2 Boot Manag 4f QNX4.x bagian ke-3 8e Linux LVM e1 akses DOS
B Win95 FAT32 5 ) OnTrack DM 93 Amuba e3 DOS R/O
C Win95 FAT32 (LB 5l OnTrack DM6 Aux 94 Amuba BBT e4 eedStor
```

```
e Win95 FAT16 (LB 52 CP/M 9f BSD/OS eb BeOS fs F Win95 Ext'd (LB 53
OnTrack DM6 Aux a0 IBM Thinkpad hi ee EFI GPT
10 OPUS 54 OnTrackDM6 a5 FreeBSD ef EFI (FAT-12/16/
11 FAT12 Tersembunyi 55 EZ-Drive a6 OpenBSD f0 Linux/PARISC b
12 Diagnosis Compaq 56 Golden Bow a7 NeXTSTEP f1 SpeedStor
14 Hidden FAT16 <3 5c Priam Edisk a8 Darwin UFS f4 SpeedStor
16 FAT16 Tersembunyi 61 SpeedStor a9 NetBSD f2 DOS sekunder
17 HPFS/NTF Tersembunyi 63 GNU HURD atau Sys ab Darwin boot fd Linux raid auto
18 AST SmartSleep 64 Novell Netware b7 BSDI fs fe LANstep
1b Tersembunyi 65 Novell Netware b8 pertukaran ff BBT
```

5.9.4. Mempartisi hard disk

Ada banyak program untuk membuat dan menghapus partisi. Sebagian besar sistem operasi memiliki sistem operasinya sendiri, dan sebaiknya menggunakan sistem operasi masing-masing, untuk berjaga-jaga jika ia melakukan sesuatu yang tidak biasa yang tidak dapat dilakukan oleh yang lain. Banyak program yang disebut fdisk, termasuk program Linux, atau variasinya. Detail tentang penggunaan fdisk Linux diberikan di halaman manualnya. Perintah cfdisk mirip dengan fdisk, tetapi memiliki antarmuka pengguna (layar penuh) yang lebih bagus.

Saat menggunakan disk IDE, partisi boot (partisi dengan file image kernel yang dapat di-boot) harus sepenuhnya berada dalam 1024 silinder pertama. Ini karena disk digunakan melalui BIOS saat boot (sebelum sistem masuk ke mode terproteksi), dan BIOS tidak dapat menangani lebih dari 1024 silinder. Kadang-kadang dimungkinkan untuk menggunakan partisi boot yang hanya sebagian dalam 1024 silinder pertama. Ini berfungsi selama semua file yang dibaca dengan BIOS berada dalam 1024 silinder pertama. Karena ini sulit untuk diatur, adalah ide yang sangat buruk untuk melakukannya; Anda tidak pernah tahu kapan pembaruan kernel atau defragmentasi disk akan menghasilkan sistem yang tidak dapat di-boot. Oleh karena itu, pastikan partisi boot Anda sepenuhnya berada dalam 1024 silinder pertama.

Namun, ini mungkin tidak lagi benar dengan versi LILO yang lebih baru yang mendukung LBA (Logical Block Addressing). Lihat dokumentasi distribusi Anda untuk melihat apakah distribusi tersebut memiliki versi LILO yang didukung LBA.

Beberapa versi disk BIOS dan IDE yang lebih baru, pada kenyataannya, dapat menangani disk dengan lebih dari 1024 silinder. Jika Anda memiliki sistem seperti itu, Anda bisa melupakan masalahnya; jika Anda tidak yakin, taruh di bagian pertama 1024 silinder.

Setiap partisi harus memiliki jumlah sektor yang genap, karena sistem file Linux menggunakan ukuran blok 1 kilobyte, yaitu, dua sektor. Jumlah sektor yang ganjil akan mengakibatkan sektor terakhir tidak digunakan. Ini tidak akan menimbulkan masalah, tetapi jelek, dan beberapa versi fdisk akan memperingatkannya.

Mengubah ukuran partisi biasanya memerlukan pencadangan terlebih dahulu semua yang ingin Anda simpan dari partisi itu (sebaiknya seluruh disk, untuk berjaga-jaga), menghapus partisi, membuat partisi baru, lalu mengembalikan semuanya ke partisi baru. Jika partisi bertambah, Anda mungkin perlu menyesuaikan ukuran (serta mencadangkan dan memulihkan) partisi yang bersebelahan juga.

Karena mengubah ukuran partisi itu menyakitkan, lebih baik untuk memperbaiki partisi pertama kali, atau memiliki sistem cadangan yang efektif dan mudah digunakan. Jika Anda menginstal dari media yang tidak memerlukan banyak campur tangan manusia (misalnya, dari CD-ROM, sebagai lawan dari disket), seringkali mudah untuk memainkan dengan konfigurasi yang berbeda pada awalnya. Karena Anda belum memiliki data untuk dicadangkan, tidak terlalu sulit untuk mengubah beberapa ukuran partisi waktu.

Ada program untuk MS-DOS, yang disebut `fips`, yang mengubah ukuran partisi MS-DOS tanpa memerlukan pencadangan dan pemulihan, tetapi untuk sistem file lain masih diperlukan.

Program `fips` disertakan di sebagian besar distribusi Linux. Manajer partisi komersial "`Partisi Magic`" juga memiliki fasilitas serupa tetapi dengan antarmuka yang lebih bagus. Harap diingat bahwa mempartisi itu berbahaya. Pastikan Anda memiliki cadangan terbaru dari semua data penting sebelum mencoba mengubah ukuran partisi "`parted`". Program `parted` dapat mengubah ukuran partisi jenis lain serta MS-DOS, tetapi terkadang dengan cara yang terbatas. Konsultasikan dokumentasi `parted` sebelum menggunakannya, lebih baik aman daripada menyesal.

5.9.5. File dan partisi perangkat

Setiap partisi dan partisi yang diperluas memiliki file perangkatnya sendiri. Konvensi penamaan untuk file-file ini adalah bahwa nomor partisi ditambahkan setelah nama seluruh disk, dengan konvensi bahwa 1-4 adalah partisi primer (terlepas dari berapa banyak partisi primer yang ada) dan angka yang lebih besar dari 5 adalah partisi logis (terlepas dari di mana partisi utama mereka berada). Sebagai contoh, `/dev/hda1` adalah partisi utama pertama pada hard disk IDE pertama, dan `/dev/sdb7` adalah partisi extended ketiga pada hard disk SCSI kedua.

5.10. Sistem file

5.10.1. Apa itu sistem file?

Sistem file adalah metode dan struktur data yang digunakan sistem operasi untuk melacak file pada disk atau partisi; yaitu, cara file diatur pada disk. Kata ini juga digunakan untuk merujuk pada partisi atau disk yang digunakan untuk menyimpan file atau jenis sistem file. Jadi, seseorang mungkin mengatakan "`Saya memiliki dua sistem file`" yang berarti satu memiliki dua partisi tempat yang satu menyimpan file, atau yang satu menggunakan "`sistem file yang diperluas`", yang berarti jenis sistem file.

Perbedaan antara disk atau partisi dan sistem file yang dikandungnya adalah penting. Beberapa program (termasuk, cukup masuk akal, program yang membuat sistem file) beroperasi langsung pada sektor mentah dari disk atau partisi; jika ada sistem file yang ada di sana akan dihancurkan atau rusak parah. Sebagian besar program beroperasi pada sistem file, dan oleh karena itu tidak akan bekerja pada partisi yang tidak berisi satu (atau yang berisi salah satu jenis).

Sebelum partisi atau disk dapat digunakan sebagai sistem file, itu perlu diinisialisasi, dan struktur data pembukuan perlu ditulis ke disk. Proses ini disebut membuat sistem file.

Sebagian besar jenis sistem file UNIX memiliki struktur umum yang serupa, meskipun detail persisnya sedikit berbeda. Konsep sentralnya adalah superblok, inode, blok data, blok folder, dan blok tipuan. Superblok berisi informasi tentang sistem file secara keseluruhan, seperti ukurannya (informasi yang tepat di sini tergantung pada sistem file). Sebuah inode berisi semua informasi tentang file, kecuali namanya. Nama disimpan dalam folder, bersama dengan nomor inode. Entri folder terdiri dari nama file dan nomor inode yang mewakili file. Inode berisi nomor dari beberapa blok data, yang digunakan untuk menyimpan data dalam file. Akan tetapi, hanya ada ruang untuk beberapa nomor blok data di inode, dan jika diperlukan lebih banyak, lebih banyak ruang untuk penunjuk ke blok data dialokasikan secara dinamis. Blok yang dialokasikan secara dinamis ini adalah blok tidak langsung; namanya menunjukkan bahwa untuk menemukan blok data, seseorang harus menemukan nomornya di blok tidak langsung terlebih dahulu.

Sistem file UNIX biasanya memungkinkan seseorang untuk membuat lubang di file (ini dilakukan dengan mencari() panggilan sistem; periksa halaman manual), yang berarti bahwa sistem file hanya berpura-pura bahwa di tempat tertentu dalam file hanya ada nol byte, tetapi tidak ada sektor disk aktual yang dicadangkan untuk tempat itu dalam file (ini berarti file akan menggunakan lebih sedikit ruang disk). Ini sering terjadi terutama untuk binari kecil, perpustakaan bersama Linux, beberapa database, dan beberapa kasus khusus lainnya. (Lubang diimplementasikan dengan menyimpan nilai khusus sebagai alamat blok data di blok atau inode tidak langsung. Alamat khusus ini berarti tidak ada blok data yang dialokasikan untuk bagian file itu, ergo, ada lubang di file.)

5.10.2. Sistem file berlimpah

Linux mendukung beberapa jenis sistem file. Sampai tulisan ini dibuat, yang terpenting adalah:

minix

Yang tertua, dianggap paling andal, tetapi fiturnya cukup terbatas (beberapa cap waktu hilang, paling banyak 30 karakter nama file) dan terbatas dalam kemampuan (paling banyak 64 MB per sistem file).

xia

Versi modifikasi dari sistem file minix yang menghilangkan batasan pada nama file dan ukuran sistem file, tetapi sebaliknya tidak memperkenalkan fitur baru. Ini tidak terlalu populer, tetapi dilaporkan bekerja dengan sangat baik.

ext3

Sistem file ext3 memiliki semua fitur sistem file ext2. Bedanya, journal sudah ditambahkan. Ini meningkatkan kinerja dan waktu pemulihan jika terjadi kerusakan sistem. Ini telah menjadi lebih populer daripada ext2.

ext2 ext

Yang paling berfitur dari sistem file Linux asli. Ini dirancang agar kompatibel ke atas dengan mudah, sehingga versi baru dari kode sistem file tidak memerlukan pembuatan ulang sistem file yang ada.

reiserfs

Versi ext2 yang lebih lama yang tidak kompatibel ke atas. Ini hampir tidak pernah digunakan dalam instalasi baru lagi, dan kebanyakan orang telah mengonversi ke ext2.

jfs

Sistem file yang lebih kuat. Penjurnalan digunakan yang membuat kehilangan data lebih kecil kemungkinannya. Penjurnalan adalah mekanisme di mana catatan disimpan dari transaksi yang akan dilakukan, atau yang telah dilakukan. Ini memungkinkan sistem file untuk merekonstruksi dirinya sendiri dengan cukup mudah setelah kerusakan yang disebabkan oleh, misalnya, shutdown yang tidak tepat.

xfs

JFS adalah sistem file jurnal yang dirancang oleh IBM untuk bekerja di lingkungan berkinerja tinggi>

XFS awalnya dirancang oleh Silicon Graphics untuk bekerja sebagai sistem file jurnal 64-bit. XFS juga dirancang untuk mempertahankan kinerja tinggi dengan file dan sistem file besar.

Selain itu, dukungan untuk beberapa sistem file asing ada, untuk memudahkan pertukaran file dengan sistem operasi lain. Filesystem asing ini bekerja seperti yang asli, kecuali bahwa mereka mungkin kurang dalam beberapa fitur UNIX biasa, atau memiliki keterbatasan aneh, atau keanehan lainnya.

Panduan Administrator Sistem Linux

Pilihan sistem file yang akan digunakan tergantung pada situasinya. Jika kompatibilitas atau alasan lain membuat salah satu sistem file non-asli diperlukan, maka itu harus digunakan. Jika seseorang dapat memilih dengan bebas, maka mungkin paling bijaksana untuk menggunakan ext3, karena ia memiliki semua fitur ext2, dan merupakan sistem file jurnal. Untuk informasi lebih lanjut tentang sistem file, lihat Bagian 5.10.6. Anda juga dapat membaca Filesystem HOWTO yang terletak di <http://www.tldp.org/HOWTO/Filesystems-HOWTO.html>

Ada juga sistem file proc, biasanya dapat diakses sebagai /proc folder, yang sebenarnya bukan sistem file sama sekali, meskipun terlihat seperti itu. Sistem file proc memudahkan untuk mengakses struktur data kernel tertentu, seperti daftar proses (oleh karena itu namanya). Itu membuat struktur data ini terlihat seperti sistem file, dan sistem file itu dapat dimanipulasi dengan semua alat file biasa. Misalnya, untuk mendapatkan daftar semua proses, seseorang dapat menggunakan perintah

```
$ ls -l /proc
jumlah 0
dr-xr-xr-x 4 akar akar 0 Jan 31 20:37 1
dr-xr-xr-x 4 liw penggun 0 Jan 31 20:37 63
dr-xr-xr-x 4 liw penggun 0 Jan 31 20:37 94
dr-xr-xr-x 4 liw penggun 0 Jan 31 20:37 95
dr-xr-xr-x 4 akar penggun 0 Jan 31 20:37 98
dr-xr-xr-x 4 liw penggun 0 Jan 31 20:37 99
r--r--r-- 1 akar akar 0 Jan 31 20:37 perangkat
r--r--r-- 1 akar akar 0 Jan 31 20:37 dma
r--r--r-- 1 akar akar 0 Jan 31 20:37 sistem file
r--r--r-- 1 akar akar 0 Jan 31 20:37 menyela
r----- 1 akar akar 8654848 Jan 31 20:37 kcore
r--r--r-- 1 akar akar 0 Jan 31 11:50 kmsg
r--r--r-- 1 akar akar 0 Jan 31 20:37 ksyms
r--r--r-- 1 akar akar 0 Jan 31 11:51 beban rata-rata
r--r--r-- 1 akar akar 0 Jan 31 20:37 informasi
r--r--r-- 1 akar akar 0 Jan 31 20:37 modul
dr-xr-xr-x 2 akar akar 0 Jan 31 20:37 bersih
dr-xr-xr-x 4 akar akar 0 Jan 31 20:37 diri sendiri
r--r--r-- 1 akar akar 0 Jan 31 20:37 status
r--r--r-- 1 akar akar 0 Jan 31 20:37 waktu aktif
r--r--r-- 1 akar akar 0 Jan 31 20:37
Versi: kapan
$
```

(Namun, akan ada beberapa file tambahan yang tidak sesuai dengan proses. Contoh di atas adalah disingkat.)

Perhatikan bahwa meskipun disebut sistem file, tidak ada bagian dari sistem file proc yang menyentuh disk apa pun. Itu hanya ada dalam imajinasi kernel. Setiap kali seseorang mencoba untuk melihat bagian mana pun dari sistem file proc, kernel membuatnya tampak seolah-olah bagian itu ada di suatu tempat, meskipun sebenarnya tidak. Jadi, meskipun ada multi-megabyte /proc/kcore file, tidak memakan ruang disk.

5.10.3. Sistem file mana yang harus digunakan?

Biasanya tidak ada gunanya menggunakan banyak sistem file yang berbeda. Saat ini, ext3 adalah sistem file yang paling populer, karena merupakan sistem file jurnal. Saat ini mungkin pilihan yang paling bijaksana. Reiserfs adalah pilihan populer lainnya karena untuk dijurnal. Tergantung pada overhead untuk struktur pembukuan, kecepatan, keandalan (dirasakan), kompatibilitas, dan berbagai alasan lainnya, mungkin disarankan untuk menggunakan sistem file lain. Ini perlu diputuskan berdasarkan kasus per kasus.

Sistem file yang menggunakan penjurnalan juga disebut sistem file terjurnal. Sebuah filesystem journal memelihara log, atau jurnal, dari apa yang telah terjadi pada filesystem. Jika terjadi kerusakan sistem, atau

jika putra Anda yang berusia 2 tahun menekan tombol daya seperti yang saya suka lakukan, sistem file jurnal dirancang untuk menggunakan log sistem file untuk membuat ulang data yang belum disimpan dan hilang. Ini membuat kehilangan data jauh lebih kecil kemungkinannya dan kemungkinan akan menjadi standar fitur dalam sistem file Linux. Namun, jangan sampai ada rasa aman yang salah dari hal ini. Seperti yang lainnya, kesalahan bisa muncul. Selalu pastikan untuk membuat cadangan data Anda jika terjadi keadaan darurat.

Lihat Bagian 5.10.6 untuk detail lebih lanjut tentang fitur dari berbagai jenis sistem file.

5.10.4. Membuat sistem file

Sistem file dibuat, yaitu, diinisialisasi, dengan perintah `mkfs`. Sebenarnya ada program terpisah untuk setiap jenis sistem file. `mkfs` hanyalah ujung depan yang menjalankan program yang sesuai tergantung pada jenis sistem file yang diinginkan. Jenis dipilih dengan `fstype` pilihan.

Program yang dipanggil oleh `mkfs` memiliki antarmuka baris perintah yang sedikit berbeda. Opsi umum dan paling penting dirangkum di bawah ini; lihat halaman manual untuk lebih lanjut.

t `fstype`

Pilih jenis sistem file.

c

Cari blok buruk dan inisialisasi daftar blokir buruk yang sesuai.

l nama file

Baca daftar blokir awal yang buruk dari file nama.

Ada juga banyak program yang ditulis untuk menambahkan opsi spesifik saat membuat sistem file tertentu. Misalnya `mkfs.ext3` menambahkan opsi `b` untuk memungkinkan administrator menentukan ukuran blok apa yang harus digunakan. Pastikan untuk mengetahui apakah ada program khusus yang tersedia untuk jenis sistem file yang ingin Anda gunakan. Untuk informasi lebih lanjut tentang menentukan ukuran blok apa yang akan digunakan, silakan lihat Bagian 5.10.5.

Untuk membuat sistem file `ext2` pada floppy, seseorang akan memberikan perintah berikut:

```
$fdformat di /dev/fd0H1440
Dua sisi, 80 trek, 18 dtk/trek. Kapasitas total
1440KB.
Memformat ... selesai
$ badblocks /dev/fd0H1440 1440 $>$
buruk-blok
$ mkfs.ext2 l bad-blocks
/dev/fd0H1440
mke2fs 0.5a, 5-Apr-94 untuk EXT2 FS 0.5, 94/03/10
360 inode, 1440 blok
72 blok (5,00%) disediakan untuk pengguna super
Blok data pertama = 1
Ukuran blok=1024 (log=0)
Ukuran fragmen=1024
(log=0)
1 grup blok
8192 blok per grup, 8192 fragmen per grup
360 inode per grup

Menulis tabel inode: selesai
Menulis informasi akuntansi superblok dan sistem file:
selesai
```

\$

Pertama, floppy diformat (the `n` opsi mencegah validasi, yaitu, pemeriksaan blok yang buruk). Kemudian blok buruk dicari dengan `badblock`, dengan output diarahkan ke file, `buruk-blok`. Akhirnya, sistem file dibuat, dengan daftar blokir buruk diinisialisasi oleh `badblock` apa pun yang ditemukan.

Itu `c` opsi dapat digunakan dengan `mkfs` alih-alih `badblock` dan file terpisah. Contoh di bawah ini melakukan itu.

```
$ mkfs.ext2 c
/dev/fd0H1440
mke2fs 0.5a, 5-Apr-94 untuk EXT2 FS 0.5, 94/03/10
360 inode, 1440 blok
72 blok (5,00%) disediakan untuk pengguna super
Blok data pertama = 1
Ukuran blok=1024 (log=0)
Ukuran fragmen=1024
(log=0)
1 grup blok
8192 blok per grup, 8192 fragmen per grup
360 inode per grup

Memeriksa blok buruk (tes baca-saja): selesai
Menulis tabel inode: selesai
Menulis informasi akuntansi superblok dan sistem file:
selesai
$
```

Itu `c` opsi lebih nyaman daripada penggunaan `badblock` secara terpisah, tetapi `badblock` diperlukan untuk memeriksa setelah sistem file dibuat.

Proses untuk menyiapkan sistem file pada hard disk atau partisi sama dengan untuk disket, kecuali bahwa pemformatan tidak diperlukan.

5.10.5. Ukuran blok sistem file

Ukuran blok menentukan ukuran yang akan digunakan sistem file untuk membaca dan menulis data. Ukuran blok yang lebih besar akan membantu meningkatkan kinerja I/O disk saat menggunakan file besar, seperti database. Hal ini terjadi karena disk dapat membaca atau menulis data untuk jangka waktu yang lebih lama sebelum harus mencari blok berikutnya.

Pada sisi negatifnya, jika Anda akan memiliki banyak file yang lebih kecil di sistem file itu, seperti `/etc`, ada potensi banyak ruang disk yang terbuang.

Misalnya, jika Anda menyetel ukuran blok ke 4096, atau 4K, dan Anda membuat file berukuran 256 byte, file tersebut masih akan menghabiskan 4K ruang di hard drive Anda. Untuk satu file yang mungkin tampak sepele, tetapi ketika sistem file Anda berisi ratusan atau ribuan file, ini bisa bertambah.

Ukuran blok juga dapat mempengaruhi ukuran file maksimum yang didukung pada beberapa sistem file. Hal ini karena banyak

sistem file modern tidak dibatasi oleh ukuran blok atau ukuran file, tetapi oleh jumlah blok. Oleh karena itu, Anda akan menggunakan rumus "ukuran blok * maks # blok = ukuran blok maks".

5.10.6. Perbandingan sistem file

Tabel 5–1. Membandingkan Fitur Sistem File

Panduan Administrator Sistem Linux

Nama FS	Tahun	OS asli	Ukuran File	Ukuran FS	Jurnal
FAT16	1983	MSDOS V2	4GB	16MB hingga	n
FAT32	1997	Jendela 95	4GB	8GB hingga 2TB	n
HPFS	1988	OS/2	4GB	2TB	n
NTFS	1993	Windows NT	16EB	16EB	kamu
HFS+	1998	Mac OS	8EB	?	n
UFS2	2002	GratisBSD	512GB hingga	1YB	n
ext2	1993	Linux	16GB hingga	2TB hingga	n
ext3	1999	Linux	16GB hingga	2TB hingga	kamu
ReiserFS3	2001	Linux	8TB8	16TB	kamu
ReiserFS4	2005	Linux	?	?	kamu
XFS	1994	IRIX	9EB	9EB	kamu
JFS	?	AIX	8EB	512TB hingga	kamu
VxFS	1991	SVR4.0	16EB	?	kamu
ZFS	2004	Solaris 10	1YB	16EB	n

Legenda

Tabel 5–2. ukuran

Kilobyte KB	1024 Byte
Megabita MB	1024 KB
Gigabyte GB	1024 MBs
Terabyte TB	1024 GB
Petabyte PB	1024 TB
Exabyte EB	1024 PB
Zettabyte ZB	1024 EB
Yottabyte YB	1024 ZB

Perlu dicatat bahwa Exabytes, Zettabytes, dan Yottabytes jarang ditemukan, jika pernah. Ada perkiraan saat ini bahwa bahan cetak dunia sama dengan 5 Exabytes. Oleh karena itu, beberapa sistem file ini keterbatasan dianggap oleh banyak orang sebagai teoritis. Namun, perangkat lunak sistem file telah ditulis dengan kemampuan ini.

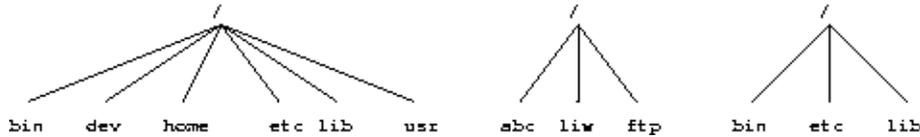
Untuk informasi lebih lengkap Anda dapat mengunjungi http://en.wikipedia.org/wiki/Comparison_of_file_systems.

5.10.7. Pemasangan dan pelepasan

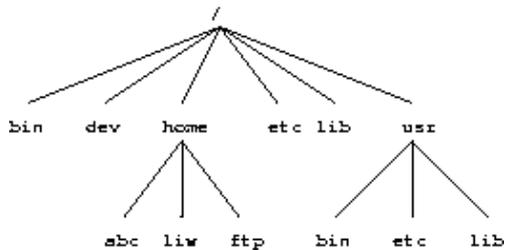
Sebelum seseorang dapat menggunakan sistem file, itu harus di-mount. Sistem operasi kemudian melakukan berbagai hal pembukuan untuk memastikan semuanya berfungsi. Karena semua file di UNIX berada dalam satu pohon folder, operasi mount akan membuatnya tampak seperti konten sistem file baru adalah konten dari subfolder yang ada di beberapa sistem file yang sudah terpasang.

Misalnya, [Gambar 5-3](#) menunjukkan tiga sistem file terpisah, masing-masing dengan folder root sendiri. Ketika dua sistem file terakhir dipasang di bawah `/home` dan `/usr`, masing-masing, pada sistem file pertama, kita bisa mendapatkan pohon folder tunggal, seperti pada [Gambar 5-4](#).

Gambar 5-3. Tiga sistem file terpisah.



Gambar 5-4. `/home` dan `/usr` telah dipasang.



Pemasangan dapat dilakukan seperti pada contoh berikut:

```

$ mount /dev/hda2 /home
$ mount /dev/hda3 /usr
$
    
```

Perintah `mount` membutuhkan dua argumen. Yang pertama adalah file perangkat yang sesuai dengan disk atau partisi yang berisi sistem file. Yang kedua adalah folder di bawahnya yang akan dipasang. Setelah perintah-perintah ini, isi dari dua sistem file terlihat seperti isi dari `/home` dan `/usr` folder, masing-masing. Seseorang kemudian akan mengatakan */dev/hda2 dipasang pada /home* ", dan juga untuk `/usr`. Untuk melihat salah satu sistem file, orang akan melihat isi folder tempat file tersebut berada di-mount, seperti halnya folder lain. Perhatikan perbedaan antara file perangkat, `/dev/hda2`, dan folder terpasang, `/home` . File perangkat memberikan akses ke konten mentah disk, folder mount-on memberikan akses ke file pada disk. Folder mount-on disebut *mount titik*.

Linux mendukung banyak jenis sistem file. `mount` mencoba menebak jenis sistem file. Anda juga dapat menggunakan

`t fstype` opsi untuk menentukan jenis secara langsung; ini terkadang diperlukan, karena penggunaan `mount` heuristik tidak selalu berhasil. Misalnya, untuk memasang floppy MS-DOS, Anda dapat menggunakan perintah berikut:

```

$ pasang di msdos /dev/fd0
    /floppy
$
    
```

Folder mount-on tidak perlu kosong, meskipun harus ada. Namun, file apa pun di dalamnya tidak akan dapat diakses berdasarkan nama saat sistem file dipasang. (Berkas apa pun yang telah dibuka akan tetap dapat diakses. Berkas yang memiliki tautan keras dari folder lain dapat diakses menggunakan nama tersebut.) Tidak ada salahnya melakukan ini, dan bahkan dapat bermanfaat. Misalnya, beberapa orang suka memiliki `/tmp` dan `/var/tmp` sinonim, dan membuat `/tmp` menjadi tautan simbolis ke `/var/tmp`. Saat sistem di-boot, sebelum `/var` sistem file sudah terpasang, `/var/tmp` folder yang berada pada sistem file root digunakan sebagai gantinya. Kapan `/var` dipasang, itu akan membuat `/var/tmp` folder pada sistem

Panduan Administrator Sistem Linux

file root tidak dapat diakses. Jika `/var/tmp` tidak ada di sistem file root, tidak mungkin menggunakan file sementara sebelum dipasang `/var`.

Jika Anda tidak bermaksud untuk menulis apa pun ke sistem file, gunakan `r` beralih ke mount untuk melakukan mount read-only. Ini akan membuat kernel menghentikan setiap upaya penulisan ke sistem file, dan juga akan menghentikan kernel dari memperbarui waktu akses file di inode. Mount read-only diperlukan untuk media yang tidak dapat ditulis, misalnya CD-ROM.

Pembaca yang waspada telah melihat sedikit masalah logistik. Bagaimana sistem file pertama (disebut sistem file root, karena berisi folder root) dipasang, karena jelas tidak dapat dipasang pada sistem file lain? Nah, jawabannya adalah bahwa hal itu dilakukan dengan sihir. Sistem file root secara ajaib dipasang saat boot waktu, dan seseorang dapat mengandalkannya untuk selalu dipasang. Jika sistem file root tidak dapat di-mount, sistem tidak bisa boot. Nama sistem file yang secara ajaib dipasang sebagai root dikompilasi ke dalam kernel, atau disetel menggunakan LILO atau rdev.

Untuk informasi lebih lanjut, lihat sumber kernel atau Panduan Peretas Kernel.

Sistem file root biasanya pertama kali dipasang hanya-baca. Skrip startup kemudian akan menjalankan fsck untuk memverifikasi validitasnya, dan jika tidak ada masalah, mereka akan memasang ulang sehingga penulisan juga akan diizinkan. fsck tidak boleh dijalankan pada sistem file yang di-mount, karena setiap perubahan pada sistem file saat fsck sedang berjalan akan menyebabkan masalah. Karena sistem file root dipasang hanya-baca saat sedang diperiksa, fsck dapat memperbaiki masalah apa pun tanpa khawatir, karena operasi remount akan menghapus metadata apa pun yang disimpan sistem file di memori.

Pada banyak sistem ada sistem file lain yang juga harus dipasang secara otomatis pada saat boot. Ini ditentukan dalam `/etc/fstab` mengajukan; lihat halaman manual `fstab` untuk detail tentang formatnya. Detail kapan tepatnya sistem file tambahan dipasang bergantung pada banyak faktor, dan dapat dikonfigurasi oleh setiap administrator jika perlu; lihat Bab 8.

Ketika sistem file tidak lagi perlu dipasang, itu dapat dilepas dengan `umount`. `umount` mengambil satu argumen: baik file perangkat atau titik mount. Misalnya, untuk meng-unmount folder dari contoh sebelumnya, seseorang dapat menggunakan perintah

```
$ umount /dev/hda2
$ umount /usr
$
```

Lihat halaman manual untuk instruksi lebih lanjut tentang cara menggunakan perintah. Sangat penting bahwa Anda selalu meng-unmount floppy terpasang. Jangan hanya mengeluarkan floppy dari drive! Karena caching disk, data tidak harus ditulis ke floppy sampai Anda melepaskannya, jadi menghapus floppy dari drive terlalu dini mungkin

menyebabkan isinya menjadi kacau. Jika Anda hanya membaca dari disket, kemungkinannya kecil, tetapi jika Anda menulis, bahkan secara tidak sengaja, hasilnya bisa menjadi bencana besar.

Pemasangan dan pelepasan memerlukan hak pengguna super, yaitu, hanya root yang dapat melakukannya. Alasan untuk ini adalah bahwa jika ada pengguna yang dapat memasang floppy di folder mana pun, maka akan lebih mudah untuk membuat floppy dengan, katakanlah, kuda Trojan yang disamarkan sebagai `/bin/sh`, atau program lain yang sering digunakan. Namun, sering kali diperlukan untuk mengizinkan pengguna menggunakan disket, dan ada beberapa cara untuk melakukannya:

- Berikan kata sandi root kepada pengguna. Ini jelas merupakan keamanan yang buruk, tetapi merupakan solusi termudah. Ini bekerja dengan baik jika

tidak ada kebutuhan untuk keamanan, seperti yang terjadi pada banyak sistem pribadi non-jaringan.

- Gunakan program seperti sudo untuk mengizinkan pengguna menggunakan mount. Ini masih merupakan keamanan yang buruk, tetapi tidak secara langsung memberikan hak pengguna super kepada semua orang. Ini membutuhkan beberapa detik pemikiran keras pada pengguna kepentingan. Selanjutnya sudo dapat dikonfigurasi untuk hanya mengizinkan pengguna menjalankan perintah tertentu. Lihat halaman manual sudo(8), sudoers(5), dan visudo(8).
- Membuat pengguna menggunakan mtools, sebuah paket untuk memanipulasi sistem file MS-DOS, tanpa memasangnya. Ini bekerja dengan baik jika hanya disket MS-DOS yang diperlukan, tetapi sebaliknya agak canggung.
- Buat daftar perangkat floppy dan titik pemasangan yang diizinkan bersama dengan opsi yang sesuai di /etc/fstab.

Alternatif terakhir dapat diimplementasikan dengan menambahkan baris seperti berikut ke/etc/fstabmengajukan:

```
/dev/fd0 /pengguna floppy msdos,noauto 0 0
```

Kolomnya adalah: file perangkat untuk dipasang, folder untuk dipasang, jenis sistem file, opsi, frekuensi pencadangan (digunakan oleh dump), dan nomor pass fsck (untuk menentukan urutan sistem file yang harus diperiksa saat boot; 0 berarti tidak ada pemeriksaan) .

Itu tidak otomatis opsi menghentikan pemasangan ini untuk dilakukan secara otomatis ketika sistem dimulai (yaitu, berhenti memasang a dari pemasangannya). Itupengguna opsi memungkinkan setiap pengguna untuk memasang sistem file, dan, karena keamanan alasan, melarang eksekusi program (normal atau setuid) dan interpretasi file perangkat dari sistem file terpasang. Setelah ini, setiap pengguna dapat memasang floppy dengan sistem file msdos dengan perintah berikut:

```
$ mount / floppy
$
```

Floppy dapat (dan perlu, tentu saja) di-unmount dengan perintah umount yang sesuai.

Jika Anda ingin memberikan akses ke beberapa jenis floppy, Anda perlu memberikan beberapa mount point. Pengaturan dapat berbeda untuk setiap titik pemasangan. Misalnya, untuk memberikan akses ke disket MS-DOS dan ext2, Anda dapat memiliki baris berikut di:/etc/fstab:

```
/dev/fd0 /mnt/dosfloppy msdos pengguna, 0 0
/dev/fd0 /mnt/ext2floppy ext2 pengguna, 0 0
noauto
```

Alternatifnya adalah dengan menambahkan satu baris yang mirip dengan berikut ini:

```
/dev/fd0 /mnt/floppy pengguna otomatis,noauto 0 0
```

Opsi "otomatis" di kolom jenis sistem file memungkinkan perintah mount untuk menanyakan sistem file dan mencoba menentukan jenisnya sendiri. Opsi ini tidak akan berfungsi pada semua jenis sistem file, tetapi berfungsi dengan baik pada yang lebih umum.

Untuk sistem file MS-DOS (bukan hanya disket), Anda mungkin ingin membatasi aksesnya dengan menggunakan: `uid`, `gid`, dan `umask` opsi sistem file, dijelaskan secara rinci di halaman manual `mount`. Jika Anda tidak berhati-hati, memasang sistem file MS-DOS memberi setiap orang setidaknya akses baca ke file di dalamnya, yang bukan ide yang baik.

5.10.8. Keamanan Sistem File

UNTUK DITAMBAHKAN

Bagian ini akan menjelaskan opsi pemasangan dan cara menggunakannya di `/etc/fstab` untuk memberikan keamanan sistem tambahan.

5.10.9. Memeriksa integritas sistem file dengan `fsck`

Sistem file adalah makhluk yang kompleks, dan karena itu, mereka cenderung agak rawan kesalahan. Kebenaran dan validitas sistem file dapat diperiksa menggunakan perintah `fsck`. Itu dapat diinstruksikan untuk memperbaiki masalah kecil yang ditemukannya, dan untuk memperingatkan pengguna jika ada masalah yang tidak dapat diperbaiki. Untungnya, kode untuk mengimplementasikan sistem file di-debug dengan cukup efektif, jadi jarang ada masalah sama sekali, dan biasanya disebabkan oleh kegagalan daya, kegagalan perangkat keras, atau kesalahan operator; misalnya, dengan tidak mematikan sistem dengan benar.

Sebagian besar sistem diatur untuk menjalankan `fsck` secara otomatis saat boot, sehingga setiap kesalahan terdeteksi (dan mudah-mudahan diperbaiki) sebelum sistem digunakan. Penggunaan sistem file yang rusak cenderung memperburuk keadaan: jika struktur data kacau, menggunakan sistem file mungkin akan lebih mengacaukannya, mengakibatkan lebih banyak kehilangan data. Namun, `fsck` dapat memakan waktu cukup lama untuk dijalankan pada sistem file besar, dan karena kesalahan hampir tidak pernah terjadi jika sistem telah dimatikan dengan benar, beberapa trik digunakan untuk menghindari pemeriksaan dalam kasus tersebut. Yang pertama adalah jika file `/etc/fastboot` ada, tidak ada pemeriksaan yang dilakukan. Yang kedua adalah bahwa sistem file `ext2` memiliki penanda khusus di superbloknya yang memberi tahu apakah sistem file telah dilepas dengan benar setelah pemasangan sebelumnya. Ini memungkinkan `e2fsck` (versi `fsck` untuk sistem file `ext2`) untuk menghindari pemeriksaan sistem file jika flag menunjukkan bahwa `umount` telah dilakukan (asumsi bahwa `umount` yang tepat menunjukkan tidak ada masalah). Apakah `/etc/fastboot` trik bekerja di sistem Anda tergantung pada skrip startup Anda, tetapi trik `ext2` berfungsi setiap kali Anda menggunakan `e2fsck`. Itu harus dilewati secara eksplisit dengan opsi ke `e2fsck` untuk dihindari. (Lihat halaman manual `e2fsck` untuk detail tentang caranya.)

Pemeriksaan otomatis hanya berfungsi untuk sistem file yang dipasang secara otomatis saat boot. Gunakan `fsck` secara manual untuk memeriksa sistem file lain, misalnya, disket.

Jika `fsck` menemukan masalah yang tidak dapat diperbaiki, Anda memerlukan pengetahuan mendalam tentang cara kerja sistem file secara umum, dan jenis sistem file yang rusak pada khususnya, atau pencadangan yang baik. Yang terakhir ini mudah (walaupun terkadang membosankan) untuk diatur, yang pertama terkadang dapat diatur melalui teman, newsgroup dan milis Linux, atau sumber dukungan lain, jika Anda sendiri tidak memiliki pengetahuan. Saya ingin memberi tahu Anda lebih banyak tentang itu, tetapi kurangnya pendidikan dan pengalaman saya dalam hal ini menghalangi saya. Program `debugfs` oleh Theodore Ts'o seharusnya berguna.

`fsck` hanya boleh dijalankan pada sistem file yang tidak di-mount, tidak pernah pada sistem file yang di-mount (dengan pengecualian `root read-only` saat startup). Ini karena ia mengakses disk mentah, dan karena itu dapat memodifikasi sistem file tanpa disadari oleh sistem operasi. Akan ada masalah, jika sistem operasinya bingung.

5.10.10. Memeriksa kesalahan disk dengan badblock

Sebaiknya periksa blok buruk secara berkala. Ini dilakukan dengan perintah badblocks. Ini menampilkan daftar nomor semua blok buruk yang dapat ditemukan. Daftar ini dapat diumpankan ke fsck untuk direkam dalam struktur data sistem file sehingga sistem operasi tidak akan mencoba menggunakan blok buruk untuk menyimpan data. Contoh berikut akan menunjukkan bagaimana hal ini dapat dilakukan.

```
$ badblocks /dev/fd0H1440 1440 >
      buruk-blok
$ fsck t ext2 l bad-blocks
      /dev/fd0H1440
Paralelisasi fsck versi 0.5a (5-Apr-94)
e2fsck 0.5a, 5-Apr-94 untuk EXT2 FS 0.5, 94/03/10
Pass 1: Memeriksa inode, blok, dan ukuran
Lulus 2: Memeriksa struktur folder
Lulus 3: Memeriksa konektivitas folder
Lulus 4: Memeriksa jumlah referensi.
Lulus 5: Memeriksa informasi ringkasan grup.

/dev/fd0H1440: ***** SISTEM FILE SUDAH DIUBAH *****
/dev/fd0H1440: 11/360 file, 63/1440 blok
$
```

Jika badblock melaporkan blok yang sudah digunakan, e2fsck akan mencoba memindahkan blok ke tempat lain. Jika blok itu benar-benar buruk, bukan hanya marginal, isi file mungkin rusak.

5.10.11. Melawan fragmentasi?

Ketika sebuah file ditulis ke disk, itu tidak selalu dapat ditulis dalam blok yang berurutan. File yang tidak disimpan dalam blok berurutan akan terfragmentasi. Diperlukan waktu lebih lama untuk membaca file yang terfragmentasi, karena kepala baca-tulis disk harus bergerak lebih banyak. Diinginkan untuk menghindari fragmentasi, meskipun ini bukan masalah dalam sistem dengan buffer cache yang baik dengan read-ahead.

Sistem file Linux modern menjaga fragmentasi seminimal mungkin dengan menjaga semua blok dalam file berdekatan, bahkan jika mereka tidak dapat disimpan di sektor yang berurutan. Beberapa sistem file, seperti ext3, secara efektif mengalokasikan blok gratis yang paling dekat dengan blok lain dalam sebuah file. Oleh karena itu tidak perlu khawatir tentang fragmentasi dalam sistem Linux.

Pada hari-hari awal sistem file ext2, ada kekhawatiran atas fragmentasi file yang mengarah pada pengembangan program defragmentasi yang disebut, defrag. Salinannya masih dapat diunduh di <http://www.go.dlr.de/linux/src/defrag-0.73.tar.gz>. Namun, SANGAT disarankan agar Anda TIDAK menggunakan

dia. Itu dirancang untuk dan versi ext2 yang lebih lama, dan belum diperbarui sejak tahun 1998! Saya hanya menyebutkannya di sini untuk tujuan referensi.

Ada banyak program defragmentasi MS-DOS yang memindahkan blok-blok di dalam sistem file untuk menghapus fragmentasi. Untuk sistem file lain, defragmentasi harus dilakukan dengan mencadangkan sistem file, membuat ulang, dan memulihkan file dari cadangan. Mencadangkan sistem file sebelum defragmentasi adalah ide yang bagus untuk semua sistem file, karena banyak hal bisa salah selama defragmentasi.

5.10.12. Alat lain untuk semua sistem file

Beberapa alat lain juga berguna untuk mengelola sistem file. `df` menunjukkan ruang disk kosong pada satu atau lebih sistem file; `du` menunjukkan berapa banyak ruang disk folder dan semua filenya. Ini dapat digunakan untuk memburu pemboros ruang disk. Keduanya memiliki halaman manual yang merinci (banyak) opsi yang dapat digunakan.

sinkronisasi memaksa semua blok tidak tertulis dalam buffer cache (lihat Bagian 6.6) untuk ditulis ke disk. Jarang diperlukan untuk melakukan ini dengan tangan; pembaruan proses daemon melakukan ini secara otomatis. Ini dapat berguna dalam bencana, misalnya jika pembaruan atau proses pembantunya `bdflush` mati, atau jika Anda harus mematikan daya sekarang dan tidak bisa menunggu pembaruan berjalan. Sekali lagi, ada halaman manual. Pria itu adalah teman terbaik Anda di Linux. Apropos sepuhnya juga sangat berguna ketika Anda tidak tahu apa nama perintah yang Anda inginkan.

5.10.13. Alat lain untuk sistem file ext2/ext3

Selain pembuat sistem file (`mke2fs`) dan pemeriksa (`e2fsck`) yang dapat diakses secara langsung atau melalui ujung depan independen jenis sistem file, sistem file ext2 memiliki beberapa alat tambahan yang dapat berguna.

tune2fs menyesuaikan parameter sistem file. Beberapa parameter yang lebih menarik adalah:

- Jumlah pemasangan maksimal. `e2fsck` memberlakukan pemeriksaan ketika sistem file telah dipasang terlalu sering, bahkan jika bendera bersih diatur. Untuk sistem yang digunakan untuk mengembangkan atau menguji sistem, mungkin:
ide yang baik untuk mengurangi batas ini.
- Waktu maksimal antara pemeriksaan. `e2fsck` juga dapat menerapkan waktu maksimal antara dua pemeriksaan, bahkan jika bendera bersih diatur, dan sistem file tidak terlalu sering dipasang. Ini dapat dinonaktifkan, Namun.
- Jumlah blok yang dicadangkan untuk root. Ext2 mencadangkan beberapa blok untuk root sehingga jika sistem file penuh, masih dimungkinkan untuk melakukan administrasi sistem tanpa harus menghapus apa pun. Jumlah yang dicadangkan secara default adalah 5 persen, yang pada sebagian besar disk tidak cukup untuk menjadi pemborosan. Namun, untuk disket tidak adanya memesan blok apa pun.

Lihat halaman manual `tune2fs` untuk informasi lebih lanjut.

dumpe2fs menampilkan informasi tentang sistem file ext2 atau ext3, sebagian besar dari superblok. Di bawah ini adalah contoh keluaran. Beberapa informasi dalam output bersifat teknis dan memerlukan pemahaman tentang cara kerja sistem file, tetapi sebagian besar mudah dipahami bahkan oleh admin awam.

```
# dumpe2fs
dumpe2fs 1.32 (09-Nov-2002)
Nama volume sistem file: /
Terakhir dipasang pada: tidak tersedia
UUID Sistem File: 51603f82-68f3-4ae7-a755-b777ff9dc739
Nomor ajaib sistem file: 0xEF53
Revisi sistem file #: 1 (dinamis)
```

Panduan Administrator Sistem Linux

```
Fitur sistem file: has_journal filetype needs_recovery sparse_super
```

```
Opsi pemasangan default: (tidak
ada) Status sistem file: bersih
Perilaku kesalahan: Lanjutkan
Jenis OS sistem file: Linux Inode
count: 3482976
Jumlah blok: 6960153
Jumlah blok yang dipesan: 348007
Blok gratis: 3873525
Inode gratis: 3136573
Blok pertama: 0
Ukuran blok: 4096
Ukuran fragmen: 4096
Blok per grup: 32768
Fragmen per grup: 32768
Inode per grup: 16352
Blok inode per grup: 511
Sistem file yang dibuat: Sel 26 Agustus 2003
Waktu pemasangan Senin 22 Des 2003
Waktu penulisan Senin 22 Des 2003
Jumlah pemasangan: 3
Jumlah pemasangan 1
Terakhir diperiksa: Senin 3 November 2003
Periksa interval: 0 (tidak ada)
Uid blok yang 0 (akar pengguna)
Blok yang dicadangkan 0 (akar grup)
inode pertama: 11
Ukuran inode: 128
UUID Jurnal: tidak ada
inode jurnal: 8
Perangkat jurnal: 0x0000
Inode yatim piatu 655612
```

Grup 0: (Blok 0-32767)

Superblok utama pada 0, Deskriptor grup pada 1-2

Blok bitmap di 3 (+3), Inode bitmap di 4 (+4)

Blok bitmap di 3 (+3), Inode bitmap di 4 (+4)

Tabel inode di 5-515 (+5)

3734 blok gratis, 16338 inode gratis, 2 folder

debugfs adalah debugger sistem file. Ini memungkinkan akses langsung ke struktur data sistem file yang disimpan di disk dan dengan demikian dapat digunakan untuk memperbaiki disk yang sangat rusak sehingga fsck tidak dapat memperbaikinya secara otomatis. Itu juga telah dikenal digunakan untuk memulihkan file yang dihapus. Namun, debugfs sangat mengharuskan Anda memahami apa yang Anda lakukan; kegagalan untuk memahami dapat menghancurkan semua data Anda.

membuang dan **restore** dapat digunakan untuk membuat cadangan filesystem ext2. Mereka adalah versi khusus ext2 dari tradisional alat cadangan UNIX. Lihat [Bagian 12.1](#) untuk informasi lebih lanjut tentang pencadangan.

5.11. Disk tanpa sistem file

Tidak semua disk atau partisi digunakan sebagai sistem file. Partisi swap, misalnya, tidak akan memiliki sistem file di dalamnya. Banyak disket digunakan dengan cara meniru drive-tape, sehingga tar (arsip kaset) atau file lain ditulis langsung pada disk mentah, tanpa sistem file. Disket boot Linux tidak berisi sistem file, hanya kernel mentah.

Menghindari sistem file memiliki keuntungan membuat lebih banyak disk dapat digunakan, karena sistem file selalu memiliki beberapa overhead pembukuan. Ini juga membuat disk lebih mudah kompatibel dengan

Bab 5. Menggunakan Disk dan Media

Panduan Administrator Sistem Linux

sistem lain: misalnya, format file tar sama di semua sistem, sedangkan sistem file berbeda di kebanyakan sistem. Anda akan segera terbiasa dengan disk tanpa sistem file jika Anda membutuhkannya. Floppy Linux yang dapat di-boot juga tidak harus memiliki sistem file, meskipun mungkin.

Salah satu alasan menggunakan disk mentah adalah untuk membuat salinan gambarnya. Misalnya, jika disk berisi sistem file yang rusak sebagian, merupakan ide yang baik untuk membuat salinan persisnya sebelum mencoba memperbaikinya, karena Anda dapat memulai lagi jika perbaikan Anda lebih merusak. Salah satu cara untuk melakukan ini adalah dengan menggunakan `dd`:

```
$ dd if=/dev/fd0H1440
    dari=floppy-ima
    ge
2880+0 catatan dalam
    2880+0 rekaman keluar
$ dd if=floppy-gambar
    dari=/dev/fd0H14
    40
2880+0 catatan dalam
    2880+0 rekaman keluar
$
```

`dd` pertama membuat gambar yang tepat dari floppy ke file `floppygambar`, yang kedua menulis gambar ke floppy. (Pengguna mungkin telah mengganti floppy sebelum perintah kedua. Jika tidak, pasangan perintah diragukan kegunaannya.)

5.12. Mengalokasikan ruang disk

5.12.1. Skema partisi

Ketika datang untuk mempartisi mesin Anda, tidak ada cara yang benar secara universal untuk melakukannya. Ada banyak faktor yang harus diperhitungkan tergantung pada tujuan mesin.

Untuk workstation sederhana dengan ruang disk terbatas, seperti laptop, Anda mungkin memiliki sedikitnya 3 partisi. SEBUAH partisi untuk `/`, `/boot`, dan bertukar. Namun, bagi sebagian besar pengguna, ini bukan solusi yang disarankan.

Cara tradisional adalah memiliki sistem file root (relatif) kecil, dan memisahkan partisi untuk sistem file seperti `/usr` dan `/home`. Membuat sistem file root terpisah jika sistem file root kecil dan tidak terlalu sering digunakan, kecil kemungkinannya menjadi rusak saat sistem mogok, dan oleh karena itu mempermudah pemulihan sistem yang mogok. Alasannya adalah untuk mencegah sistem file root terisi dan menyebabkan sistem crash.

Saat membuat skema partisi Anda, ada beberapa hal yang perlu Anda ingat. Anda tidak dapat membuat partisi terpisah untuk folder berikut: `/bin`, `/etc`, `/dev`, `/initrd`, `/lib`, dan `/sbin`. Isi folder ini diperlukan saat boot dan harus selalu menjadi bagian dari `/` partisi.

Anda juga disarankan untuk membuat partisi terpisah untuk `/var` dan `/tmp`. Ini karena kedua folder biasanya memiliki data yang terus berubah. Tidak membuat partisi terpisah untuk sistem file ini membuat Anda berisiko memiliki file log yang mengisi `/` partisi.

Contoh partisi server adalah:

```
Ukuran Sistem File yang Digunakan Tersedia Penggunaan% Dipasang di
/dev/hda2 9.7G 1.3G 8.0G 14% /
/dev/hda1 128M 44M 82M 34% /boot
```

```
/dev/hda3 4.9G 4.0G 670M 86% /usr  
/dev/hda5 4.9G 2.1G 2.5G 46% /var  
/dev/hda7 31G 24G 5.6G 81% /home  
/dev/hda8 4.9G 2.0G 670M 43% /opt
```

Masalah dengan memiliki banyak partisi adalah membagi jumlah total ruang disk kosong menjadi banyak bagian kecil. Salah satu cara untuk menghindari masalah ini adalah dengan menggunakan `lvm` untuk membuat Volume Logis.

5.12.2. Manajer Volume Logis (LVM)

Menggunakan LVM memungkinkan administrator fleksibilitas untuk membuat disk logis yang dapat diperluas secara dinamis karena lebih banyak ruang disk diperlukan.

Ini dilakukan terlebih dahulu dengan membuat partisi dengan sebagai tipe partisi LVM Linux `0x8e`. Kemudian Partisi Fisik ditambahkan ke Grup Volume dan dipecah menjadi beberapa bagian, atau Grup Volume Tingkat Fisik. Ekstensi ini kemudian dapat dikelompokkan ke dalam Volume Logis. Volume Logis ini kemudian dapat diformat seperti partisi fisik. Perbedaan besar adalah bahwa mereka dapat diperluas dengan menambahkan lebih banyak luasan kepada mereka.

Saat ini, diskusi lengkap tentang LVM berada di luar cakupan panduan ini. Namun, dan sumber yang bagus untuk mempelajari lebih lanjut tentang LVM dapat ditemukan di <http://www.tldp.org/HOWTO/LVM-HOWTO.html>.

5.12.3. Persyaratan ruang

Distribusi Linux yang Anda instal akan memberikan beberapa indikasi tentang berapa banyak ruang disk yang Anda butuhkan untuk berbagai konfigurasi. Program yang diinstal secara terpisah juga dapat melakukan hal yang sama. Ini akan membantu Anda merencanakan penggunaan ruang disk Anda, tetapi Anda harus bersiap untuk masa depan dan mencadangkan beberapa ruang ekstra untuk hal-hal yang akan Anda perhatikan nanti yang Anda perlukan.

Jumlah yang Anda butuhkan untuk file pengguna tergantung pada apa yang ingin dilakukan pengguna Anda. Kebanyakan orang tampaknya membutuhkan ruang sebanyak mungkin untuk file mereka, tetapi jumlah yang mereka akan hidup bahagia dengan sangat bervariasi. Beberapa orang hanya melakukan pemrosesan teks ringan dan akan bertahan dengan baik dengan beberapa megabyte, yang lain melakukan pemrosesan gambar yang berat dan akan membutuhkan gigabyte.

Omong-omong, ketika membandingkan ukuran file yang diberikan dalam kilobyte atau megabita dan ruang disk yang diberikan dalam megabita, penting untuk mengetahui bahwa kedua unit tersebut dapat berbeda. Beberapa produsen disk suka berpura-pura bahwa satu kilobyte adalah 1000 byte dan satu megabyte adalah 1000 kilobyte, sementara semua dunia komputasi lainnya menggunakan 1024 untuk kedua faktor tersebut. Oleh karena itu, hard disk 345 MB sebenarnya adalah hard disk 330 MB.

Alokasi ruang swap dibahas dalam [Bagian 6.5](#).

5.12.4. Contoh alokasi hard disk

Dulu saya punya hardisk 10 GB. Sekarang saya menggunakan hard disk 30 GB. Saya akan menjelaskan bagaimana dan mengapa saya mempartisi disk tersebut.

Panduan Administrator Sistem Linux

Pertama, saya membuat `/boot` partisi pada 128 MG. Ini lebih besar dari yang saya perlukan, dan cukup besar untuk memberi saya ruang jika saya membutuhkannya. Saya membuat yang terpisah `/boot` partisi untuk memastikan bahwa sistem file ini tidak akan pernah terisi, dan karenanya dapat di-boot. Lalu saya membuat 5 GB `/var` partisi. Sejak `/var` sistem file adalah tempat file log dan email disimpan. Saya ingin mengisolasinya dari partisi root saya. (Saya memiliki file log yang tumbuh dalam semalam dan mengisi sistem file root saya di masa lalu.) Selanjutnya, saya membuat 15 GB `/home` partisi. Ini berguna jika terjadi kerusakan sistem. Jika saya harus menginstal ulang Linux dari awal, saya dapat memberi tahu program penginstalan untuk tidak memformat partisi ini, dan sebaliknya memasang ulang tanpa kehilangan data. Akhirnya karena saya memiliki 512 MG RAM, saya membuat partisi swap 1024 MG (atau 1 GB). Ini meninggalkan saya dengan kira-kira sistem file root 9 GB. Saya menggunakan hard drive 10 GB lama saya, saya membuat 8 GB `/usr` partisi dan meninggalkan 2 GB tidak terpakai. Ini memetikan saya membutuhkan lebih banyak ruang di masa depan.

Pada akhirnya, tabel partisi saya terlihat seperti ini:

Tabel 5–3. Partisi Saya

9 GB	sistem file root
1 GB	tukar partisi
5 GB	<code>/var</code> berkas
15 GB	<code>/home</code> berkas
8 GB	<code>/usr</code> berkas
2 GB	partisi awal

5.12.5. Menambahkan lebih banyak ruang disk untuk Linux

Menambahkan lebih banyak ruang disk untuk Linux itu mudah, setidaknya setelah perangkat keras terpasang dengan benar (penginstalan perangkat keras berada di luar cakupan buku ini). Anda memformatnya jika perlu, lalu buat partisi dan sistem file seperti yang dijelaskan di atas, dan tambahkan baris yang tepat ke `/etc/fstab` sehingga terpasang secara otomatis.

5.12.6. Kiat untuk menghemat ruang disk

Tip terbaik untuk menghemat ruang disk adalah menghindari menginstal program yang tidak perlu. Sebagian besar distribusi Linux memiliki opsi untuk menginstal hanya sebagian dari paket yang dikandungnya, dan dengan menganalisis kebutuhan Anda, Anda mungkin menyadari bahwa Anda tidak membutuhkan sebagian besar dari mereka. Ini akan membantu menghemat banyak ruang disk, karena banyak program berukuran cukup besar. Bahkan jika Anda memang membutuhkan paket atau program tertentu, Anda mungkin tidak membutuhkan semuanya. Misalnya, beberapa dokumentasi online mungkin tidak diperlukan, seperti beberapa file Elisp untuk GNU Emacs, beberapa font untuk X11, atau beberapa perpustakaan untuk pemrograman.

Jika Anda tidak dapat menghapus paket, Anda mungkin melihat ke dalam kompresi. Program kompresi seperti `gzip` atau `zip` akan memampatkan (dan membuka kompresi) file individual atau grup file. Sistem `gzexe` akan mengompres dan membuka kompresi program tanpa terlihat oleh pengguna (program yang tidak digunakan dikompresi, kemudian tidak dikompresi saat digunakan). Sistem `Double` eksperimental akan memampatkan semua file dalam sistem file, tanpa terlihat oleh program yang menggunakannya. (Jika Anda sudah familiar dengan produk seperti `Stacker` untuk MS-DOS atau `DriveSpace` untuk Windows, prinsipnya sama.)

Cara lain untuk menghemat ruang adalah dengan berhati-hati saat memformat partisi Anda. Sebagian besar sistem file modern memungkinkan Anda menentukan ukuran blok. Ukuran blok adalah ukuran potongan yang akan digunakan sistem file untuk membaca dan menulis data. Ukuran blok yang lebih besar akan

Panduan Administrator Sistem Linux

membantu kinerja I/O disk saat menggunakan file besar, seperti database. Hal ini terjadi karena disk dapat membaca atau menulis data untuk jangka waktu yang lebih lama sebelum harus mencari blok berikutnya. Itu

Bab 6. Manajemen Memori

"Minnet, jag har tappat mitt minne, r jag svensk eller finne, kommer inte ihg..." (Bosse sterberg)

Lagu minum Swedia, terjemahan (kasar): ``Memori, saya telah kehilangan ingatan saya. Apakah saya orang Swedia atau Finlandia? aku tidak ingat"

Bagian ini menjelaskan fitur manajemen memori Linux, yaitu memori virtual dan cache buffer disk. Tujuan dan cara kerja dan hal-hal yang perlu dipertimbangkan oleh administrator sistem dijelaskan.

6.1. Apa itu memori virtual?

Linux mendukung memori virtual, yaitu menggunakan disk sebagai perpanjangan RAM sehingga ukuran efektif dari memori yang dapat digunakan tumbuh sesuai. Kernel akan menulis isi blok memori yang saat ini tidak digunakan ke hard disk sehingga memori tersebut dapat digunakan untuk tujuan lain. Ketika konten asli dibutuhkan lagi, mereka dibaca kembali ke dalam memori. Ini semua dibuat sepenuhnya transparan bagi pengguna; program yang berjalan di Linux hanya melihat jumlah memori yang tersedia lebih besar dan tidak memperhatikan bahwa sebagian dari mereka berada di disk dari waktu ke waktu. Tentu saja, membaca dan menulis hard disk lebih lambat (dalam urutan seribu kali lebih lambat) daripada menggunakan memori nyata, sehingga program tidak berjalan dengan cepat. Bagian dari hard disk yang digunakan sebagai memori virtual disebut ruang swap.

Linux dapat menggunakan file normal di sistem file atau partisi terpisah untuk ruang swap. Partisi swap lebih cepat, tetapi lebih mudah untuk mengubah ukuran file swap (tidak perlu mempartisi ulang seluruh hard disk, dan mungkin menginstal semuanya dari awal). Ketika Anda tahu berapa banyak ruang swap yang Anda butuhkan, Anda harus menggunakan partisi swap, tetapi jika Anda tidak yakin, Anda dapat menggunakan file swap terlebih dahulu, gunakan sistem untuk sementara waktu sehingga Anda dapat merasakan berapa banyak swap yang Anda lakukan. perlu, lalu buat partisi swap saat Anda yakin dengan ukurannya.

Anda juga harus tahu bahwa Linux mengizinkan seseorang untuk menggunakan beberapa partisi swap dan/atau file swap pada saat yang bersamaan. Ini berarti bahwa jika Anda hanya sesekali membutuhkan jumlah ruang swap yang tidak biasa, Anda dapat mengatur file swap tambahan pada waktu tersebut, alih-alih menyimpan seluruh jumlah yang dialokasikan sepanjang waktu.

Catatan tentang terminologi sistem operasi: ilmu komputer biasanya membedakan antara swapping (menulis seluruh proses ke ruang swap) dan paging (menulis hanya bagian ukuran tetap, biasanya beberapa kilobyte, pada satu waktu). Paging biasanya lebih efisien, dan itulah yang dilakukan Linux, tetapi terminologi Linux tradisional berbicara tentang swapping.

6.2. Membuat ruang swap

File swap adalah file biasa; itu sama sekali tidak istimewa untuk kernel. Satu-satunya hal yang penting bagi kernel adalah tidak memiliki lubang, dan siap untuk digunakan dengan mkswap. Namun, itu harus berada di disk lokal; itu tidak dapat berada di sistem file yang telah dipasang di atas NFS karena alasan implementasi.

Sedikit tentang lubang itu penting. File swap menyimpan ruang disk sehingga kernel dapat dengan cepat menukar halaman tanpa harus melalui semua hal yang diperlukan saat mengalokasikan sektor disk ke file.

Kernel hanya menggunakan sektor apa pun yang telah dialokasikan ke file. Karena lubang dalam file berarti tidak ada sektor disk yang dialokasikan (untuk tempat itu di file), kernel tidak baik untuk mencoba menggunakannya.

Salah satu cara yang baik untuk membuat file swap tanpa lubang adalah melalui perintah berikut:

```
$ dd if=/dev/zero of=/extra-swap bs=1024
   count=1024
1024+0 catatan dalam
   1024+0 rekaman keluar
$
```

di mana `/ekstra-tukar` adalah nama file swap dan ukurannya diberikan setelah `hitung =`. Yang terbaik adalah ukurannya menjadi kelipatan 4, karena kernel menulis halaman memori, yang berukuran 4 kilobyte. jika

size bukan kelipatan 4, beberapa kilobyte terakhir mungkin tidak digunakan.

Partisi swap juga tidak istimewa. Anda membuatnya seperti partisi lainnya; satu-satunya perbedaan adalah digunakan sebagai partisi mentah, yaitu, tidak akan berisi sistem file sama sekali. Sebaiknya tandai partisi swap sebagai tipe 82 (Linux swap); ini akan membuat daftar partisi lebih jelas, meskipun tidak sepenuhnya diperlukan untuk kernel.

Setelah Anda membuat file swap atau partisi swap, Anda perlu menulis tanda tangan di awal; ini berisi beberapa informasi administratif dan digunakan oleh kernel. Perintah untuk melakukan ini adalah `mkswap`, digunakan seperti ini:

```
$ mkswap /ekstra-swap 1024
Menyiapkan swap space, ukuran =
   1044480 byte
$
```

Perhatikan bahwa ruang swap masih belum digunakan: itu ada, tetapi kernel tidak menggunakannya untuk menyediakan memori virtual.

Anda harus sangat berhati-hati saat menggunakan `mkswap`, karena `mkswap` tidak memeriksa apakah file atau partisi tidak digunakan untuk hal lain. Anda dapat dengan mudah menimpa file dan partisi penting dengan `mkswap`! Untungnya, Anda hanya perlu menggunakan `mkswap` ketika Anda menginstal sistem Anda.

Manajer memori Linux membatasi ukuran setiap ruang swap hingga 2 GB. Namun, Anda dapat menggunakan hingga 8 ruang swap secara bersamaan, dengan total 16GB.

6.3. Menggunakan ruang swap

Ruang swap yang diinisialisasi mulai digunakan dengan `swapon`. Perintah ini memberitahu kernel bahwa ruang swap dapat digunakan. Jalur ke ruang swap diberikan sebagai argumen, jadi untuk mulai menukar file swap sementara, seseorang dapat menggunakan perintah berikut.

```
$ swapon / ekstra swap
$
```

Ruang swap dapat digunakan secara otomatis dengan mendaftarkannya di `/etc/fstab` mengajukan.

/dev/hda8	tidak	menukar	sw	0	0
/swapfile	tidak	menukar	sw	0	0

Skrip startup akan menjalankan perintah `swapon a`, yang akan mulai menukar semua ruang swap yang terdaftar di `/etc/fstab`. Oleh karena itu, perintah `swapon` biasanya digunakan hanya jika diperlukan swap tambahan.

Anda dapat memantau penggunaan ruang swap dengan `gratis`. Ini akan memberi tahu jumlah total ruang swap yang digunakan.

```
$ gratis
      total      diguna   Gratis   bersama
penyangga
saya:  15152      14896      256     12404     2528
/+ buffer:
Menukar:  32452      6684      25768
$
```

Baris pertama keluaran (`saya:`) menunjukkan memori fisik. Kolom total tidak menunjukkan memori fisik yang digunakan oleh kernel, yang biasanya sekitar satu megabyte. Kolom yang digunakan menunjukkan jumlah

memori yang digunakan (baris kedua tidak menghitung buffer). Kolom bebas menunjukkan memori yang sama sekali tidak digunakan. Kolom bersama menunjukkan jumlah memori yang digunakan bersama oleh beberapa proses; semakin banyak, semakin meriah. Kolom buffer menunjukkan ukuran cache buffer disk saat ini.

Baris terakhir itu (`Menukar:`) menunjukkan informasi serupa untuk ruang swap. Jika baris ini semuanya nol, ruang swap Anda tidak diaktifkan.

Informasi yang sama tersedia melalui `top`, atau menggunakan sistem file `proc` dalam file `/proc/meminfo`. Saat ini sulit untuk mendapatkan informasi tentang penggunaan ruang swap tertentu.

Ruang swap dapat dihapus dari penggunaan dengan `swapoff`. Biasanya tidak perlu melakukannya, kecuali untuk ruang swap sementara. Setiap halaman yang digunakan di ruang swap akan ditukar terlebih dahulu; jika tidak ada cukup fisik

memori untuk menyimpannya, mereka kemudian akan ditukar (ke beberapa ruang swap lainnya). Jika memori virtual tidak cukup untuk menampung semua halaman, Linux akan mulai rusak; setelah beberapa lama itu akan pulih, tetapi sementara itu sistem tidak dapat digunakan. Anda harus memeriksa (misalnya, dengan `gratis`) bahwa ada cukup memori bebas sebelum menghapus ruang swap dari penggunaan.

Semua ruang swap yang digunakan secara otomatis dengan `swapon a` dapat dihapus dari penggunaan dengan `swapoff a`; itu terlihat pada file `/etc/fstab` untuk menemukan apa yang harus dihapus. Setiap ruang swap yang digunakan secara manual akan tetap digunakan.

Terkadang banyak ruang swap dapat digunakan meskipun ada banyak memori fisik yang kosong. Ini bisa terjadi misalnya jika pada satu titik ada kebutuhan untuk bertukar, tetapi kemudian proses besar yang menghabiskan banyak memori fisik berakhir dan membebaskan memori. Data yang bertukar-keluar tidak secara otomatis bertukar sampai dibutuhkan, sehingga memori fisik dapat tetap kosong untuk waktu yang lama. Tidak perlu khawatir tentang hal ini, tetapi mengetahui apa yang sedang terjadi dapat menghibur.

6.4. Berbagi ruang swap dengan sistem operasi lain

Memori virtual dibangun ke dalam banyak sistem operasi. Karena mereka masing-masing membutuhkannya hanya ketika mereka sedang berjalan, yaitu, tidak pernah pada waktu yang sama, ruang swap dari semua kecuali yang sedang berjalan akan terbuang sia-sia. Itu akan lebih efisien bagi mereka untuk berbagi ruang swap tunggal. Ini mungkin, tetapi membutuhkan sedikit peretasan. Tips-HOWTO di <http://www.tldp.org/HOWTO/Tips-HOWTO.html>, yang berisi beberapa saran tentang cara menerapkan ini.

6.5. Mengalokasikan ruang swap

Beberapa orang akan memberi tahu Anda bahwa Anda harus mengalokasikan ruang swap dua kali lebih banyak daripada yang Anda miliki memori fisik, tetapi ini adalah aturan palsu. Berikut cara melakukannya dengan benar:

- Perkirakan total kebutuhan memori Anda. Ini adalah jumlah memori terbesar yang mungkin Anda perlukan dalam satu waktu, yaitu jumlah kebutuhan memori untuk semua program yang ingin Anda jalankan pada saat yang sama. Hal ini dapat dilakukan dengan menjalankan pada waktu yang sama semua program yang kemungkinan besar akan Anda jalankan pada waktu yang sama.

Misalnya, jika Anda ingin menjalankan X, Anda harus mengalokasikan sekitar 8 MB untuk itu, gcc menginginkan beberapa megabita (beberapa file membutuhkan jumlah yang luar biasa besar, hingga puluhan megabita, tetapi biasanya sekitar empat megabita), dan seterusnya. Kernel akan menggunakan sekitar satu megabyte dengan sendirinya, dan shell biasa dan utilitas kecil lainnya mungkin beberapa ratus kilobyte (katakanlah satu megabyte bersama-sama). Tidak perlu mencoba untuk menjadi tepat, perkiraan kasar baik-baik saja, tetapi Anda mungkin ingin berada di sisi pesimis.

Ingatlah bahwa jika ada beberapa orang yang menggunakan sistem pada saat yang bersamaan, mereka semua akan menghabiskan memori. Namun, jika dua orang menjalankan program yang sama pada saat yang sama, konsumsi memori total biasanya tidak dua kali lipat, karena halaman kode dan pustaka bersama hanya ada sekali.

Perintah `free` dan `ps` berguna untuk memperkirakan kebutuhan memori.

- Tambahkan beberapa keamanan ke perkiraan di langkah 1. Ini karena perkiraan ukuran program mungkin salah, karena Anda mungkin akan melupakan beberapa program yang ingin Anda jalankan, dan untuk memastikan bahwa Anda memiliki ruang ekstra untuk berjaga-jaga. Beberapa megabyte seharusnya baik-baik saja. (Lebih baik mengalokasikan terlalu banyak daripada terlalu sedikit ruang swap, tetapi tidak perlu berlebihan dan mengalokasikan seluruh disk, karena ruang swap yang tidak digunakan adalah ruang yang terbuang; lihat nanti tentang menambahkan lebih banyak swap.) Juga, karena memang demikian lebih baik untuk menangani angka genap, Anda dapat membulatkan nilainya hingga megabyte penuh berikutnya.
- Berdasarkan perhitungan di atas, Anda tahu berapa banyak memori yang Anda perlukan secara total. Jadi, untuk mengalokasikan ruang swap, Anda hanya perlu mengurangi ukuran memori fisik Anda dari total memori yang dibutuhkan, dan Anda tahu berapa banyak ruang swap yang Anda butuhkan. (Pada beberapa versi UNIX, Anda juga perlu mengalokasikan ruang untuk gambar memori fisik, jadi jumlah yang dihitung pada langkah 2 adalah yang Anda butuhkan dan Anda tidak boleh melakukan pengurangan.)

- Jika ruang swap yang Anda hitung jauh lebih besar dari memori fisik Anda (lebih dari beberapa kali lebih besar), Anda mungkin harus berinvestasi di lebih banyak memori fisik, jika tidak, kinerjanya akan terlalu rendah.

Merupakan ide bagus untuk memiliki setidaknya beberapa ruang swap, bahkan jika perhitungan Anda menunjukkan bahwa Anda tidak membutuhkannya. Linux menggunakan ruang swap agak agresif, sehingga sebanyak mungkin memori fisik dapat tetap bebas. Linux akan menukar halaman memori yang belum digunakan, bahkan jika memori tersebut belum diperlukan untuk apa pun. Hal ini untuk menghindari menunggu swapping saat dibutuhkan: swapping dapat dilakukan lebih awal, saat disk dalam keadaan idle.

Ruang swap dapat dibagi di antara beberapa disk. Ini terkadang dapat meningkatkan kinerja, tergantung pada kecepatan relatif disk dan pola akses disk. Anda mungkin ingin bereksperimen dengan beberapa skema, tetapi perlu diketahui bahwa melakukan eksperimen dengan benar cukup sulit. Anda tidak boleh percaya klaim bahwa salah satu skema lebih unggul dari yang lain, karena tidak akan selalu benar.

6.6. Cache penyangga

Membaca dari disk sangat lambat dibandingkan dengan mengakses memori (nyata). Selain itu, adalah umum untuk membaca bagian yang sama dari disk beberapa kali selama periode waktu yang relatif singkat. Misalnya, seseorang mungkin pertama membaca pesan email, lalu membaca surat itu ke editor saat membalasnya, lalu membuat program email membacanya lagi saat menyalinnya ke folder. Atau, pertimbangkan seberapa sering perintah `ls` dijalankan pada sistem dengan banyak pengguna. Dengan membaca informasi dari disk hanya sekali dan kemudian menyimpannya dalam memori sampai tidak diperlukan lagi, seseorang dapat mempercepat semua kecuali pembacaan pertama. Ini disebut buffering disk, dan memori yang digunakan untuk tujuan tersebut disebut buffer cache.

Karena memori, sayangnya, adalah sumber daya yang terbatas, bahkan langka, cache buffer biasanya tidak cukup besar (tidak dapat menampung semua data yang ingin digunakan). Ketika cache terisi, data yang telah lama tidak digunakan akan dibuang dan memori yang dibebaskan digunakan untuk data baru.

Penyangga disk juga berfungsi untuk penulisan. Di satu sisi, data yang ditulis sering segera dibaca kembali (misalnya, file kode sumber disimpan ke file, kemudian dibaca oleh kompilator), jadi memasukkan data yang ditulis ke dalam cache adalah ide yang baik. Di sisi lain, dengan hanya memasukkan data ke dalam cache, tidak menulisnya ke disk sekaligus, program yang menulis akan berjalan lebih cepat. Penulisan kemudian dapat dilakukan di latar belakang, tanpa memperlambat program lain.

Sebagian besar sistem operasi memiliki cache buffer (walaupun mereka mungkin disebut sesuatu yang lain), tetapi tidak semuanya bekerja sesuai dengan prinsip di atas. Beberapa di antaranya adalah write-through: data ditulis ke disk sekaligus (disimpan dalam cache juga, tentu saja). Cache disebut write-back jika penulisan dilakukan di lain waktu. Write-back lebih efisien daripada write-through, tetapi juga sedikit lebih rentan terhadap kesalahan: jika mesin mogok, atau listrik terputus pada saat yang buruk, atau floppy dihapus dari disk drive sebelum data dalam cache yang menunggu untuk ditulis ditulis, perubahan dalam cache biasanya hilang. Ini bahkan mungkin berarti bahwa sistem file (jika ada) tidak berfungsi penuh, mungkin karena data tidak tertulis memiliki perubahan penting pada informasi pembukuan.

Oleh karena itu, Anda tidak boleh mematikan daya tanpa menggunakan prosedur shutdown yang benar atau melepas floppy dari drive disk sampai telah dilepas (jika sudah terpasang) atau setelah program apa pun yang menggunakannya telah memberi tanda bahwa itu selesai dan lampu floppy drive tidak bersinar lagi. Perintah sinkronisasi menghapus buffer, yaitu, memaksa semua data tidak tertulis untuk ditulis ke disk, dan dapat digunakan ketika seseorang ingin memastikan bahwa semuanya ditulis dengan aman. Dalam sistem UNIX tradisional, ada program yang disebut pembaruan yang berjalan di latar belakang yang melakukan

Panduan Administrator Sistem Linux

sinkronisasi setiap 30 detik, jadi biasanya tidak perlu menggunakan sinkronisasi. Linux memiliki daemon tambahan, `bdflush`, yang melakukan sinkronisasi lebih tidak sempurna lebih sering untuk menghindari pembekuan mendadak karena I/O disk berat yang terkadang menyebabkan sinkronisasi.

Di Linux, `bdflush` dimulai dengan pembaruan. Biasanya tidak ada alasan untuk mengkhawatirkannya, tetapi jika `bdflush` mati karena suatu alasan, kernel akan memperingatkan tentang hal ini, dan Anda harus memulainya dengan tangan (`/sbin/update`).

Cache sebenarnya tidak menyangga file, tetapi memblokir, yang merupakan unit terkecil dari disk I/O (di Linux, biasanya berukuran 1 KB). Dengan cara ini, folder, blok super, data pembukuan sistem file lainnya, dan disk non-sistem file juga di-cache.

Efektivitas cache terutama ditentukan oleh ukurannya. Cache kecil hampir tidak berguna: cache akan menampung begitu sedikit data sehingga semua data cache dihapus dari cache sebelum digunakan kembali. Ukuran kritis tergantung pada seberapa banyak data yang dibaca dan ditulis, dan seberapa sering data yang sama diakses. Satu-satunya cara untuk mengetahuinya adalah dengan bereksperimen.

Jika cache berukuran tetap, juga tidak baik untuk menyimpannya terlalu besar, karena dapat membuat memori kosong terlalu kecil dan menyebabkan swapping (yang juga lambat). Untuk membuat penggunaan memori nyata yang paling efisien, Linux secara otomatis menggunakan semua RAM kosong untuk cache buffer, tetapi juga secara otomatis membuat cache lebih kecil ketika program membutuhkan lebih banyak memori.

Di Linux, Anda tidak perlu melakukan apa pun untuk menggunakan cache, itu terjadi sepenuhnya secara otomatis. Kecuali mengikuti prosedur yang benar untuk mematikan dan menghapus disket, Anda tidak perlu mengkhawatirkannya.

Bab 7. Pemantauan Sistem

"Itu Hall Monitor untukmu!" Spongebob Squarepants

Salah satu tanggung jawab terpenting yang dimiliki administrator sistem, adalah memantau sistem mereka. Sebagai administrator sistem, Anda memerlukan kemampuan untuk mengetahui apa yang terjadi pada sistem Anda pada waktu tertentu. Baik itu persentase sumber daya sistem yang saat ini digunakan, perintah apa yang sedang dijalankan, atau siapa yang masuk. Bab ini akan membahas cara memantau sistem Anda, dan dalam beberapa kasus, cara mengatasi masalah yang mungkin muncul.

Ketika masalah kinerja muncul, ada 4 area utama yang perlu dipertimbangkan: CPU, Memori, Disk I/O, dan Jaringan. Kemampuan untuk menentukan di mana hambatan dapat menghemat banyak waktu Anda.

7.1. Sumber Daya Sistem

Mampu memantau kinerja sistem Anda sangat penting. Jika sumber daya sistem menjadi rendah dapat menyebabkan banyak masalah. Sumber daya sistem dapat diambil oleh pengguna individu, atau oleh layanan yang mungkin dihosting oleh sistem Anda seperti email atau halaman web. Kemampuan untuk mengetahui apa yang terjadi dapat membantu menentukan apakah pemutakhiran sistem diperlukan, atau jika beberapa layanan perlu dipindahkan ke komputer lain.

7.1.1. Perintah teratas.

Yang paling umum dari perintah ini adalah atas. Bagian atas akan menampilkan laporan penggunaan sumber daya sistem yang terus diperbarui.

```
# atas
12:10:49 hingga 1 hari, 3:47, 7 pengguna, rata-rata memuat: 0,23, 0,19, 0,10
125 proses: 105 tidur, 2 berjalan, 18 zombie, 0 berhenti
Status CPU: 5,1% pengguna 1,1% sistem 0,0% bagus 0,0% iowait 93,6% menganggur
Mem: 512716k av, 506176k digunakan, 6540k gratis, 0k shrd, 21888k buff Tukar:
1044216k av, 161672k digunakan, 882544k gratis 199388k cache

  PID PENGGUN  PR  NI  UKUR  RSS  BAGIK  STAT  %CPU  %MEM  WAKT  CPU  MEMERINTAH
2330 akar      15   0  161M  70M  2132  S    4.9  14.0  1000m  0  x
2605 minggu    15   0   8240 6340  3804  S    0,3  1.2   1:12  0  kdeinit
3413 minggu    15   0   6668 5324  3216  R    0,3  1.0   0:20  0  kdeinit
18734 akar     15   0   1192 1192   868  R    0,3  0.2   0:00  0  atas
1619 akar     15   0    776  608   504  S    0.1  0.1   0:53  0  klien
1    akar     15   0    480  448   424  S    0,0  0,0   0:03  0  init
2    akar     15   0  0     0     0     SW   0,0  0,0   0:00  0  keventd
3    akar     15   0  0     0     0     SW   0,0  0,0   0:00  0  kapmd
4    akar     35  19  0     0     0     SWN  0,0  0,0   0:00  0  ksoftirqd_CPU0
9    akar     25   0  0     0     0     SW   0,0  0,0   0:00  0  bdflood
5    akar     15   0  0     0     0     SW   0,0  0,0   0:00  0  kswapd
10   akar     15   0  0     0     0     SW   0,0  0,0   0:00  0  kupdate
11   akar     25   0  0     0     0     SW   0,0  0,0   0:00  0  mdrecoveryd
15   akar     15   0  0     0     0     SW   0,0  0,0   0:01  0  jurnal
81   akar     25   0  0     0     0     SW   0,0  0,0   0:00  0  khubd
1188 akar     15   0  0     0     0     SW   0,0  0,0   0:00  0  jurnal
1675 akar     15   0   604  572   520  S    0,0  0.1   0:00  0  syslogd
1679 akar     15   0   428  376   372  S    0,0  0,0   0:00  0  klogd
1707 rpc       15   0   516  440   436  S    0,0  0,0   0:00  0  peta port
1776 akar     25   0   476  428   424  S    0,0  0,0   0:00  0  apmd
```

1813 akar	25	0	752	528	524	S	0,0	0.1	0:00	0	sshd
1828 akar	25	0	704	548	544	S	0,0	0.1	0:00	0	xinetd
1847 ntp	15	0	2396	2396	2160	S	0,0	0.4	0:00	0	ntpd
1930 akar	24	0	76	4	0	S	0,0	0,0	0:00	0	rpc.rquotad

Bagian atas laporan mencantumkan informasi seperti waktu sistem, waktu aktif, penggunaan CPU, fisik, dan swap penggunaan memori, dan jumlah proses. Di bawah ini adalah daftar proses yang diurutkan berdasarkan penggunaan CPU.

Anda dapat memodifikasi output top saat sedang berjalan. Jika Anda menekan `Saya`, top tidak akan lagi menampilkan proses idle. Memukul `Saya` lagi untuk melihat mereka lagi. Memukul `M` akan mengurutkan berdasarkan penggunaan memori, `S` akan mengurutkan berdasarkan berapa lama prosesnya telah berjalan, dan `P` akan mengurutkan berdasarkan penggunaan CPU lagi.

Selain opsi tampilan, Anda juga dapat memodifikasi proses dari dalam perintah teratas. Kamu dapat memakai `kamu` untuk melihat proses yang dimiliki oleh pengguna tertentu, `k` untuk mematikan proses, dan `R` untuk memperlakukan mereka.

Untuk informasi lebih mendalam tentang proses Anda dapat melihat di `/proc` berkas sistem. Dalam `/proc` sistem file Anda akan menemukan serangkaian sub-folder dengan nama numerik. Folder ini terkait dengan id proses dari proses yang sedang berjalan. Di setiap folder Anda akan menemukan serangkaian file yang berisi informasi tentang proses tersebut.

ANDA HARUS PERHATIAN EKSTREM UNTUK TIDAK MENGUBAH FILE INI, MELAKUKANNYA DAPAT MENYEBABKAN MASALAH SISTEM!

7.1.2. Perintah iostat.

Iostat akan menampilkan rata-rata beban CPU saat ini dan informasi I/O disk. Ini adalah perintah yang bagus untuk memantau penggunaan I/O disk Anda.

```
# iostat
Linux 2.4.20-24.9 (hosting saya) 23/12/2003

rata-      %pengg   %baik    %sys    %meng
          62.09    0.32    2.97    34.62

Perangkat:      tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
dev3-0          2.22         15.20         47.16     1546846     4799520
```

Untuk kernel 2.4, perangkat diberi nama menggunakan nomor mayor dan minor perangkat. Dalam hal ini perangkat yang terdaftar adalah `/dev/hda`. Untuk meminta iostat mencetak ini untuk Anda, gunakan `x`.

```
# iostat x
Linux 2.4.20-24.9 (hosting saya) 23/12/2003

rata-      %pengg   %baik    %sys    %meng
          62.01    0.32    2.97    34.71

Perangka rrqm/s wrqm/s r/sw/s rsec/s    wsec/s rkB/s wkB/s avgrq-sz avgqu-sz menunggu svctm
t:         0.00 0.00    .00 0.00 0.00    %util
/dev/hdc   1.13 4.50    .08 1,39 15,18    0.00 0.00 0.00 0.00 2.35 0.00 0.00 14.71
/dev/hda   1,08 3,98    .73 1,27 14,49    47,14 7,59 23,57 28,24 1,99 63,76 70,48 15,56
/dev/hda1  0,00 0,51    0,07 0,12 0,55    42,05 7,25 21,02 28,22 0,44 21,82 4,97 1,00
/dev/hda2  0,05 0,01    0,02 0,00 0,14    5.07 0.27 2.54 30.35 0.97 52.67 61.73 2.99
/dev/hda3          0,02    0,07    0,01    8,51    0,00    12,55    2,95    0,01
```

Halaman manual iostat berisi penjelasan terperinci tentang arti masing-masing kolom ini.

7.1.3. Perintah ps

PS akan memberi Anda daftar proses yang sedang berjalan. Ada berbagai macam opsi yang diberikan perintah ini kepada Anda.

Penggunaan yang umum adalah membuat daftar semua proses yang sedang berjalan. Untuk melakukan ini, Anda akan menggunakan `ps ef` memerintah. (Output layar dari perintah ini terlalu besar untuk disertakan, berikut ini hanya sebagian output.)

UID	PID	PPID	C	WAKTU	TTY	WAKT	cmd
akar	1	0	0	22	?	00:00:03	init
akar	2	1	0	22	?	00:00:00	[keventd]
akar	3	1	0	22	?	00:00:00	[kapmd]
akar	4	1	0	22	?	00:00:00	[ksoftirqd_CPU0]
akar	9	1	0	22	?	00:00:00	[bdflush]
akar	5	1	0	22	?	00:00:00	[swap]
akar	6	1	0	22	?	00:00:00	[kscand/DMA]
akar	7	1	0	22	?	00:01:28	[kscand/Normal]
akar	8	1	0	22	?	00:00:00	[kscand/HighMem]
akar	10	1	0	22	?	00:00:00	[kupdate]
akar	11	1	0	22	?	00:00:00	[mdrecoveryd]
akar	15	1	0	22	?	00:00:01	[kjurnal]
akar	81	1	0	22	?	00:00:00	[khubd]
akar	1188	1	0	22	?	00:00:00	[kjurnal]
akar	1675	1	0	22	?	00:00:00	syslogd m 0
akar	1679	1	0	22	?	00:00:00	klogd x
rpc	1707	1	0	22	?	00:00:00	peta port
akar	1813	1	0	22	?	00:00:00	/usr/sbin/sshd
ntp	1847	1	0	22	?	00:00:00	ntpd U ntp
akar	1930	1	0	22	?	00:00:00	rpc.rquotad
akar	1934	1	0	22	?	00:00:00	[nfsd]
akar	1942	1	0	22	?	00:00:00	[terkunci]
akar	1943	1	0	22	?	00:00:00	[rpciod]
akar	1949	1	0	22	?	00:00:00	rpc.mountd
akar	1961	1	0	22	?	00:00:00	/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
akar	2057	1	0	22	?	00:00:00	/usr/bin/spamd d c a
akar	2066	1	0	22	?	00:00:00	gpm t ps/2 m /dev/psaux
tempat	2076	1	0	22	?	00:00:00	/usr/sbin/cannaserver syslog u bin
akar	2087	1	0	22	?	00:00:00	cron
daemon	2195	1	0	22	?	00:00:00	/usr/sbin/atd
akar	2215	1	0	22	?	00:00:11	/usr/sbin/rcd
minggu	3414	3413	0	22	poin/1	00:00:00	/bin/bash
minggu	4342	3413	0	22	poin/2	00:00:00	/bin/bash
minggu	19121	18668	0	12:58	poin/2	00:00:00	ps ef

Kolom pertama menunjukkan siapa yang memiliki proses. Kolom kedua adalah ID proses. Kolom ketiga adalah

ID proses induk. Ini adalah proses yang menghasilkan, atau memulai, proses. Kolom keempat adalah penggunaan CPU (dalam persen). Kolom kelima adalah waktu mulai, tanggal jika proses sudah berjalan cukup lama. Kolom keenam adalah tty yang terkait dengan proses, jika berlaku. Kolom ketujuh adalah penggunaan CPU kumulatif (jumlah total waktu CPU telah digunakan saat berjalan). Kolom kedelapan adalah perintah itu sendiri.

Dengan informasi ini, Anda dapat melihat dengan tepat apa yang sedang berjalan di sistem Anda dan menghentikan proses yang berjalan, atau proses yang menyebabkan masalah.

7.1.4. Perintah vmstat

Perintah vmstat akan memberikan laporan yang menunjukkan statistik untuk proses sistem, memori, swap, I/O, dan CPU. Statistik ini dihasilkan menggunakan data dari terakhir kali perintah dijalankan hingga saat ini. Dalam hal perintah tidak pernah dijalankan, data akan dimulai dari reboot terakhir hingga saat ini.

#vmstat															
prose Penyimpanan					menuk			io		sistem		cpu			
rbw	swpd	buff	cache	gratis	si	jad	dua	bo	dalam	cs	kami	sy	id		
0	0	0	181604	17000	26296	201120	0	2	8	24	149	9	61	3	36

Berikut ini diambil dari halaman manual vmstat.

DESKRIPSI LAPANGAN

PROSES

- r : Jumlah proses yang menunggu run time.
- b: Jumlah proses dalam uninterruptable sleep.
- w: Jumlah proses yang ditukar tetapi sebaliknya dapat dijalankan.
Bidang ini dihitung, tetapi Linux tidak pernah putus asa swap.

Penyimpanan

- swpd: jumlah memori virtual yang digunakan (kB).
- free: jumlah memori idle (kB).
- buff: jumlah memori yang digunakan sebagai buffer (kB).

Menukar

- si: Jumlah memori yang ditukar dari disk (kB/s).
- jadi: Jumlah memori yang ditukar ke disk (kB/s).

saya

- bi: Blok dikirim ke perangkat blok (blok/dtk).
- bo: Blok yang diterima dari perangkat blok (blok/dtk).

Sistem

- in: Jumlah interupsi per detik, termasuk jam.
- cs: Jumlah sakelar konteks per detik.

CPU

- Ini adalah persentase dari total waktu CPU.
- kami: waktu pengguna
- sy: waktu sistem
- id: waktu mengganggu

7.1.5. Perintah lsof

Perintah lsof akan mencetak daftar setiap file yang sedang digunakan. Karena Linux menganggap semuanya sebagai file, daftar ini bisa sangat panjang. Namun, perintah ini dapat berguna dalam mendiagnosis masalah. Contohnya adalah jika Anda ingin meng-unmount sistem file, tetapi Anda diberitahu bahwa itu sedang digunakan. Anda dapat menggunakan perintah ini dan grep untuk nama sistem file untuk melihat siapa yang menggunakannya.

Atau misalkan Anda ingin melihat semua file yang digunakan oleh proses tertentu. Untuk melakukan ini, Anda akan menggunakan `ls` proses.

7.1.6. Menemukan Lebih Banyak Utilitas

Untuk mempelajari lebih lanjut tentang alat baris perintah apa yang tersedia, Chris Karakas telah menulis [panduan referensi berjudul Perintah GNU/Linux—Ringkasan Alat Baris](#). Ini adalah sumber yang bagus untuk mempelajari alat apa yang ada di luar sana dan bagaimana melakukan sejumlah tugas.

7.2. Penggunaan Sistem File

Banyak laporan saat ini berbicara tentang betapa mudahnya penyimpanan, tetapi jika Anda seperti kebanyakan dari kita, itu tidak cukup murah. Sebagian besar dari kita memiliki jumlah ruang yang terbatas, dan harus dapat memantau dan mengontrol penggunaannya.

7.2.1. Perintah `df`

`Df` adalah alat paling sederhana yang tersedia untuk melihat penggunaan disk. Cukup ketik `df` dan Anda akan diperlihatkan penggunaan disk untuk semua sistem file yang Anda pasang dalam 1K blok

```

pengguna@serve df
r :~> Sistem      1K-blok      Digu Tersedia Meng Dipasang pada
/dev/hda3         5242904      759692      4483212    15% /
tmpfs             127876      8           127868     1% /dev/shm
/dev/hda1         127351      33047       87729     28% /boot
/dev/hda9         10485816    33508      10452308   1% /home
/dev/hda8         5242904     932468     4310436   18% /srv
/dev/hda7         3145816     32964      3112852    2% /tmp
/dev/hda5         5160416     474336     4423928   10% /usr
/dev/hda6         3145816     412132     2733684   14% /var
    
```

Anda juga dapat menggunakan `h` untuk melihat output dalam format "dapat dibaca manusia". Ini akan berada di K, Megs, atau Gigs tergantung pada ukuran sistem file. Sebagai alternatif, Anda juga dapat menggunakan `B` untuk menentukan ukuran blok.

Selain penggunaan ruang, Anda dapat menggunakan opsi `i` untuk melihat jumlah inode yang digunakan dan yang tersedia.

```

pengguna@serve df i
r :~> Sistem      inode      digun      Saya      Saya      Dipasang pada
/dev/hda3         0          0          0          -          /
tmpfs             31969     5          31964     1% /dev/shm
/dev/hda1         32912     47         32865     1% /boot
/dev/hda9         0          0          0          -          /home
/dev/hda8         0          0          0          -          /srv
/dev/hda7         0          0          0          -          /tmp
/dev/hda5         656640    26651     629989     5% /usr
/dev/hda6         0          0          0          -          /var
    
```

7.2.2. Perintah `du`

Panduan Administrator Sistem Linux

Sekarang Anda tahu berapa banyak ruang yang telah digunakan pada sistem file, bagaimana Anda bisa mengetahui di mana data itu? Untuk melihat penggunaan oleh folder atau file, Anda dapat menggunakan `du`. Kecuali Anda menentukan nama file `du` akan bertindak secara rekursif. Sebagai contoh:

```
pengguna@server :~> du
file.txt
1300 file.txt
```

Atau seperti `df` saya bisa menggunakan `h` dan mendapatkan output yang sama dalam bentuk "dapat dibaca manusia".

```
pengguna@server :~> du h
file.txt
1,3 juta file.txt
```

Kecuali Anda menentukan nama file `du` akan bertindak secara rekursif.

```
pengguna@server :~> du h
/usr/local
4.0K /usr/local/games
16K /usr/local/include/nessus/net
180K /usr/local/include/nessus
208K /usr/local/include
62M /usr/local/lib/nessus/plugins/.desc
97 juta /usr/local/lib/nessus/plugins
164K /usr/local/lib/nessus/plugins_factory
97 juta /usr/local/lib/nessus
12K /usr/local/lib/pkgconfig
2,7 juta
/usr/local/lib/ladspa
104 juta
/usr/local/lib
112K /usr/local/man/man1
4.0K /usr/local/man/man2
4.0K /usr/local/man/man3
4.0K /usr/local/man/man4
16K /usr/local/man/man5
4.0K /usr/local/pria/pria
```

Jika Anda hanya ingin ringkasan folder itu, Anda dapat menggunakan opsi `s`.

```
pengguna@server :~> du hs
/usr/local
210jt /usr/lokal
```

7.2.3. kuota

Untuk informasi lebih lanjut tentang kuota Anda dapat membaca [The Quota HOWTO](#).

7.3. Memantau Pengguna

Hanya karena Anda paranoid bukan berarti mereka TIDAK keluar untuk menangkap Anda... Sumber Tidak Diketahui

Dari waktu ke waktu akan ada kesempatan di mana Anda ingin tahu persis apa yang dilakukan orang di sistem Anda. Mungkin Anda memperhatikan bahwa banyak RAM yang digunakan, atau banyak aktivitas CPU. Anda

akan ingin melihat siapa yang ada di sistem, apa yang mereka jalankan, dan sumber daya apa yang mereka gunakan.

7.3.1. Yang memerintahkan

Cara termudah untuk melihat siapa yang ada di sistem adalah dengan melakukan `w`. > `who` adalah alat sederhana yang mencantumkan siapa yang masuk —> pada sistem dan port atau terminal apa yang mereka masuki.

```
pengguna@server :~>
who
sjapas poin/0      Bol 23 09:33
wally  poin/3      Bol 20 11:35
minggu poin/1      Bol 22 11:03
minggu poin/2      Bol 23 15:04
```

7.3.2. Perintah ps lagi!

Di bagian sebelumnya kita dapat melihat bahwa pengguna seminggu masuk ke keduanya poin/1 dan poin/2, tapi bagaimana jika kita ingin melihat apa yang mereka lakukan? Kita bisa ke ps u aweeks dan mendapatkan output berikut

```
pengguna@server :~> ps u aweeks
20876 poin/1 00:00:00 bash
20904 poin/2 00:00:00 bash
20951 poin/2 00:00:00 ssh
21012 poin/1 00:00:00 ps
```

Dari sini kita dapat melihat bahwa pengguna melakukan ps ssh.

Ini adalah penggunaan ps yang jauh lebih terkonsolidasi daripada yang dibahas sebelumnya.

7.3.3. perintah w

Bahkan lebih mudah daripada menggunakan perintah who dan ps u adalah dengan menggunakan w. w akan mencetak tidak hanya siapa yang ada di sistem, tetapi juga perintah yang mereka jalankan.

```
pengguna@server :~> w
minggu :0 09:32 ?xdm? 30:09 0,02 detik :0
minggu  poin/0   09:33   5:49m 0.00   0.82s kdeinit: kded
minggu  poin/2   09:35   8.00s 0,55   0.36s vi sag-0.9.shtml
minggu  poin/1   15:03  59.00 0,03   0,03  /bin/bash
```

Dari sini kita dapat melihat bahwa saya menjalankan sesi kde, saya sedang mengerjakan dokumen ini :-), dan memiliki yang lain terminal terbuka duduk diam di bash Prompt.

7.3.4. Perintah keterampilan

Untuk ditambahkan

7.3.5. bagus dan renice

Untuk

ditambahkan

Bab 8. Boot dan Shutdown

```
Mulai saya
Ah... kau harus... kau harus
Jangan pernah, jangan pernah berhenti
Mulailah
Ah ... memulainya, tidak pernah, tidak pernah, tidak pernah
Anda membuat pria dewasa
menangis, Anda membuat
pria dewasa menangis
(Batu Bergulir)
```

Bagian ini menjelaskan apa yang terjadi ketika sistem Linux dinaikkan dan diturunkan, dan bagaimana hal itu harus dilakukan dengan benar. Jika prosedur yang benar tidak diikuti, file mungkin rusak atau hilang.

8.1. Ikhtisar tentang boot dan shutdown

Tindakan menyalakan sistem komputer dan menyebabkan sistem operasinya dimuat disebut booting. Nama itu berasal dari gambar komputer yang menarik dirinya dari bootstrapnya, tetapi tindakan itu sendiri sedikit lebih realistis.

Selama bootstrap, komputer pertama-tama memuat sepotong kecil kode yang disebut bootstrap loader, yang kemudian memuat dan memulai sistem operasi. Bootstrap loader biasanya disimpan di lokasi tetap pada hard disk atau floppy. Alasan untuk proses dua langkah ini adalah karena sistem operasinya besar dan rumit, tetapi potongan kode pertama yang dimuat komputer harus sangat kecil (beberapa ratus byte), untuk menghindari kerumitan yang tidak perlu pada firmware.

Komputer yang berbeda melakukan bootstrap secara berbeda. Untuk PC, komputer (BIOS-nya) membaca di sektor pertama (disebut boot sector) dari floppy atau hard disk. Bootstrap loader terdapat di dalam sektor ini. Ini memuat sistem operasi dari tempat lain di disk (atau dari tempat lain).

Setelah Linux dimuat, ia menginisialisasi driver perangkat keras dan perangkat, dan kemudian menjalankan init. init memulai proses lain untuk mengizinkan pengguna masuk, dan melakukan sesuatu. Rincian bagian ini akan dibahas di bawah ini.

Untuk mematikan sistem Linux, pertama-tama semua proses diperintahkan untuk dihentikan (ini membuat mereka menutup semua file dan melakukan hal-hal lain yang diperlukan untuk menjaga semuanya tetap rapi), kemudian sistem file dan area swap dilepas, dan akhirnya sebuah pesan dicetak ke sistem. konsol bahwa daya dapat dimatikan. Jika prosedur yang benar tidak diikuti, hal-hal buruk dapat dan akan terjadi; yang paling penting, cache buffer sistem file mungkin tidak dihapus, yang berarti bahwa semua data di dalamnya hilang dan sistem file pada disk tidak konsisten, dan oleh karena itu mungkin tidak dapat digunakan.

8.2. Proses boot dilihat lebih dekat

Saat PC di-boot, BIOS akan melakukan berbagai tes untuk memeriksa apakah semuanya terlihat baik-baik saja, dan kemudian akan memulai booting yang sebenarnya. Proses ini disebut power on self test, atau disingkat POST. Ini akan memilih drive disk (biasanya floppy drive pertama, jika ada floppy dimasukkan, jika tidak, hard disk pertama, jika ada yang diinstal di komputer; urutannya mungkin dapat dikonfigurasi) dan kemudian akan membaca sektor pertamanya. Ini disebut sektor boot; untuk hard disk, ini juga disebut

Panduan Administrator Sistem Linux

master boot record, karena hard disk dapat berisi beberapa partisi, masing-masing dengan sektor bootnya sendiri.

Sektor boot berisi program kecil (cukup kecil untuk masuk ke dalam satu sektor) yang bertanggung jawab untuk membaca sistem operasi yang sebenarnya dari disk dan memulainya. Saat mem-boot Linux dari floppy disk, sektor boot berisi kode yang hanya membaca beberapa ratus blok pertama (tentu saja tergantung pada ukuran kernel yang sebenarnya) ke tempat yang telah ditentukan di memori. Pada disket boot Linux, tidak ada sistem file, kernel hanya disimpan di sektor yang berurutan, karena ini menyederhanakan proses boot. Akan tetapi, dimungkinkan untuk mem-boot dari floppy dengan sistem file, dengan menggunakan LILO, LINUX LOader, atau GRUB, GRand Unifying Bootloader.

Saat booting dari hard disk, kode di master boot record akan memeriksa tabel partisi (juga di master boot record), mengidentifikasi partisi yang aktif (partisi yang ditandai dapat di-boot), membaca sektor boot dari partisi itu, lalu mulai kode di sektor boot tersebut. Kode di sektor boot partisi melakukan apa yang dilakukan sektor boot floppy disk: itu akan membaca di kernel dari partisi dan memulainya. Rinciannya bervariasi, namun, karena umumnya tidak berguna untuk memiliki partisi terpisah hanya untuk gambar kernel, sehingga kode di sektor boot partisi tidak dapat hanya membaca disk secara berurutan, ia harus menemukan sektor di mana pun sistem file memiliki letakkan. Ada beberapa cara untuk mengatasi masalah ini, tetapi cara yang paling umum adalah dengan menggunakan boot loader seperti LILO atau GRUB. (Detail tentang bagaimana melakukan ini tidak relevan untuk diskusi ini; lihat dokumentasi LILO atau GRUB untuk informasi lebih lanjut; ini yang paling menyeluruh.)

Saat boot, bootloader biasanya akan langsung melanjutkan dan membaca dan mem-boot kernel default. Dimungkinkan juga untuk mengonfigurasi boot loader agar dapat mem-boot salah satu dari beberapa kernel, atau bahkan sistem operasi lain selain Linux, dan pengguna dapat memilih kernel atau sistem operasi mana yang akan di-boot pada saat boot. LILO, misalnya, dapat dikonfigurasi sehingga jika seseorang menahan tombol alt, shift, atau ctrl pada saat boot (ketika LILO dimuat), LILO akan menanyakan apa yang akan di-boot dan tidak langsung mem-boot default. Atau, bootloader dapat dikonfigurasi agar selalu bertanya, dengan batas waktu opsional yang akan menyebabkan kernel default di-boot.

Dimungkinkan juga untuk memberikan argumen baris perintah kernel, setelah nama kernel atau sistem operasi. Untuk daftar opsi yang memungkinkan, Anda dapat membaca <http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html>.

Booting dari floppy dan dari hard disk memiliki keuntungan masing-masing, tetapi umumnya booting dari hard disk lebih baik, karena menghindari kerumitan bermain-main dengan disket. Hal ini juga lebih cepat. Sebagian besar distribusi Linux akan mengatur bootloader untuk Anda selama proses instalasi.

Setelah kernel Linux telah dibaca ke dalam memori, dengan cara apa pun, dan dimulai secara nyata, kira-kira hal-hal berikut terjadi:

- Kernel Linux diinstal terkompresi, sehingga pertama-tama akan membuka kompresinya sendiri. Bagian awal gambar kernel berisi program kecil yang melakukan ini.
- Jika Anda memiliki kartu super-VGA yang dikenali Linux dan memiliki beberapa mode teks khusus (seperti 100 kolom kali 40 baris), Linux menanyakan mode mana yang ingin Anda gunakan. Selama kompilasi kernel, dimungkinkan untuk mengatur mode video, sehingga ini tidak pernah ditanyakan. Ini juga dapat dilakukan dengan LILO, GRUB atau rdev.
- Setelah ini, kernel memeriksa perangkat keras lain yang ada (hard disk, disket, adaptor jaringan, dll), dan mengkonfigurasi beberapa driver perangkatnya dengan tepat; saat melakukan ini, ia mengeluarkan pesan tentang temuannya. Misalnya, ketika saya boot, saya terlihat seperti ini:

Panduan Administrator Sistem Linux

```
LILO      boot:
Memuat linux.
Konsol: warna EGA+ 80x25, 8 konsol virtual
Driver serial versi 3.94 tanpa opsi serial yang
diaktifkan tty00 di 0x03f8 (irq = 4) adalah 16450
tty01 di 0x02f8 (irq = 3) adalah 16450
lp_init: lp1 ada (0), menggunakan driver polling
Memori: 7332k/8192k tersedia (kode kernel 300k, 384k dicadangkan, 176k
data)
Floppy drive(s): fd0 adalah 1,44M, fd1
adalah 1,2M perangkat Loopback init
apan peringatan WD8013 tidak ditemukan di i/o
= 280. Koprocesor matematika menggunakan
pelaporan kesalahan irq13. Pemeriksaan
partisi:
  hda: hda1 hda2 hda3
VFS: Root terpasang (sistem file ext).
Linux versi 0.99.pl9-1 ( root@haven ) 01/05/93 14:12:20
```

Teks persisnya berbeda pada sistem yang berbeda, tergantung pada perangkat kerasnya, versi Linux yang digunakan, dan cara konfigurasinya.

- Kemudian kernel akan mencoba me-mount sistem file root. Tempat dapat dikonfigurasi pada waktu kompilasi, atau kapan saja dengan rdev atau bootloader. Jenis sistem file terdeteksi secara otomatis. Jika pemasangan sistem berkas root gagal, misalnya karena Anda tidak ingat untuk memasukkan driver sistem berkas yang sesuai di kernel, kernel panik dan menghentikan sistem (tidak banyak yang bisa dilakukan).

Sistem file root biasanya dipasang hanya-baca (ini dapat diatur dengan cara yang sama seperti tempatnya). Ini memungkinkan untuk memeriksa sistem file saat dipasang; bukanlah ide yang baik untuk memeriksa sistem file yang di-mount baca-tulis.

- Setelah ini, kernel memulai program init (terletak di/sbin/init) di latar belakang (ini akan selalu menjadi proses nomor 1). init melakukan berbagai tugas startup. Hal-hal persisnya bergantung pada cara konfigurasinya; lihat Bagian 2.3.1 untuk informasi lebih lanjut (belum ditulis). Setidaknya akan memulai beberapa daemon latar belakang yang penting.
- init kemudian beralih ke mode multi-pengguna, dan memulai getty untuk konsol virtual dan saluran serial. getty adalah program yang memungkinkan orang masuk melalui konsol virtual dan terminal serial. init juga dapat memulai beberapa program lain, tergantung pada cara konfigurasinya.
- Setelah ini, booting selesai, dan sistem aktif dan berjalan normal.

8.2.1. Sepatah Kata Tentang Bootloader

UNTUK DITAMBAHKAN

Bagian ini akan memberikan gambaran tentang perbedaan antara GRUB dan LILO.

Untuk informasi lebih lanjut tentang LILO, Anda dapat membaca <http://www.tldp.org/HOWTO/LILO.html>

Untuk informasi lebih lanjut tentang GRUB, Anda dapat mengunjungi <http://www.gnu.org/software/grub/grub.html>

8.3. Lebih lanjut tentang shutdown

Penting untuk mengikuti prosedur yang benar saat Anda mematikan sistem Linux. Jika Anda gagal melakukannya, sistem file Anda mungkin akan menjadi sampah dan file mungkin akan diacak. Ini karena

Panduan Administrator Sistem Linux

Linux memiliki cache disk yang tidak akan menulis sesuatu ke disk sekaligus, tetapi hanya pada interval tertentu. Ini sangat meningkatkan kinerja tetapi juga berarti bahwa jika Anda mematikan daya secara tiba-tiba, cache mungkin menyimpan banyak data dan bahwa apa yang ada di disk mungkin bukan sistem file yang berfungsi penuh (karena hanya beberapa hal yang telah ditulis ke disk).

Alasan lain untuk tidak membalik sakelar daya adalah karena dalam sistem multi-tugas, ada banyak hal yang terjadi di latar belakang, dan mematikan daya bisa sangat berbahaya. Dengan menggunakan urutan shutdown yang tepat, Anda memastikan bahwa semua proses latar belakang dapat menyimpan datanya. Perintah untuk mematikan sistem Linux dengan benar adalah shutdown. Biasanya digunakan dalam salah satu dari dua cara.

Jika Anda menjalankan sistem di mana Anda adalah satu-satunya pengguna, cara biasa menggunakan shutdown adalah dengan menutup semua program yang sedang berjalan, logout di semua konsol virtual, login sebagai root pada salah satunya (atau tetap login sebagai root jika Anda sudah ada, tetapi Anda harus mengubah ke folder home root atau folder root, untuk menghindari masalah dengan unmount), kemudian berikan perintah shutdown h sekarang (pengganti sekarang dengan tanda tambah dan angka dalam hitungan menit jika Anda menginginkan penundaan, meskipun biasanya tidak pada sistem pengguna tunggal).

Atau, jika sistem Anda memiliki banyak pengguna, gunakan perintah shutdown h +time message, dimanawaktu adalah waktu dalam menit sampai sistem dihentikan, dan pesan adalah penjelasan singkat mengapa sistem dimatikan.

```
# shutdown h +10 'Kami akan menginstal
disk baru. Sistem harus
> kembali online dalam tiga jam.'
#
```

Ini akan memperingatkan semua orang bahwa sistem akan dimatikan dalam sepuluh menit, dan bahwa mereka sebaiknya tersesat atau kehilangan data. Peringatan dicetak ke setiap terminal tempat seseorang

```
login, termasuk semua xterms:
Pesan siaran dari root (tty0) Rabu 2 Agustus 01:03:25 1995...

Kami akan menginstal disk baru. Sistem
akan kembali online dalam tiga jam.
Sistem akan TURUN untuk sistem berhenti dalam 10 menit !!
```

Peringatan secara otomatis diulang beberapa kali sebelum boot, dengan interval yang semakin pendek seiring waktu habis.

Ketika shutdown sebenarnya dimulai setelah penundaan, semua sistem file (kecuali yang root) dilepas, proses pengguna (jika ada yang masih login) dimatikan, daemon dimatikan, semua sistem file dilepas, dan umumnya semuanya beres. Setelah selesai, init mencetak pesan bahwa Anda dapat mematikan mesin. Kemudian, dan baru setelah itu, Anda harus menggerakkan jari ke arah sakelar daya.

Terkadang, meskipun jarang pada sistem yang bagus, tidak mungkin untuk mematikan dengan benar. Misalnya, jika kernel panik dan mogok dan terbakar dan umumnya berperilaku buruk, mungkin sama sekali tidak mungkin untuk memberikan perintah baru, maka mematikan dengan benar agak sulit, dan hampir semua yang dapat Anda lakukan adalah berharap tidak ada yang rusak terlalu parah. dan matikan listrik. Jika masalahnya tidak terlalu parah (misalnya, seseorang menekan keyboard Anda dengan kapak), dan kernel serta program pembaruan masih berjalan normal, mungkin ada baiknya menunggu beberapa menit untuk memberikan kesempatan pembaruan untuk menyiram cache buffer, dan hanya memotong daya setelah itu.

Di masa lalu, beberapa orang suka mematikan menggunakan perintah sinkronisasi tiga kali, menunggu I/O disk berhenti, lalu mematikan daya. Jika tidak ada program yang berjalan, ini sama dengan menggunakan

shutdown. Namun, itu tidak meng-unmount sistem file apa pun dan ini dapat menyebabkan masalah dengan flag ``clean filesystem" ext2fs. Metode sinkronisasi rangkap tiga tidak disarankan.

(Jika Anda bertanya-tanya: alasan untuk tiga sinkronisasi adalah bahwa pada hari-hari awal UNIX, ketika perintah diketik secara terpisah, itu biasanya memberikan waktu yang cukup untuk menyelesaikan sebagian besar I/O disk.)

8.4. Mem-boot ulang

Reboot berarti mem-boot sistem lagi. Ini dapat dicapai dengan terlebih dahulu mematikannya sepenuhnya, mematikan daya, dan kemudian menyalakannya kembali. Cara yang lebih sederhana adalah dengan meminta shutdown untuk me-reboot sistem, bukannya hanya menghentikannya. Hal ini dicapai dengan menggunakan `r` pilihan untuk shutdown, misalnya dengan memberikan perintah `shutdown r now`.

Sebagian besar sistem Linux menjalankan shutdown atau sekarang ketika `ctrl-alt-del` ditekan pada keyboard. Ini me-reboot sistem. Namun, aksi pada `ctrl-alt-del` dapat dikonfigurasi, dan mungkin lebih baik membiarkan beberapa penundaan sebelum reboot pada mesin multiuser. Sistem yang secara fisik dapat diakses oleh siapa saja bahkan mungkin dikonfigurasi untuk tidak melakukan apa pun saat `ctrl-alt-del` ditekan.

8.5. Mode pengguna tunggal

Perintah `shutdown` juga dapat digunakan untuk membawa sistem ke mode pengguna tunggal, di mana tidak ada yang bisa masuk, tetapi root dapat menggunakan konsol. Ini berguna untuk tugas administrasi sistem yang tidak dapat dilakukan saat sistem berjalan normal.

8.6. Floppy boot darurat

Tidak selalu mungkin untuk mem-boot komputer dari hard disk. Misalnya, jika Anda membuat kesalahan dalam mengonfigurasi LILO, Anda mungkin membuat sistem Anda tidak dapat di-boot. Untuk situasi ini, Anda memerlukan cara boot alternatif yang akan selalu berfungsi (selama perangkat kerasnya berfungsi). Untuk PC biasa, ini berarti booting dari floppy drive.

Sebagian besar distribusi Linux memungkinkan seseorang untuk membuat disket boot darurat selama instalasi. Ini adalah ide yang baik untuk melakukan ini. Namun, beberapa disk boot tersebut hanya berisi kernel, dan menganggap Anda akan menggunakan program pada disk instalasi distribusi untuk memperbaiki masalah apa pun yang Anda miliki. Terkadang program tersebut tidak cukup; misalnya, Anda mungkin harus memulihkan beberapa file dari cadangan yang dibuat dengan perangkat lunak bukan pada disk instalasi.

Jadi, mungkin perlu untuk membuat floppy root kustom juga. Bootdisk HOWTO oleh Graham Chapman berisi instruksi untuk melakukan ini. Anda dapat menemukan HOWTO ini di <http://www.tldp.org/HOWTO/Bootdisk-HOWTO/index.html>. Anda harus, tentu saja, ingat untuk selalu memperbarui boot darurat dan disket root Anda.

Anda tidak dapat menggunakan floppy drive yang Anda gunakan untuk memasang floppy root untuk hal lain. Ini bisa merepotkan jika Anda hanya memiliki satu floppy drive. Namun, jika Anda memiliki cukup memori, Anda dapat mengkonfigurasi floppy boot Anda untuk memuat disk root ke ramdisk (kernel boot floppy perlu dikonfigurasi secara khusus untuk ini). Setelah floppy root dimuat ke dalam ramdisk, floppy drive bebas untuk ~~memasang disk lain~~

Bab 9. init

"Uuno on numero yksi" (Slogan untuk serangkaian film Finlandia.)

Bab ini menjelaskan proses `init`, yang merupakan proses tingkat pengguna pertama yang dimulai oleh kernel. `init` memiliki banyak tugas penting, seperti memulai `getty` (agar pengguna dapat login), mengimplementasikan run level, dan menangani proses yatim piatu. Bab ini menjelaskan bagaimana `init` dikonfigurasi dan bagaimana Anda dapat menggunakan level run yang berbeda.

9.1. `init` datang lebih dulu

`init` adalah salah satu program yang sangat penting untuk pengoperasian sistem Linux, tetapi sebagian besar masih dapat Anda abaikan. Distribusi Linux yang baik akan datang dengan konfigurasi untuk `init` yang akan bekerja untuk sebagian besar sistem, dan pada sistem ini tidak ada yang perlu Anda lakukan tentang `init`. Biasanya, Anda hanya perlu mengkhawatirkan `init` jika Anda menghubungkan terminal serial, modem dial-in (bukan dial-out), atau jika Anda ingin mengubah run level default.

Ketika kernel telah memulai dirinya sendiri (telah dimuat ke dalam memori, telah mulai berjalan, dan telah menginisialisasi semua driver perangkat dan struktur data dan sebagainya), kernel menyelesaikan bagiannya sendiri dari proses boot dengan memulai program tingkat pengguna, `init`. Jadi, `init` selalu merupakan proses pertama (nomor prosesnya selalu 1).

Kernel mencari `init` di beberapa lokasi yang secara historis digunakan untuk itu, tetapi lokasi yang tepat untuk itu (pada sistem Linux) adalah `/sbin/init`. Jika kernel tidak dapat menemukan `init`, kernel mencoba untuk menjalankan `/bin/sh`, dan jika itu juga gagal, startup sistem akan gagal.

Ketika `init` dimulai, ia menyelesaikan proses boot dengan melakukan sejumlah tugas administratif, seperti memeriksa sistem file, membersihkan `/tmp`, memulai berbagai layanan, dan memulai `getty` untuk setiap terminal dan konsol virtual tempat pengguna seharusnya dapat masuk (lihat Bab 10).

Setelah sistem dinyalakan dengan benar, `init` me-restart `getty` untuk setiap terminal setelah pengguna keluar (sehingga pengguna berikutnya dapat masuk). `init` juga mengadopsi proses anak yatim: ketika sebuah proses memulai proses anak dan mati sebelum anaknya, anak tersebut segera menjadi anak `init`. Hal ini penting untuk berbagai alasan teknis, tetapi baik untuk mengetahuinya, karena memudahkan untuk memahami daftar proses dan grafik pohon proses. Ada beberapa varian `init` yang tersedia. Sebagian besar distribusi Linux menggunakan `sysvinit` (ditulis oleh Miquel van Smoorenburg), yang didasarkan pada desain `init` System V. Versi BSD dari Unix memiliki `init` yang berbeda. Perbedaan utama adalah run level: Sistem V memilikinya, BSD tidak (setidaknya secara tradisional). Perbedaan ini tidak esensial. Kami akan melihat `sysvinit` saja.

9.2. Mengonfigurasi `init` untuk memulai `getty`: `the/etc/inittab` mengajukan

Saat dijalankan, `init` membaca `/etc/inittab` berkas konfigurasi. Saat sistem sedang berjalan, sistem akan membaca ulang, jika mengirim sinyal HUP (membunuh HUP 1); fitur ini membuat sistem tidak perlu mem-boot untuk membuat perubahan pada konfigurasi `init` berlaku.

`/etc/inittab` filenya agak rumit. Kita akan mulai dengan kasus sederhana dalam mengonfigurasi garis `getty`. Garis masuk `/etc/inittab` terdiri dari empat bidang yang dibatasi titik dua:

```
id:runlevel:aksi:proses
```

Bidang dijelaskan di bawah ini. Selain itu, `/etc/inittab` dapat berisi baris kosong, dan baris yang dimulai dengan tanda angka (^ #); ini keduanya diabaikan.

Indo

Ini mengidentifikasi baris dalam file. Untuk baris `getty`, ini menentukan terminal yang dijalankannya (karakter setelah `/dev/tty` dalam nama file perangkat). Untuk baris lain, tidak masalah (kecuali untuk batasan panjang), tetapi harus unik.

runlevel

Tingkat lari garis harus dipertimbangkan. Level run diberikan sebagai satu digit, tanpa pembatas. (Level lari dijelaskan di bagian berikutnya.) **tindakan**

Tindakan apa yang harus diambil oleh garis, misalnya, `muncul kembali` untuk menjalankan perintah di bidang berikutnya lagi, ketika keluar, atau `sekali` untuk menjalankannya sekali saja.

Perintah untuk menjalankan.

Untuk memulai `getty` di terminal virtual pertama (`/dev/tty1`), di semua level run multi-pengguna normal (2-5), seseorang akan menulis baris berikut:

```
1:2345:respawn:/sbin/getty 9600 tty1
```

Bidang pertama mengatakan bahwa ini adalah garis untuk `/dev/tty1`. Bidang kedua mengatakan bahwa itu berlaku untuk menjalankan level 2, 3, 4, dan 5. Kolom ketiga berarti perintah harus dijalankan lagi, setelah keluar (agar bisa login, login keluar, lalu masuk lagi). Kolom terakhir adalah perintah yang menjalankan `getty` di terminal virtual pertama.

Versi `getty` yang berbeda dijalankan secara berbeda. Lihat halaman manual Anda, dan pastikan itu adalah halaman manual yang benar.

Jika Anda ingin menambahkan terminal atau menyambungkan saluran modem ke sistem, Anda dapat menambahkan lebih banyak saluran ke `/etc/inittab`, satu untuk setiap terminal atau saluran dial-in. Untuk lebih jelasnya, lihat halaman manual `init`, `inittab`, dan `getty`.

Jika sebuah perintah gagal saat dimulai, dan `init` dikonfigurasi `kemengulang kembali` itu, itu akan menggunakan banyak sumber daya sistem: `init` memulainya, gagal, `init` memulainya, gagal, `init` memulainya, gagal, dan seterusnya, *ad infinitum*. Untuk mencegah hal ini, `init` akan melacak seberapa sering ia memulai ulang suatu perintah, dan jika frekuensinya meningkat menjadi tinggi, ia akan menunda selama lima menit ~~sebelum memulai kembali.~~

9.3. Jalankan level

Run level adalah keadaan `init` dan keseluruhan sistem yang mendefinisikan layanan sistem apa yang sedang beroperasi. Tingkat lari diidentifikasi dengan angka. Beberapa administrator sistem menggunakan run level untuk menentukan subsistem mana yang berfungsi, misalnya apakah X sedang berjalan, apakah jaringan beroperasi, dan seterusnya. Lainnya memiliki semua subsistem yang selalu berjalan atau memulai dan menghentikannya satu per satu, tanpa mengubah level run, karena run level terlalu kasar untuk mengontrol sistem mereka. Anda perlu memutuskan sendiri, tetapi mungkin paling mudah untuk mengikuti cara distribusi Linux Anda melakukan sesuatu.

Panduan Administrator Sistem Linux

Tabel berikut mendefinisikan bagaimana sebagian besar Distribusi Linux menentukan tingkat run yang berbeda. Namun, run-level 2 hingga 5 dapat dimodifikasi sesuai selera Anda.

Tabel 9-1. Jalankan nomor level

0	Menghentikan sistem.
1	Mode pengguna tunggal (untuk administrasi khusus).
2	Multiuser Lokal dengan Jaringan tetapi tanpa layanan jaringan (seperti NFS)
3	Multiuser Penuh dengan Jaringan
4	Tidak digunakan
5	Multiuser Penuh dengan Jaringan dan X Windows (GUI)
6	Menyalakan ulang.

Layanan yang dimulai pada runtime tertentu ditentukan oleh konten dari berbagai `rcN.d` folder. Sebagian besar distribusi menemukan folder ini baik di `/etc/init.d/rcN.d` atau `/etc/rcN.d`. (Ganti N dengan nomor run-level.)

Di setiap run-level Anda akan menemukan serangkaian tautan `if` yang menunjuk ke skrip start-up yang terletak di `/etc/init.d`. Nama-nama tautan ini semuanya dimulai sebagai `K` atau `S`, diikuti dengan angka. Jika nama tautan dimulai dengan `S`, maka itu menunjukkan bahwa layanan akan dimulai ketika Anda masuk ke level run itu. Jika nama tautan dimulai dengan `K`, layanan akan dimatikan (jika berjalan).

Angka yang mengikuti `K` atau `S` menunjukkan urutan skrip akan dijalankan. Berikut adalah contoh dari apa yang

`/etc/init.d/rc3` mungkin terlihat seperti.

```
# ls -l /etc/init.d/rc3.d
lrwxrwxrwx 1 root root 10 2004-11-29 22:09 K12nfsboot > ../nfsboot
lrwxrwxrwx 1 root root 6 2005-03-29 13:42 K15xdm > ../xdm lrwxrwxrwx
1 root root 9 2004-11-29 22:08 S01pcmcia > ../pcmcia lrwxrwxrwx 1
root root 9 2004-11-29 22:06 S01random > ../random
lrwxrwxrwx 1 root root 11 2005-03-01 11:56 S02firewall > ../firewall
lrwxrwxrwx 1 root root 10 2004-11-29 22:34 S05network > ../network
lrwxrwxrwx 1 root root 9 2004-11-29 22:07 S06syslog > ../syslog
lrwxrwxrwx 1 root root 10 2004-11-29 22:09 S08portmap > ../portmap
lrwxrwxrwx 1 root root 9 2004-11-29 22:07 S08resmgr > ../resmgr
lrwxrwxrwx 1 root root 6 2004-11-29 22:09 S10nfs > ../nfs
lrwxrwxrwx 1 root root 12 2004-11-29 22:40 S12alsound > ../alsasound
lrwxrwxrwx 1 root root 8 2004-11-29 22:09 S12fbset > ../fbset lrwxrwxrwx
1 root root 7 2004-11-29 22:10 S12sshd > ../sshd
lrwxrwxrwx 1 root root 8 2005-02-01 09:24 S12xntpd > ../xntpd
lrwxrwxrwx 1 root root 7 2004-12-02 20:34 S13cups > ../cups
lrwxrwxrwx 1 root root 6 2004-11-29 22:09 S13kbd > ../kbd
lrwxrwxrwx 1 root root 13 2004-11-29 22:10 S13powersaved > ../powersaved
lrwxrwxrwx 1 root root 9 2004-11-29 22:09 S14hwscan > ../hwscan lrwxrwxrwx
1 root root 7 2004-11-29 22:10 S14nscd > ../nscd
lrwxrwxrwx 1 root root 10 2004-11-29 22:10 S14postfix > ../postfix
lrwxrwxrwx 1 root root 6 2005-02-04 13:27 S14smb > ../smb lrwxrwxrwx
1 root root 7 2004-11-29 22:10 S15cron > ../cron lrwxrwxrwx 1 root
root 8 2004-12-22 20:35 S15smbfs > ../smbfs
```

Bagaimana run level mulai dikonfigurasi di `/etc/inittab` dengan garis seperti berikut:

```
l2:2:tunggu:/etc/init.d/rc 2
```

Bidang pertama adalah label arbitrer, yang kedua berarti ini berlaku untuk run level 2. Bidang ketiga berarti `init` harus menjalankan perintah di bidang keempat sekali, ketika run level dimasukkan, dan `init` harus

Panduan Administrator Sistem Linux

menunggunya untuk menyelesaikan. `/etc/init.d/rc` command menjalankan perintah apa pun yang diperlukan untuk memulai dan menghentikan layanan untuk memasuki run level 2.

Perintah di bidang keempat melakukan semua kerja keras untuk menyiapkan run level. Ini memulai layanan yang belum berjalan, dan menghentikan layanan yang seharusnya tidak berjalan di tingkat proses baru lagi. Apa persisnya perintah itu, dan bagaimana run level dikonfigurasi, tergantung pada distribusi Linux.

Saat init dimulai, ia mencari baris di `/etc/inittab` yang menentukan level run default:

```
id:2:initdefault:
```

Anda dapat meminta init untuk pergi ke run level non-default saat startup dengan memberikan kernel argumen baris perintah dari `Lajangatau` keadaan darurat. Argumen baris perintah kernel dapat diberikan melalui LILO, misalnya. Ini memungkinkan Anda untuk memilih mode pengguna tunggal (menjalankan level 1).

Saat sistem sedang berjalan, perintah `telinit` dapat mengubah level run. Saat level lari diubah, `init` menjalankan perintah yang relevan dari `/etc/inittab`.

9.4. Konfigurasi khusus di `/etc/inittab`

`/etc/inittab` memiliki beberapa fitur khusus yang memungkinkan init bereaksi terhadap keadaan khusus. Fitur khusus ini ditandai dengan kata kunci khusus di kolom ketiga. Beberapa contoh:

unggu sebentar

Mengizinkan init mematikan sistem, saat daya mati. Ini mengasumsikan penggunaan UPS, dan perangkat lunak yang mengawasi UPS dan menginformasikan init bahwa listrik mati.

ctrlaltdel

Memungkinkan init untuk mem-boot ulang sistem, saat pengguna menekan `ctrl-alt-del` pada keyboard konsol. Catatan

bahwa administrator sistem dapat mengonfigurasi reaksi terhadap `ctrl-alt-del` menjadi sesuatu yang lain, misalnya, diabaikan, jika sistem berada di lokasi publik. (Atau untuk memulai `nethack`.)

sysinit

Perintah untuk dijalankan saat sistem di-boot. Perintah ini biasanya membersihkan `/tmp`, Misalnya.

Daftar di atas tidak lengkap. Lihat `Andainittab` halaman manual untuk semua kemungkinan, dan untuk detail tentang cara menggunakan yang di atas.

9.5. Booting dalam mode pengguna tunggal

Run level yang penting adalah mode pengguna tunggal (run level 1), di mana hanya administrator sistem yang menggunakan mesin dan sesedikit mungkin layanan sistem, termasuk login, yang sedang berjalan. Mode pengguna tunggal diperlukan untuk beberapa tugas administratif, seperti menjalankan `fsck` di `a/usr` partisi, karena ini mengharuskan partisi di-unmount, dan itu tidak dapat terjadi, kecuali hampir semua layanan sistem dimatikan.

Sistem yang sedang berjalan dapat dibawa ke mode pengguna tunggal dengan menggunakan `telinit` untuk meminta run level 1. Saat boot, dapat dimasukkan dengan memberikan kata `Lajangatau` keadaan darurat pada baris perintah kernel: kernel memberikan baris perintah ke `init` juga, dan `init` memahami dari kata itu bahwa ia tidak boleh menggunakan run level default. (Baris perintah kernel dimasukkan dengan cara yang bergantung pada cara Anda mem-boot sistem.)

Panduan Administrator Sistem Linux

Boot ke mode pengguna tunggal terkadang diperlukan agar seseorang dapat menjalankan fsck dengan tangan, sebelum apa pun dipasang atau menyentuh yang rusak /usr partisi (aktivitas apa pun pada sistem file yang rusak kemungkinan besar akan lebih merusaknya, jadi fsck harus dijalankan sesegera mungkin).

Skrip boot yang dijalankan init akan otomatis masuk ke mode single user, jika fsck otomatis saat booting gagal. Ini adalah upaya untuk mencegah sistem menggunakan sistem file yang rusak sehingga fsck tidak dapat memperbaikinya secara otomatis. Kerusakan seperti itu relatif jarang, dan biasanya melibatkan hard disk yang rusak atau rilis kernel eksperimental, tetapi ada baiknya untuk dipersiapkan.

Sebagai tindakan keamanan, sistem yang dikonfigurasi dengan benar akan meminta kata sandi root sebelum memulai shell dalam mode pengguna tunggal. Jika tidak, akan mudah untuk memasukkan baris yang sesuai ke LILO untuk masuk sebagai root. (Ini akan pecah jika /etc/passwd telah rusak oleh masalah sistem file, tentu saja, dan dalam hal ini Anda sebaiknya memiliki disket boot.)

Bab 10. Masuk Dan Keluar

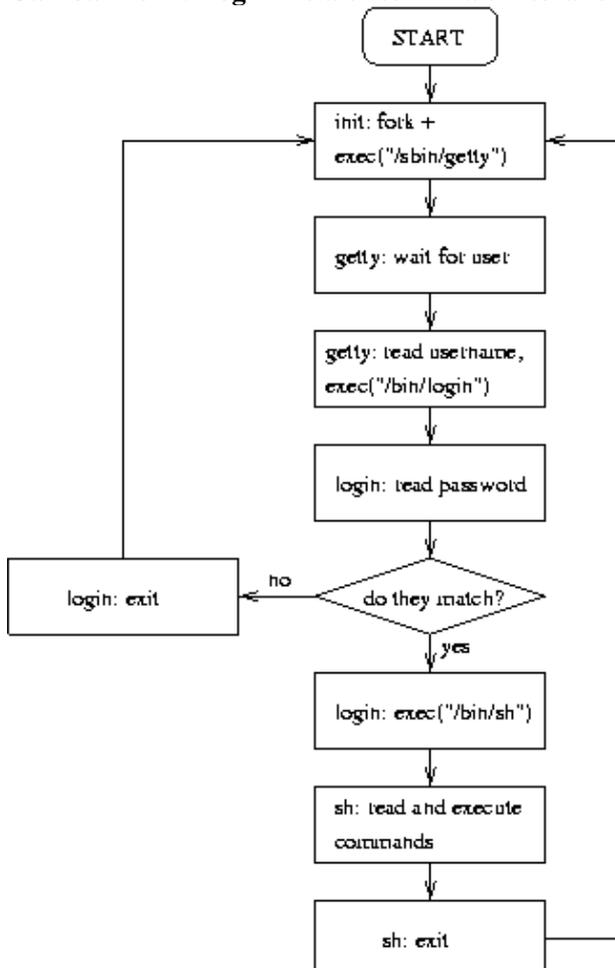
"Saya tidak peduli untuk menjadi bagian dari klub yang menerima orang-orang seperti saya sebagai anggota." (Groucho Marx)

Bagian ini menjelaskan apa yang terjadi ketika pengguna masuk atau keluar. Berbagai interaksi proses latar belakang, file log, file konfigurasi, dan sebagainya dijelaskan secara rinci.

10.1. Login melalui terminal

Bagian 2.3.2 menunjukkan bagaimana proses masuk melalui terminal. Pertama, `init` memastikan ada program `getty` untuk koneksi terminal (atau konsol). `getty` mendengarkan di terminal dan menunggu pengguna memberi tahu bahwa dia siap untuk masuk (ini biasanya berarti pengguna harus mengetik sesuatu). Saat ia melihat pengguna, `getty` mengeluarkan pesan selamat datang (disimpan di `/etc/issue`), dan meminta nama pengguna, dan akhirnya menjalankan `login` program. `login` mendapatkan nama pengguna sebagai parameter, dan meminta kata sandi kepada pengguna. Jika ini cocok, `login` memulai shell yang dikonfigurasi untuk pengguna; selain itu hanya keluar dan menghentikan proses (mungkin setelah memberi pengguna kesempatan lain untuk memasukkan nama pengguna dan kata sandi). `init` memperhatikan bahwa proses dihentikan, dan memulai `getty` baru untuk terminal.

Gambar 10–1. Login melalui terminal: interaksi `init`, `getty`, `login`, dan shell.



Perhatikan bahwa satu-satunya proses baru adalah yang dibuat oleh `init` (menggunakan `arpup` panggilan sistem); `getty` dan `login` hanya menggantikan program yang sedang berjalan (menggunakan `eksekutif` panggilan sistem).

Program terpisah, untuk memperhatikan pengguna, diperlukan untuk saluran serial, karena dapat (dan biasanya) rumit untuk diperhatikan ketika terminal menjadi aktif. `getty` juga menyesuaikan dengan kecepatan dan pengaturan koneksi lainnya, yang penting terutama untuk koneksi dial-in, di mana parameter ini dapat berubah dari panggilan ke panggilan.

Ada beberapa versi `getty` dan `init` yang digunakan, semua dengan poin baik dan buruknya. Sebaiknya pelajari tentang versi pada sistem Anda, dan juga tentang versi lainnya (Anda dapat menggunakan Peta Perangkat Lunak Linux untuk mencarinya). Jika Anda tidak memiliki panggilan, Anda mungkin tidak perlu khawatir tentang `getty`, tetapi `init` tetap penting.

10.2. Login melalui jaringan

Dua komputer dalam jaringan yang sama biasanya dihubungkan melalui satu kabel fisik. Ketika mereka berkomunikasi melalui jaringan, program di setiap komputer yang mengambil bagian dalam komunikasi dihubungkan melalui koneksi virtual, semacam kabel imajiner. Sejauh program di kedua ujung koneksi virtual yang bersangkutan, mereka memiliki monopoli atas kabel mereka sendiri. Namun, karena kabel tidak nyata, hanya imajiner, sistem operasi kedua komputer dapat memiliki beberapa koneksi virtual berbagi kabel fisik yang sama. Dengan cara ini, hanya dengan menggunakan satu kabel, beberapa program dapat berkomunikasi tanpa harus mengetahui atau peduli tentang

komunikasi lainnya. Bahkan dimungkinkan untuk memiliki beberapa komputer menggunakan kabel yang sama; koneksi virtual ada di antara dua komputer, dan komputer lain mengabaikan koneksi yang tidak mereka ikuti.

Itu adalah deskripsi realitas yang rumit dan terlalu abstrak. Namun, mungkin cukup baik untuk memahami alasan penting mengapa login jaringan agak berbeda dari login normal. Koneksi virtual dibuat ketika ada dua program di komputer berbeda yang ingin berkomunikasi. Karena pada prinsipnya memungkinkan untuk masuk dari komputer mana pun dalam jaringan ke komputer lain, ada sejumlah besar potensi komunikasi virtual. Karena itu, tidak praktis untuk memulai `getty` untuk setiap login potensial.

Ada satu proses `inetd` (sesuai dengan `getty`) yang menangani semua login jaringan. Ketika pemberitahuan login jaringan masuk (yaitu, pemberitahuan bahwa itu mendapat koneksi virtual baru ke beberapa komputer lain), itu memulai proses baru untuk menangani login tunggal itu. Proses asli tetap dan terus mendengarkan login baru.

Untuk membuat segalanya sedikit lebih rumit, ada lebih dari satu protokol komunikasi untuk login jaringan. Dua yang paling penting adalah `telnet` dan `rlogin`. Selain login, masih banyak koneksi virtual lain yang dapat dibuat (untuk FTP, Gopher, HTTP, dan layanan jaringan lainnya). Itu tidak akan efektif untuk memiliki proses terpisah yang mendengarkan jenis koneksi tertentu, jadi hanya ada satu pendengar yang dapat mengenali jenis koneksi dan dapat memulai jenis program yang benar untuk menyediakan layanan. Pendengar tunggal ini disebut `inetd`; lihat Panduan Administrator Jaringan Linux untuk informasi lebih lanjut.

10.3. Login apa

Program login menangani autentikasi pengguna (memastikan bahwa nama pengguna dan sandi cocok), dan menyiapkan lingkungan awal bagi pengguna dengan menyetel izin untuk jalur serial dan memulai shell.

Panduan Administrator Sistem Linux

Bagian dari pengaturan awal adalah mengeluarkan konten file `file/etc/motd` (singkatan dari pesan hari ini) dan memeriksa surat elektronik. Ini dapat dinonaktifkan dengan membuat file bernama `.hushlog` di folder home pengguna.

Jika file `file/etc/nologin` ada, login dinonaktifkan. File itu biasanya dibuat oleh shutdown dan kerabat. login memeriksa file ini, dan akan menolak untuk menerima login jika ada. Jika memang ada, login mengeluarkan isinya ke terminal sebelum ditutup.

Gabung mencatat semua upaya login yang gagal dalam file log sistem (melalui `syslog`). Itu juga mencatat semua login dengan root. Kedua hal ini dapat berguna saat melacak penyusup.

Saat ini orang yang masuk terdaftar di `file/var/run/utmp`. File ini hanya valid hingga sistem di-boot ulang atau dimatikan berikutnya; itu dihapus ketika sistem di-boot. Ini mencantumkan setiap pengguna dan terminal (atau koneksi jaringan) yang dia gunakan, bersama dengan beberapa informasi berguna lainnya. `Who`, `w`, dan perintah serupa lainnya terlihat di `utmp` untuk melihat siapa yang login.

Semua login yang berhasil dicatat ke dalam `file/var/log/wtmp`. File ini akan berkembang tanpa batas, sehingga harus dibersihkan secara berkala, misalnya dengan melakukan tugas cron mingguan untuk menghapusnya. Perintah terakhir menelusuri `wtmp`.

Keduanya `utmp` dan `wtmp` dalam format biner (lihat `utmp` halaman manual); sayangnya tidak nyaman untuk memeriksa tanpa program khusus.

10.4. X dan xdm

XXX X mengimplementasikan login melalui xdm; juga: xterm ls

UNTUK DITAMBAHKAN

10.5. Kontrol akses

Basis data pengguna secara tradisional terkandung dalam `/etc/passwd` mengajukan. Beberapa sistem menggunakan kata sandi bayangan, dan telah memindahkan kata sandi ke `/etc/shadow`. Situs dengan banyak komputer yang berbagi akun menggunakan NIS atau metode lain untuk menyimpan database pengguna; mereka mungkin juga secara otomatis menyalin database dari satu lokasi pusat ke semua komputer lain.

Basis data pengguna tidak hanya berisi kata sandi, tetapi juga beberapa informasi tambahan tentang pengguna, seperti nama asli mereka, folder home, dan shell login. Informasi lain ini perlu dipublikasikan, sehingga siapa pun dapat membacanya. Oleh karena itu password disimpan terenkripsi. Ini memang memiliki kelemahan bahwa siapa pun yang memiliki akses ke kata sandi terenkripsi dapat menggunakan berbagai metode kriptografi untuk menebaknya, tanpa mencoba untuk benar-benar masuk ke komputer. Sandi bayangan mencoba menghindari hal ini dengan memindahkan sandi ke file lain, yang hanya dapat dibaca oleh root (kata sandi masih disimpan terenkripsi). Namun, menginstal kata sandi bayangan nanti ke sistem yang tidak mendukungnya bisa jadi sulit.

Dengan atau tanpa kata sandi, penting untuk memastikan bahwa semua kata sandi dalam suatu sistem baik, yaitu tidak mudah ditebak. Program crack dapat digunakan untuk memecahkan kata sandi; kata sandi apa pun yang dapat ditemukan menurut definisi bukan kata sandi yang baik. Meskipun crack dapat dijalankan oleh penyusup, crack juga dapat dijalankan oleh administrator sistem untuk menghindari kata sandi yang buruk. Kata sandi yang baik juga dapat diterapkan oleh program `passwd`; ini sebenarnya lebih efektif dalam Siklus CPU, karena meretas kata sandi membutuhkan banyak komputasi.

Basis data grup pengguna disimpan di `/etc/group`; untuk sistem dengan sandi bayangan, mungkin ada a `/etc/shadow.group`.

root biasanya tidak dapat masuk melalui sebagian besar terminal atau jaringan, hanya melalui terminal yang terdaftar di `/etc/securetty` mengajukan. Ini membuatnya perlu untuk mendapatkan akses fisik ke salah satu terminal ini. Namun, dimungkinkan untuk masuk melalui terminal apa pun seperti pengguna lain, dan menggunakan perintah `su` untuk menjadi root.

10.6. Pembukaan cangkang

Ketika shell login interaktif dimulai, shell secara otomatis mengeksekusi satu atau lebih file yang telah ditentukan sebelumnya. Shell yang berbeda mengeksekusi file yang berbeda; lihat dokumentasi setiap shell untuk informasi lebih lanjut.

Sebagian besar shell pertama-tama menjalankan beberapa file global, misalnya, shell Bourne (`/bin/sh`) dan turunannya dijalankan

`/etc/profil`; selain itu, mereka mengeksekusi `.Profile` di folder home pengguna. `/etc/profil` memungkinkan administrator sistem untuk mengatur lingkungan pengguna umum, terutama dengan menyetel `JALUR` untuk memasukkan folder perintah lokal selain yang normal. Di samping itu, `.Profile` memungkinkan pengguna untuk menyesuaikan lingkungan dengan selera mereka sendiri dengan `meskipun demikian, jika perlu, lingkungan default.`

Bab 11. Mengelola akun pengguna

"Kesamaan sysadmin dan pengedar narkoba: keduanya mengukur barang di Ks, dan keduanya memiliki pengguna." (Lelucon komputer lama yang lelah.)

Bab ini menjelaskan cara membuat akun pengguna baru, cara mengubah properti akun tersebut, dan cara menghapus akun. Sistem Linux yang berbeda memiliki alat yang berbeda untuk melakukan ini.

11.1. Apa itu akun?

Ketika komputer digunakan oleh banyak orang, biasanya perlu untuk membedakan antara pengguna, misalnya, agar file pribadi mereka dapat disimpan secara pribadi. Ini penting bahkan jika komputer hanya dapat digunakan oleh satu orang pada satu waktu, seperti kebanyakan komputer mikro. Jadi, setiap pengguna diberi nama pengguna yang unik, dan nama itu digunakan untuk masuk.

Namun, ada lebih banyak pengguna daripada sekadar nama. Akun adalah semua file, sumber daya, dan informasi milik satu pengguna. Istilah ini mengisyaratkan bank, dan dalam sistem komersial setiap akun biasanya memiliki sejumlah uang yang melekat padanya, dan uang itu hilang dengan kecepatan yang berbeda tergantung pada seberapa banyak pengguna menekankan sistem. Misalnya, ruang disk mungkin memiliki harga per megabyte dan hari, dan waktu pemrosesan mungkin memiliki harga per detik.

11.2. Membuat pengguna

Kernel Linux sendiri memperlakukan pengguna hanyalah angka. Setiap pengguna diidentifikasi dengan bilangan bulat unik, id pengguna atau uid, karena angka lebih cepat dan lebih mudah diproses oleh komputer daripada nama tekstual. Basis data terpisah di luar kernel memberikan nama tekstual, nama pengguna, untuk setiap id pengguna. Basis data juga berisi informasi tambahan.

Untuk membuat pengguna, Anda perlu menambahkan informasi tentang pengguna ke database pengguna, dan membuat folder home untuknya. Mungkin juga perlu untuk mendidik pengguna, dan menyiapkan lingkungan awal yang cocok untuknya.

Sebagian besar distribusi Linux dilengkapi dengan program untuk membuat akun. Ada beberapa program seperti itu yang tersedia. Dua alternatif baris perintah adalah `adduser` dan `useradd`; mungkin ada alat GUI juga. Apa pun programnya, hasilnya hanya sedikit jika ada pekerjaan manual yang harus dilakukan. Sekalipun detailnya banyak dan rumit, program-program ini membuat semuanya tampak sepele. Namun, Bagian 11.2.4 menjelaskan bagaimana melakukannya dengan tangan.

11.2.1. `/etc/passwd` dan file informatif lainnya

Basis data pengguna dasar dalam sistem Unix adalah file teks, `/etc/passwd` (disebut file kata sandi), yang mencantumkan semua nama pengguna yang valid dan informasi terkaitnya. File memiliki satu baris per nama pengguna, dan dibagi menjadi tujuh bidang yang dibatasi titik dua:

- Nama pengguna.
- Sebelumnya ini adalah tempat penyimpanan kata sandi pengguna.
- ID pengguna numerik.
- ID grup numerik.

- Nama lengkap atau deskripsi akun lainnya.
- Folder rumah.
- Shell Login (program untuk dijalankan saat login).

Formatnya dijelaskan lebih detail di `passwd` halaman manual.

Sebagian besar sistem Linux menggunakan kata sandi bayangan. Seperti yang disebutkan, kata sandi sebelumnya disimpan di

`/etc/passwd` mengajukan. Metode penyimpanan kata sandi yang lebih baru ini: kata sandi terenkripsi disimpan dalam file terpisah, `/etc/bayangan`, yang hanya dapat dibaca oleh root. `/etc/passwd` file hanya berisi penanda khusus di bidang kedua. Setiap program yang perlu memverifikasi pengguna adalah setuid, dan karena itu dapat mengakses file sandi bayangan. Program normal, yang hanya menggunakan bidang lain dalam file kata sandi, tidak bisa mendapatkan kata sandi.

11.2.2. Memilih pengguna numerik dan id grup

Pada sebagian besar sistem, tidak masalah apa pengguna numerik dan id grup, tetapi jika Anda menggunakan sistem file Jaringan (NFS), Anda harus memiliki uid dan gid yang sama di semua sistem. Ini karena NFS juga mengidentifikasi pengguna dengan uid numerik. Jika Anda tidak menggunakan NFS, Anda dapat membiarkan alat pembuatan akun Anda memilihnya secara otomatis.

Jika Anda menggunakan NFS, Anda harus menemukan mekanisme untuk menyinkronkan informasi akun. Salah satu alternatifnya adalah sistem NIS (lihat panduan admin-jaringan XXX).

Namun, Anda harus mencoba untuk menghindari penggunaan kembali uid numerik (dan nama pengguna tekstual), karena pemilik baru uid (atau nama pengguna) mungkin mendapatkan akses ke file pemilik lama (atau surat, atau apa pun).

11.2.3. Lingkungan awal: `/etc/skel`

Ketika folder home untuk pengguna baru dibuat, ini diinisialisasi dengan file dari `/etc/skel` folder. Administrator sistem dapat membuat file di `/etc/skel` yang akan menyediakan lingkungan default yang bagus untuk pengguna. Misalnya, dia mungkin membuat `/etc/skel/.profile` yang mengatur variabel lingkungan EDITOR ke beberapa editor yang ramah terhadap pengguna baru.

Namun, biasanya yang terbaik adalah mencoba untuk tetap `/etc/skel` sekecil mungkin, karena hampir tidak mungkin untuk memperbaiki file pengguna yang ada. Misalnya, jika nama editor ramah berubah, semua pengguna yang ada harus mengedit `.Profile`. Administrator sistem dapat mencoba melakukannya secara otomatis, dengan skrip, tetapi itu hampir pasti akan merusak file seseorang.

Bila memungkinkan, lebih baik untuk menempatkan konfigurasi global ke dalam file global, seperti: `/etc/profile`. Dengan cara ini dimungkinkan untuk memperbaruinya tanpa merusak pengaturan pengguna sendiri.

11.2.4. Membuat pengguna dengan tangan

Untuk membuat akun baru secara manual, ikuti langkah-langkah berikut:

- **Sunting** `/etc/passwd` dengan `vipw` dan tambahkan baris baru untuk akun baru. Hati-hati dengan sintaks. Jangan mengedit langsung dengan editor! `vipw` mengunci file, sehingga perintah lain tidak

Panduan Administrator Sistem Linux

akan mencoba memperbaruinya secara bersamaan. Anda harus membuat bidang kata sandi menjadi `*`, sehingga tidak mungkin untuk masuk.

- Demikian pula, `edit/etc/grup` dengan `vigr`, jika Anda perlu membuat grup baru juga.
- Buat folder home pengguna dengan `mkdir`.
- Salin file dari `/etc/skel` ke folder home baru.
- Perbaiki kepemilikan dan izin dengan `chown` dan `chmod`. ItuR pilihan yang paling berguna. yang benar izin sedikit berbeda dari satu situs ke situs lainnya, tetapi biasanya perintah berikut melakukan yang benar

hal:

```
cd /home/namapengguna baru
chown R username.group .
chmod R go=u,go-w .
chmod go= .
```

- Atur kata sandi dengan `passwd`.

Setelah Anda mengatur kata sandi di langkah terakhir, akun akan berfungsi. Anda tidak boleh mengaturnya sampai semuanya selesai, jika tidak, pengguna mungkin secara tidak sengaja masuk saat Anda masih menyalin file.

Terkadang perlu membuat akun dummy yang tidak digunakan oleh orang lain. Misalnya, untuk menyiapkan server FTP anonim (sehingga siapa pun dapat mengunduh file darinya, tanpa harus membuat akun terlebih dahulu), Anda perlu membuat akun bernama `ftp`. Dalam kasus seperti itu, biasanya tidak perlu mengatur kata sandi (langkah terakhir di atas). Memang sebaiknya tidak, agar tidak ada yang bisa menggunakan akun tersebut, kecuali mereka yang terlebih dahulu menjadi root, karena root bisa menjadi pengguna mana saja.

11.3. Mengubah properti pengguna

Ada beberapa perintah untuk mengubah berbagai properti akun (yaitu, bidang yang relevan di `/etc/passwd`):

chfn

Ubah bidang nama

chsh

lengkap. Ubah shell login.

passwd

Ubah kata sandi.

Pengguna super dapat menggunakan perintah ini untuk mengubah properti akun mana pun. Pengguna normal hanya dapat mengubah properti akun mereka sendiri. Kadang-kadang mungkin perlu untuk menonaktifkan perintah ini (dengan `chmod`) untuk pengguna normal, misalnya di lingkungan dengan banyak pengguna pemula.

Tugas lain perlu dilakukan dengan tangan. Misalnya, untuk mengubah nama pengguna, Anda perlu mengedit `/etc/passwd` langsung (dengan `vipw`, ingat). Demikian juga, untuk menambah atau menghapus pengguna ke lebih banyak grup, Anda perlu mengedit `/etc/grup` (dengan semangat). Namun, tugas seperti itu cenderung jarang, dan harus dilakukan dengan hati-hati: untuk contoh, jika Anda mengubah nama pengguna, email tidak akan lagi menjangkau pengguna, kecuali jika Anda juga membuat alias email.

11.4. Menghapus pengguna

Panduan Administrator Sistem Linux

Untuk menghapus pengguna, Anda terlebih dahulu menghapus semua filenya, kotak surat, alias surat, tugas cetak, cron dan di tugas, dan semua referensi lain untuk pengguna. Kemudian Anda menghapus baris yang relevan dari `/etc/passwd` dan `/etc/group` (ingat untuk menghapus nama pengguna dari semua grup yang telah ditambahkan). Sebaiknya nonaktifkan akun terlebih dahulu (lihat di bawah), sebelum Anda mulai menghapus barang, untuk mencegah pengguna menggunakan akun saat sedang dihapus.

Ingat bahwa pengguna mungkin memiliki file di luar folder home mereka. Perintah `find` dapat menemukannya:

```
temukan / nama pengguna pengguna
```

Namun, perhatikan bahwa perintah di atas akan memakan waktu lama, jika Anda memiliki disk yang besar. Jika Anda memasang disk jaringan, Anda harus berhati-hati agar tidak membuang jaringan atau server.

Beberapa distribusi Linux datang dengan perintah khusus untuk melakukan ini; cari deluser atau userdel. Namun, mudah untuk melakukannya dengan tangan juga, dan perintah mungkin tidak melakukan semuanya.

11.5. Menonaktifkan pengguna untuk sementara

Terkadang perlu menonaktifkan akun untuk sementara, tanpa menghapusnya. Misalnya, pengguna mungkin belum membayar biayanya, atau administrator sistem mungkin menduga bahwa seorang cracker telah mendapatkan kata sandi akun tersebut.

Cara terbaik untuk menonaktifkan akun adalah dengan mengubah cangkangnya menjadi program khusus yang hanya mencetak pesan. Dengan cara ini, siapa pun yang mencoba masuk ke akun, akan gagal, dan akan tahu alasannya. Pesan tersebut dapat memberitahu pengguna untuk menghubungi administrator sistem sehingga masalah dapat ditangani.

Dimungkinkan juga untuk mengubah nama pengguna atau kata sandi menjadi sesuatu yang lain, tetapi kemudian pengguna tidak akan tahu apa yang sedang terjadi. Pengguna yang bingung berarti lebih banyak pekerjaan.

Cara sederhana untuk membuat program khusus adalah dengan menulis `skrip ekor`:

```
#!/usr/bin/ekor +2
Akun ini telah ditutup karena pelanggaran keamanan.
Silakan hubungi 555-1234 dan tunggu pria berbaju hitam tiba.
```

Dua karakter pertama (`#!`) beri tahu kernel bahwa sisa baris adalah perintah yang perlu dijalankan untuk menginterpretasikan file ini. Perintah tail dalam hal ini menampilkan semuanya kecuali baris pertama ke output standar.

Jika tagihan pengguna dicurigai melakukan pelanggaran keamanan, administrator sistem akan melakukan sesuatu seperti ini:

```
# chsh s
/usr/local/lib/no-login/tagihan keamanan
# su penguji
Akun ini telah ditutup karena pelanggaran keamanan.
Silakan hubungi 555-1234 dan tunggu pria berbaju hitam tiba.
#
```

Tujuan dari su adalah untuk menguji apakah perubahan itu berhasil, tentu saja.

Skrip ekor harus disimpan dalam folder terpisah, sehingga namanya tidak mengganggu perintah pengguna biasa.

Bab 12. Cadangan

Perangkat keras tidak dapat diandalkan.
Perangkat lunak secara deterministik tidak dapat diandalkan. Orang-orang secara indeterministik tidak dapat diandalkan.
Alam secara deterministik dapat diandalkan.

Bab ini menjelaskan tentang mengapa, bagaimana, dan kapan harus membuat cadangan, dan bagaimana ~~memulihkan sesuatu dari cadangan.~~

12.1. Tentang pentingnya di-backup

Data Anda berharga. Ini akan menghabiskan waktu dan usaha Anda untuk membuatnya kembali, dan itu membutuhkan uang atau setidaknya kesedihan dan air mata pribadi; kadang-kadang bahkan tidak dapat dibuat ulang, misalnya, jika itu adalah hasil dari beberapa eksperimen. Karena ini adalah investasi, Anda harus melindunginya dan mengambil langkah-langkah untuk menghindari kehilangannya.

Pada dasarnya ada empat alasan mengapa Anda mungkin kehilangan data: kegagalan perangkat keras, bug perangkat lunak, tindakan manusia, atau bencana alam. Meskipun perangkat keras modern cenderung cukup andal, perangkat ini masih dapat rusak secara spontan. Bagian paling penting dari perangkat keras untuk menyimpan data adalah hard disk, yang bergantung pada medan magnet kecil yang tetap utuh di dunia yang penuh dengan kebisingan elektromagnetik. Perangkat lunak modern bahkan cenderung tidak dapat diandalkan; program rock solid adalah pengecualian, bukan aturan. Manusia sangat tidak dapat diandalkan, mereka akan membuat kesalahan, atau mereka akan jahat dan menghancurkan data dengan sengaja. Alam mungkin tidak jahat, tetapi bisa mendatangkan malapetaka bahkan ketika sedang baik. Secara keseluruhan, ini adalah keajaiban kecil bahwa semuanya berhasil.

Cadangan adalah cara untuk melindungi investasi dalam data. Dengan memiliki beberapa salinan data, tidak masalah jika satu dimusnahkan (biayanya hanya untuk memulihkan data yang hilang dari cadangan).

Penting untuk melakukan backup dengan benar. Seperti semua hal lain yang terkait dengan dunia fisik, cepat atau lambat pencadangan akan gagal. Bagian dari melakukan pencadangan dengan baik adalah memastikannya berfungsi; Anda tidak ingin melihat bahwa cadangan Anda tidak berfungsi. Menambahkan penghinaan pada cedera, Anda mungkin mengalami crash yang buruk saat Anda membuat cadangan; jika Anda hanya memiliki satu media cadangan, itu mungkin akan hancur juga, meninggalkan Anda dengan abu merokok dari kerja keras. Atau Anda mungkin memperhatikan, ketika mencoba memulihkan, bahwa Anda lupa mencadangkan sesuatu yang penting, seperti basis data pengguna di situs pengguna 15000. Yang terbaik dari semuanya, semua cadangan Anda mungkin bekerja dengan sempurna, tetapi tape drive terakhir yang diketahui membaca jenis kaset yang Anda gunakan adalah yang sekarang memiliki seember air di dalamnya.

Ketika datang ke backup, paranoia ada dalam deskripsi pekerjaan.

12.2. Memilih media cadangan

Keputusan paling penting mengenai pencadangan adalah pilihan media pencadangan. Anda perlu mempertimbangkan biaya, keandalan, kecepatan, ketersediaan, dan kegunaan.

Biaya itu penting, karena Anda sebaiknya memiliki penyimpanan cadangan beberapa kali lebih banyak daripada yang Anda perlukan untuk data. Sebuah media murah biasanya suatu keharusan.

Keandalan sangat penting, karena cadangan yang rusak dapat membuat pria dewasa menangis. Media cadangan harus mampu menyimpan data tanpa kerusakan selama bertahun-tahun. Cara Anda menggunakan media memengaruhi keandalannya sebagai media cadangan. Hard disk biasanya sangat andal, tetapi sebagai media cadangan, hard disk tidak terlalu andal, jika di komputer yang sama dengan disk yang Anda buat cadangannya.

Kecepatan biasanya tidak terlalu penting, jika pencadangan dapat dilakukan tanpa interaksi. Tidak masalah jika backup memakan waktu dua jam, selama tidak membutuhkan pengawasan. Di sisi lain, jika pencadangan tidak dapat dilakukan saat komputer tidak digunakan, maka kecepatan menjadi masalah.

Ketersediaan jelas diperlukan, karena Anda tidak dapat menggunakan media cadangan jika tidak ada. Yang kurang jelas adalah kebutuhan akan media yang akan tersedia bahkan di masa depan, dan pada komputer selain milik Anda. Jika tidak, Anda mungkin tidak dapat memulihkan cadangan setelah bencana.

Kegunaan adalah faktor besar dalam seberapa sering pencadangan dilakukan. Semakin mudah membuat cadangan, semakin baik. SEBUAH media cadangan tidak boleh sulit atau membosankan untuk digunakan.

Alternatif tipikal adalah disket dan kaset. Floppy sangat murah, cukup andal, tidak terlalu cepat, sangat tersedia, tetapi tidak terlalu berguna untuk data dalam jumlah besar. Kaset murah sampai agak mahal, cukup andal, cukup cepat, cukup tersedia, dan, tergantung pada ukuran pita, cukup nyaman.

Ada alternatif lain. Mereka biasanya tidak terlalu bagus dalam hal ketersediaan, tetapi jika itu tidak menjadi masalah, mereka bisa lebih baik dengan cara lain. Sebagai contoh, disk magneto-optical dapat memiliki sisi baik dari kedua disket (mereka akses acak, membuat pemulihan satu file cepat) dan kaset (berisi banyak data).

12.3. Memilih alat pencadangan

Ada banyak tools yang bisa digunakan untuk melakukan backup. Alat UNIX tradisional yang digunakan untuk pencadangan adalah tar, cpio, dan dump. Selain itu, ada banyak paket pihak ketiga (baik freeware maupun komersial) yang dapat digunakan. Pilihan media cadangan dapat mempengaruhi pilihan alat.

tar dan **cpio** serupa, dan sebagian besar setara dari sudut pandang cadangan. Keduanya mampu menyimpan file pada kaset, dan mengambil file dari mereka. Keduanya mampu menggunakan hampir semua media, karena driver perangkat kernel menangani penanganan perangkat tingkat rendah dan semua perangkat cenderung mirip dengan program tingkat pengguna. Beberapa versi UNIX dari tar dan cpio mungkin memiliki masalah dengan file yang tidak biasa (tautan simbolik, file perangkat, file dengan nama path yang sangat panjang, dan sebagainya), tetapi versi Linux harus menangani semua file dengan benar.

membuang berbeda karena membaca sistem file secara langsung dan tidak melalui sistem file. Itu juga ditulis khusus untuk cadangan; tar dan cpio benar-benar untuk file pengarsipan, meskipun mereka bekerja untuk backup juga.

Membaca sistem file secara langsung memiliki beberapa keuntungan. Itu memungkinkan untuk membuat cadangan file tanpa mempengaruhi stempel waktu mereka; untuk tar dan cpio, Anda harus me-mount sistem file read-only terlebih dahulu. Membaca sistem file secara langsung juga lebih efektif, jika semuanya perlu dicadangkan, karena dapat dilakukan dengan lebih sedikit gerakan kepala disk. Kerugian utama adalah

membuat program cadangan khusus untuk satu jenis sistem file; program dump Linux hanya memahami sistem file ext2.

membuang juga secara langsung mendukung level cadangan (yang akan kita bahas di bawah); dengan tar dan cpio ini harus diimplementasikan dengan alat lain.

Perbandingan alat pencadangan pihak ketiga berada di luar cakupan buku ini. Peta Perangkat Lunak Linux mencantumkan banyak yang gratis.

12.4. Cadangan sederhana

Skema pencadangan sederhana adalah mencadangkan semuanya sekali, lalu mencadangkan semua yang telah dimodifikasi sejak pencadangan sebelumnya. Cadangan pertama disebut cadangan penuh, yang berikutnya adalah cadangan tambahan. Sebuah full backup seringkali lebih sulit daripada yang inkremental, karena ada lebih banyak data untuk ditulis ke tape dan full backup mungkin tidak muat ke dalam satu tape (atau floppy). Memulihkan dari cadangan tambahan bisa berkali-kali lebih banyak pekerjaan daripada dari yang lengkap. Restorasi dapat dioptimalkan sehingga Anda selalu mencadangkan semuanya sejak pencadangan penuh sebelumnya; dengan cara ini, pencadangan sedikit lebih berfungsi, tetapi seharusnya tidak pernah ada kebutuhan untuk memulihkan lebih dari cadangan penuh dan cadangan tambahan.

Jika Anda ingin membuat cadangan setiap hari dan memiliki enam kaset, Anda dapat menggunakan pita 1 untuk cadangan penuh pertama (misalnya, pada hari Jumat), dan pita 2 hingga 5 untuk cadangan tambahan (Senin sampai Kamis). Kemudian Anda membuat full backup baru pada tape 6 (Jumat kedua), dan mulai melakukan inkremental dengan tape 2 sampai 5 lagi. kamu tidak ingin menimpa pita 1 sampai Anda memiliki cadangan lengkap yang baru, jangan sampai terjadi sesuatu saat Anda membuat cadangan penuh. Setelah Anda membuat cadangan penuh ke pita 6, Anda ingin menyimpan pita 1 di tempat lain, sehingga ketika pita cadangan Anda yang lain hancur dalam kebakaran, Anda masih memiliki setidaknya sesuatu yang tersisa. Saat Anda perlu membuat cadangan lengkap berikutnya, Anda mengambil pita 1 dan meninggalkan pita 6 di tempatnya.

Jika Anda memiliki lebih dari enam kaset, Anda dapat menggunakan yang ekstra untuk cadangan penuh. Setiap kali Anda membuat cadangan penuh, Anda menggunakan kaset tertua. Dengan cara ini Anda dapat memiliki cadangan penuh dari beberapa minggu sebelumnya, yang bagus jika Anda ingin menemukan file lama yang sudah dihapus, atau file versi lama.

12.4.1. Membuat cadangan dengan tar

Cadangan lengkap dapat dengan mudah dibuat dengan tar:

```
# tar buat file /dev/ftape
/usr/src
tar: Menghapus awalan / dari nama jalur absolut di
arsip
#
```

Contoh di atas menggunakan tar versi GNU dan nama opsinya yang panjang. Versi tar tradisional hanya memahami opsi karakter tunggal. Versi GNU juga dapat menangani backup yang tidak muat pada satu tape atau floppy, dan juga path yang sangat panjang; tidak semua versi tradisional dapat melakukan hal ini. (Linux hanya menggunakan tar GNU.)

Jika cadangan Anda tidak muat pada satu kaset, Anda perlu menggunakan `multivolume (M)` pilihan:

```
# tar cMf /dev/fd0H1440
/usr/src
```

```
tar: Menghapus awalan / dari nama jalur absolut di
arsip
Siapkan volume #2 untuk /dev/fd0H1440 dan tekan kembali:
#
```

Perhatikan bahwa Anda harus memformat disket sebelum Anda memulai pencadangan, atau gunakan jendela lain atau terminal virtual dan lakukan ketika tar meminta disket baru.

Setelah Anda membuat cadangan, Anda harus memeriksa apakah tidak apa-apa, menggunakan `bandingkan` (d) pilihan:

```
# tar bandingkan verbose f
/dev/ftape
usr/src/usr/s
rc/linux
```

```
usr/src/linux-1.2.10-termasuk/  
....  
#
```

Gagal memeriksa cadangan berarti Anda tidak akan melihat bahwa cadangan Anda tidak berfungsi sampai Anda kehilangan data asli.

Pencadangan tambahan dapat dilakukan dengan tar menggunakan lebih baru (T) pilihan:

```
# tar buat baru '8 Sep 1995'  
file /dev/ftape /usr/src  
verbose  
tar: Menghapus awalan / dari nama jalur absolut di  
arsip  
usr/src/                usr/src/linux-1.2.10-includes/  
usr/src/linux-1.2.10-includes/include/  
usr/src/linux-1.2.10-includes/include/linux/  
usr/src/  
linux-1.2.10-includes/include/linux/modules/  
usr/src/linux-1.2.10-includes/include/asm-generic/  
usr/src/linux-1.2.10-includes/include/asm-i386/  
usr/src/linux-1.2.10-includes/include/asm-mips/  
usr/src/linux-1.2.10-includes/include/asm-alpha/  
usr/src/linux-1.2.10-includes/include/asm-m68k/  
usr/src/linux-1.2.10-includes/include/asm-sparc/  
usr/src/patch-1.2.11.gz  
#
```

Sayangnya, tar tidak dapat melihat ketika informasi inode file telah berubah, misalnya, bahwa bit izinnya telah diubah, atau ketika namanya telah diubah. Ini dapat diatasi dengan menggunakan find dan membandingkan status sistem file saat ini dengan daftar file yang telah dicadangkan sebelumnya. Skrip dan program untuk melakukan ini dapat ditemukan di situs ftp Linux.

12.4.2. Memulihkan file dengan tar

Itu ekstrak(x) opsi untuk mengekstrak file tar:

```
# tar ekstrak sama-izin  
verbose file  
/dev/fd0H1440  
usr/src/usr/s  
rc/linux  
usr/src/linux-1.2.10-includes/  
usr/src/linux-1.2.10-includes/include/  
usr/src/linux-1.2.10-includes/include/linux/  
usr/src/linux-1.2.10-includes/include/linux/hdreg.h  
usr/src/linux-1.2.10-includes/include/linux/kernel.h  
...  
#
```

Anda juga hanya mengekstrak file atau folder tertentu (yang mencakup semua file dan subfoldernya) dengan memberi nama pada baris perintah:

```
# tar xpvf /dev/fd0H1440  
usr/src/linux-1.2.10-includes/include/linux/hdreg.h  
usr/src/linux-1.2.10-includes/include/linux/hdreg.h  
#
```

Menggunakan daftar (t) opsi, jika Anda hanya ingin melihat file apa yang ada di volume cadangan:

```
# tar daftar file  
/dev/fd0H1440  
usr/src/
```

```
usr/src/linux          usr/src/linux-1.2.10-includes/  
usr/src/linux-1.2.10-includes/include/  
usr/src/linux-1.2.10-includes/include/linux/  usr/src  
/linux-1.2.10-includes/include/linux/hdreg.h  
usr/src/linux-1.2.10-includes/include/linux/kernel.h  
...  
#
```

~~Perhatikan bahwa tar selalu membaca volume cadangan secara berurutan, jadi untuk volume besar agak lambat. Namun, tidak mungkin menggunakan teknik database akses acak saat menggunakan tape drive atau media sekuensial lainnya.~~

ter tidak menangani file yang dihapus dengan benar. Jika Anda perlu memulihkan sistem file dari cadangan penuh dan cadangan tambahan, dan Anda telah menghapus file di antara kedua cadangan, file itu akan ada lagi setelah Anda melakukan pemulihan. Ini bisa menjadi masalah besar, jika file tersebut memiliki data sensitif yang seharusnya tidak lagi tersedia.

12.5. Pencadangan bertingkat

Metode pencadangan sederhana yang diuraikan di bagian sebelumnya seringkali cukup memadai untuk penggunaan pribadi atau situs kecil. Untuk penggunaan tugas yang lebih berat, pencadangan bertingkat lebih sesuai.

Metode sederhana memiliki dua tingkat pencadangan: pencadangan penuh dan cadangan tambahan. Ini dapat digeneralisasikan ke sejumlah level. Cadangan penuh akan menjadi level 0, dan berbagai level cadangan inkremental level 1, 2, 3, dll. Pada setiap tingkat pencadangan tambahan, Anda mencadangkan semua yang telah berubah sejak pencadangan sebelumnya pada tingkat yang sama atau sebelumnya.

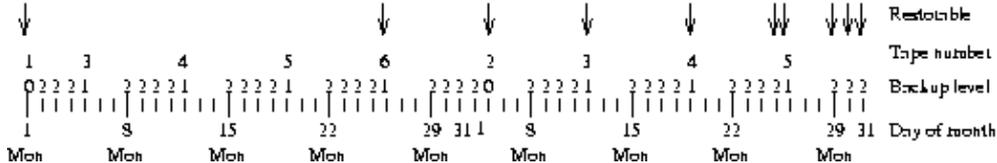
Tujuan untuk melakukan ini adalah memungkinkan riwayat pencadangan yang lebih lama dengan murah. Pada contoh di bagian sebelumnya, riwayat pencadangan kembali ke pencadangan lengkap sebelumnya. Ini dapat diperpanjang dengan memiliki lebih banyak kaset, tetapi hanya seminggu untuk setiap kaset baru, yang mungkin terlalu mahal. Riwayat pencadangan yang lebih panjang berguna, karena file yang terhapus atau rusak sering kali tidak diperhatikan untuk waktu yang lama. Bahkan versi file yang tidak terlalu mutakhir lebih baik daripada tidak ada file sama sekali.

Dengan beberapa level, riwayat pencadangan dapat diperpanjang dengan lebih murah. Misalnya, jika kita membeli sepuluh kaset, kita bisa menggunakan kaset 1 dan 2 untuk backup bulanan (Jumat pertama setiap bulan), kaset 3 sampai 6 untuk backup mingguan (Jumat lainnya; perhatikan bahwa bisa ada lima hari Jumat dalam satu bulan, jadi kita membutuhkan empat kaset lagi), dan kaset 7 sampai 10 untuk backup harian (Senin sampai Kamis). Dengan hanya empat kaset lagi, kami dapat memperpanjang cadangan sejarah dari dua minggu (setelah semua kaset harian telah digunakan) sampai dua bulan. Memang benar bahwa kami tidak dapat memulihkan setiap versi dari setiap file selama dua bulan itu, tetapi apa yang dapat kami pulihkan seringkali cukup baik.

Gambar 12-1 menunjukkan level cadangan mana yang digunakan setiap hari, dan cadangan mana yang dapat dipulihkan mulai akhir bulan.

Gambar 12–1. Contoh jadwal pencadangan bertingkat.

Panduan Administrator Sistem Linux



Tingkat cadangan juga dapat digunakan untuk menjaga waktu pemulihan sistem file seminimal mungkin. Jika Anda memiliki banyak cadangan inkremental dengan jumlah level yang tumbuh secara monoton, Anda perlu memulihkan semuanya jika Anda perlu membangun kembali seluruh sistem file. Sebagai gantinya, Anda dapat menggunakan nomor level yang tidak monoton, dan mengurangi jumlah cadangan untuk dipulihkan.

Untuk meminimalkan jumlah kaset yang diperlukan untuk memulihkan, Anda dapat menggunakan tingkat yang lebih kecil untuk setiap kaset tambahan. Namun, kemudian waktu untuk membuat cadangan meningkat (setiap cadangan menyalin semuanya sejak cadangan lengkap sebelumnya). Skema yang lebih baik disarankan oleh halaman manual dump dan dijelaskan oleh tabel XX (tingkat-cadangan yang efisien). Gunakan urutan level pencadangan berikut: 3, 2, 5, 4, 7, 6, 9, 8, 9, dll. Ini membuat waktu pencadangan dan pemulihan tetap rendah. Yang paling Anda harus backup adalah senilai dua hari kerja. Jumlah kaset untuk pemulihan tergantung pada berapa lama Anda menyimpan antara cadangan penuh, tetapi kurang dari skema sederhana.

Tabel 12–1. Skema pencadangan yang efisien menggunakan banyak level pencadangan

Tape	Tingkat	Cadangan (hari)	Kembalikan
1	0	tidak ada	1
2	3	1	1, 2
3	2	2	1, 3
4	5	1	1, 2, 4
5	4	2	1, 2, 5
6	7	1	1, 2, 5, 6
7	6	2	1, 2, 5, 7
8	9	1	1, 2, 5, 7, 8
9	8	2	1, 2, 5, 7, 9
10	9	1	1, 2, 5, 7, 9, 10
11	9	1	1, 2, 5, 7, 9, 10, 11
...	9	1	1, 2, 5, 7, 9, 10, 11, ...

Skema mewah dapat mengurangi jumlah tenaga kerja yang dibutuhkan, tetapi itu berarti ada lebih banyak hal yang harus diperhatikan. Anda harus memutuskan apakah itu layak.

membuang memiliki dukungan bawaan untuk level pencadangan. Untuk tar dan cpio harus diimplementasikan dengan skrip shell.

12.6. Apa yang harus dicadangkan?

Anda ingin mencadangkan sebanyak mungkin. Pengecualian utama adalah perangkat lunak yang dapat dengan mudah diinstal ulang, tetapi bahkan mereka mungkin memiliki file konfigurasi yang penting untuk dicadangkan, agar Anda tidak perlu melakukan semua pekerjaan untuk mengonfigurasinya lagi. Pengecualian

Panduan Administrator Sistem Linux

utama lainnya adalah `/proc` berkas sistem; karena itu hanya berisi data yang selalu dihasilkan oleh kernel secara otomatis, tidak pernah merupakan ide yang baik untuk mencadangkannya. Terutama `/proc/kcore` file tidak diperlukan, karena itu hanya gambar dari memori fisik Anda saat ini; itu cukup besar juga.

Area abu-abu termasuk gulungan berita, file log, dan banyak hal lainnya di `/var`. Anda harus memutuskan apa yang Anda anggap penting.

Hal yang jelas untuk dicadangkan adalah file pengguna (`/home`) dan file konfigurasi sistem (`/etc`, tetapi mungkin hal-hal lain tersebar di seluruh sistem file).

12.7. Cadangan terkompresi

Pencadangan membutuhkan banyak ruang, yang dapat menghabiskan banyak uang. Untuk mengurangi ruang yang dibutuhkan, cadangan dapat dikompresi. Ada beberapa cara untuk melakukan ini. Beberapa program memiliki dukungan untuk kompresi bawaan; misalnya, `gzip` (`z`) opsi untuk GNU tar menyalurkan seluruh cadangan melalui program kompresi `gzip`, sebelum menuliskannya ke media cadangan.

Sayangnya, cadangan terkompresi dapat menyebabkan masalah. Karena sifat cara kerja kompresi, jika satu bit salah, semua data terkompresi lainnya tidak akan dapat digunakan. Beberapa program pencadangan memiliki beberapa koreksi kesalahan bawaan, tetapi tidak ada metode yang dapat menangani sejumlah besar kesalahan. Ini berarti bahwa jika cadangan dikompresi dengan cara GNU tar melakukannya, dengan seluruh output dikompresi sebagai satu unit, satu kesalahan membuat semua sisa cadangan hilang. Cadangan harus dapat diandalkan, dan metode kompresi ini bukanlah ide yang baik.

Cara alternatif adalah dengan mengompres setiap file secara terpisah. Ini masih berarti bahwa satu file hilang, tetapi semua file lainnya tidak terluka. File yang hilang akan tetap rusak, jadi situasi ini tidak jauh lebih buruk daripada tidak menggunakan kompresi sama sekali. Program `afio` (varian dari `cpio`) dapat melakukan ini.

Kompresi membutuhkan waktu, yang mungkin membuat program pencadangan tidak dapat menulis data dengan cukup cepat untuk tape drive. Hal ini dapat dihindari dengan buffering output (baik secara internal, jika program cadangan jika cukup pintar, atau dengan menggunakan program lain), tetapi bahkan itu mungkin tidak bekerja dengan cukup baik. Ini seharusnya hanya menjadi masalah pada komputer yang lambat.

Bab 13. Otomatisasi Tugas Untuk

"Jangan pernah menunda sampai besok apa yang bisa kamu lakukan lusa. 'Mark Twain'"

Diskusi dasar tentang scripting, cron & di lihat HOWTO lain untuk detailnya. Diskusikan pekerjaan cron non-crontab seperti yang ada di folder /etc.

"Waktu adalah ilusi. Waktu makan siang jadi dua kali lipat." (Douglas Adams.)

Bab ini menjelaskan bagaimana sistem Linux menjaga waktu, dan apa yang perlu Anda lakukan untuk menghindari masalah. Biasanya, Anda tidak perlu melakukan apa pun tentang waktu, tetapi ada baiknya untuk memahaminya.

14.1. Konsep waktu setempat

Pengukuran waktu sebagian besar didasarkan pada fenomena alam biasa, seperti periode terang dan gelap yang bergantian yang disebabkan oleh rotasi planet. Total waktu yang dibutuhkan oleh dua periode berturut-turut adalah konstan, tetapi panjang periode terang dan gelap bervariasi. Satu konstanta sederhana adalah tengah hari.

Siang hari adalah waktu dimana Matahari berada pada posisi tertingginya. Karena (menurut penelitian terbaru) Bumi itu bulat, siang terjadi pada waktu yang berbeda di tempat yang berbeda. Ini mengarah pada konsep waktu lokal. Manusia mengukur waktu dalam banyak unit, yang sebagian besar terkait dengan fenomena alam seperti siang hari. Selama Anda tinggal di tempat yang sama, tidak masalah perbedaan waktu setempat.

Segera setelah Anda perlu berkomunikasi dengan tempat-tempat yang jauh, Anda akan menyadari perlunya waktu yang sama. Di zaman modern, sebagian besar tempat di dunia berkomunikasi dengan sebagian besar tempat lain di dunia, sehingga standar global untuk mengukur waktu telah ditetapkan. Waktu ini disebut waktu universal (UT atau UTC, sebelumnya dikenal sebagai Greenwich Mean Time atau GMT, karena dulunya waktu lokal di Greenwich, Inggris). Ketika orang-orang dengan waktu lokal yang berbeda perlu berkomunikasi, mereka dapat mengekspresikan waktu dalam waktu universal, sehingga tidak ada kebingungan tentang kapan sesuatu harus terjadi.

Setiap waktu setempat disebut zona waktu. Sementara geografi akan memungkinkan semua tempat yang memiliki siang pada waktu yang sama memiliki zona waktu yang sama, politik membuatnya sulit. Untuk berbagai alasan, banyak negara menggunakan waktu musim panas, yaitu, mereka memindahkan jam mereka untuk mendapatkan lebih banyak cahaya alami saat bekerja, dan kemudian memindahkan jam. kembali selama musim dingin. Negara lain tidak melakukan ini. Mereka yang melakukannya, tidak setuju kapan jam harus dipindahkan, dan mereka mengubah aturan dari tahun ke tahun. Hal ini membuat konversi zona waktu jelas tidak sepele.

Zona waktu paling baik dinamai berdasarkan lokasi atau dengan membedakan antara waktu lokal dan universal. Di AS dan beberapa negara lain, zona waktu lokal memiliki nama dan singkatan tiga huruf. Namun, singkatannya tidak unik, dan tidak boleh digunakan kecuali jika negaranya juga disebutkan. Lebih baik berbicara tentang waktu lokal di, katakanlah, Helsinki, daripada tentang waktu Eropa Timur, karena tidak semua negara di Eropa Timur mengikuti aturan yang sama.

Linux memiliki paket zona waktu yang mengetahui semua zona waktu yang ada, dan dapat dengan mudah diperbarui saat aturan berubah. Yang perlu dilakukan administrator sistem adalah memilih zona waktu yang sesuai. Selain itu, setiap pengguna dapat mengatur zona waktunya sendiri; ini penting karena banyak orang bekerja dengan komputer di berbagai negara melalui Internet. Ketika aturan untuk waktu musim panas berubah di zona waktu lokal Anda, pastikan Anda memutakhirkan setidaknya bagian itu dari sistem Linux Anda. Selain mengatur zona waktu sistem dan memutakhirkan file data zona waktu, Anda tidak perlu repot dengan waktu.

14.2. Jam perangkat keras dan perangkat lunak

Komputer pribadi memiliki jam perangkat keras yang digerakkan oleh baterai. Baterai memastikan bahwa jam akan tetap berfungsi meskipun bagian lain komputer tidak ada listrik. Jam perangkat keras dapat diatur dari layar pengaturan BIOS atau dari sistem operasi apa pun yang sedang berjalan.

Kernel Linux melacak waktu secara independen dari jam perangkat keras. Selama boot, Linux menyetel jamnya sendiri ke waktu yang sama dengan jam perangkat keras. Setelah ini, kedua jam berjalan secara independen. Linux mempertahankan jamnya sendiri karena melihat perangkat kerasnya lambat dan rumit.

Jam kernel selalu menunjukkan waktu universal. Dengan cara ini, kernel tidak perlu tahu tentang zona waktu sama sekali. Kesederhanaan menghasilkan keandalan yang lebih tinggi dan membuatnya lebih mudah untuk memperbarui informasi zona waktu. Setiap proses menangani konversi zona waktu itu sendiri (menggunakan alat standar yang merupakan bagian dari paket zona waktu).

Jam perangkat keras dapat dalam waktu lokal atau dalam waktu universal. Biasanya lebih baik memilikinya dalam waktu universal, karena Anda tidak perlu mengubah jam perangkat keras saat waktu musim panas dimulai atau berakhir (UTC tidak memiliki DST). Sayangnya, beberapa sistem operasi PC, termasuk MS-DOS, Windows, dan OS/2, menganggap jam perangkat keras menunjukkan waktu lokal. Linux dapat menangani keduanya, tetapi jika jam perangkat keras menunjukkan waktu lokal, maka jam tersebut harus dimodifikasi ketika waktu musim panas dimulai atau berakhir (jika tidak, waktu setempat tidak akan ~~ditampilkan~~).

14.3. Menampilkan dan mengatur waktu

Di Linux, zona waktu sistem ditentukan oleh tautan simbolis `/etc/waktu` setempat. Tautan ini menunjuk ke file data zona waktu yang menjelaskan zona waktu lokal. File data zona waktu terletak di `/usr/lib/zoneinfo` atau `/usr/share/zoneinfo` tergantung pada distribusi Linux apa yang Anda gunakan.

Misalnya, pada sistem SuSE yang berlokasi di New Jersey, `/etc/waktu` setempat tautan akan mengarah ke `/usr/share/zoneinfo/US/Eastern`. Pada sistem Debian, `/etc/waktu` setempat tautan akan menunjuk ke `/usr/lib/zoneinfo/US/Eastern`.

Jika Anda gagal menemukan `info` zona folder baik di `/usr/lib` atau `/usr/share` folder, lakukan **temukan /usr cetak | info zona grep** atau lihat dokumentasi distribusi Anda.

Apa yang terjadi ketika Anda memiliki pengguna yang berada di zona waktu yang berbeda? Seorang pengguna dapat mengubah zona waktu pribadinya dengan mengatur variabel lingkungan TZ. Jika tidak disetel, zona waktu sistem diasumsikan. Sintaks variabel TZ dijelaskan dalam `tzset` halaman manual.

Perintah tanggal menunjukkan tanggal dan waktu saat ini. Sebagai contoh:

```
$ tanggal
Minggu 14 Jul 21:53:41 EET DST 1996
$
```

Waktu itu adalah Minggu, 14 Juli 1996, sekitar pukul sepuluh sebelum sepuluh malam, di zona waktu yang disebut ```EET DST``` (yang mungkin merupakan Waktu Musim Panas Eropa Timur). tanggal juga dapat menunjukkan waktu universal:

```
$ tanggal u
Minggu 14 Juli 18:53:42 UTC 1996
$
```

tanggal juga digunakan untuk mengatur jam perangkat lunak kernel:

```
# tanggal 07142157
Minggu 14 Jul 21:57:00 EET DST 1996
# tanggal
Minggu 14 Jul 21:57:02 EET DST 1996
#
```

Lihat halaman manual **tanggal** untuk lebih jelasnya; sintaksnya agak misterius. Hanya root yang dapat mengatur waktu. Meskipun setiap pengguna dapat memiliki zona waktunya sendiri, jamnya sama untuk semua orang.

Waspadalah terhadap perintah waktu. Ini tidak digunakan untuk mendapatkan waktu sistem. Sebaliknya itu digunakan untuk waktu berapa lama sesuatu berlangsung. Lihat halaman manual waktu.

tanggal hanya menampilkan atau menyetel jam perangkat lunak. Perintah jam menyinkronkan jam perangkat keras dan perangkat lunak. Ini digunakan saat sistem melakukan booting, untuk membaca jam perangkat keras dan mengatur jam perangkat lunak. Jika Anda perlu menyetel kedua jam, pertama-tama Anda menyetel jam perangkat lunak dengan **tanggal**, lalu jam perangkat keras dengan jam w.

Itu `u` pilihan untuk jam memberitahu bahwa jam perangkat keras dalam waktu universal. Anda harus menggunakan `u` pilihan dengan benar. Jika tidak, komputer Anda akan sangat bingung tentang jam berapa sekarang.

Jam harus diubah dengan hati-hati. Banyak bagian dari sistem Unix membutuhkan jam untuk bekerja dengan benar. Misalnya, daemon cron menjalankan perintah secara berkala. Jika Anda mengubah jam, dapat bingung apakah perlu menjalankan perintah atau tidak. Pada satu sistem Unix awal, seseorang mengatur jam dua puluh tahun ke depan, dan cron ingin menjalankan semua perintah periodik selama dua puluh tahun sekaligus. Versi cron saat ini dapat menangani ini dengan benar, tetapi Anda tetap harus berhati-hati. Lompatan besar atau lompat mundur lebih berbahaya daripada lompat kecil atau maju.

14.4. Ketika jam salah

Jam perangkat lunak Linux tidak selalu akurat. Itu terus berjalan oleh interupsi pengatur waktu berkala yang dihasilkan oleh perangkat keras PC. Jika sistem memiliki terlalu banyak proses yang berjalan, mungkin diperlukan waktu terlalu lama untuk melayani interupsi pengatur waktu, dan jam perangkat lunak mulai tertinggal. Jam perangkat keras berjalan secara independen dan biasanya lebih akurat. Jika Anda sering mem-boot komputer Anda (seperti yang terjadi pada kebanyakan sistem yang bukan server), biasanya akan menjaga waktu yang cukup akurat.

Jika Anda perlu menyesuaikan jam perangkat keras, biasanya yang paling sederhana adalah mem-boot ulang, masuk ke layar pengaturan BIOS, dan lakukan dari sana. Ini menghindari semua masalah yang mungkin disebabkan oleh perubahan waktu sistem. Jika melakukannya melalui BIOS bukanlah pilihan, atur waktu baru dengan **tanggal** dan jam (dalam urutan itu), tetapi bersiaplah untuk reboot, jika beberapa bagian dari sistem mulai bertingkah aneh.

Metode lain adalah dengan menggunakan `hwclock w` atau `hwclock systohc` untuk menyinkronkan jam perangkat keras ke jam perangkat lunak. Jika Anda ingin menyinkronkan jam perangkat lunak Anda ke jam perangkat keras Anda, maka Anda akan menggunakan `hwclock satau jam hwtsys`. Untuk informasi lebih lanjut tentang perintah ini baca `man hwclock`.

14.5. NTP Protokol Waktu Jaringan

Panduan Administrator Sistem Linux

Komputer berjaringan (bahkan jika hanya melalui modem) dapat memeriksa jamnya sendiri secara otomatis dengan membandingkannya dengan waktu di komputer lain yang dikenal memiliki waktu yang akurat. Network Time Protocol (atau NTP) melakukan hal itu. Ini adalah metode untuk memverifikasi dan mengoreksi waktu komputer Anda dengan menyinkronkannya dengan sistem lain. Dengan NTP, waktu sistem Anda dapat dipertahankan dalam milidetik dari Waktu Universal Terkoordinasi. Mengunjungi <http://www.time.gov/about.html> untuk info lebih lanjut.

Untuk pengguna Linux yang lebih kasual, ini hanyalah kemewahan yang menyenangkan. Di rumah saya, semua jam kami diatur berdasarkan apa yang dikatakan sistem Linux saya tentang waktu. Untuk organisasi yang lebih besar, "kemewahan" ini bisa menjadi penting. Mampu mencari file log untuk acara berdasarkan waktu dapat membuat hidup jauh lebih mudah dan menghilangkan banyak "kerja menebak" dari debugging.

Contoh lain betapa pentingnya NTP adalah dengan SAN. Beberapa SAN mengharuskan NTP dikonfigurasi dan berjalan dengan benar untuk memungkinkan sinkronisasi yang tepat atas penggunaan sistem file, dan kontrol stempel waktu yang tepat. Beberapa SAN (dan beberapa aplikasi) dapat menjadi bingung ketika berhadapan dengan file yang memiliki stempel waktu yang akan datang.

Sebagian besar distribusi Linux datang dengan beberapa jenis paket NTP, baik paket .deb atau .rpm. Anda dapat menggunakannya untuk menginstal NTP, atau Anda dapat mengunduh file sumber dari <http://www.ntp.org/downloads.html> dan kompilasi sendiri. Bagaimanapun, konfigurasi dasarnya sama.

14.6. Konfigurasi NTP dasar

Program NTP dikonfigurasi menggunakan salah satu dari: `/etc/ntp.conf` atau `/etc/xntp.conf` tergantung pada distribusi Linux yang Anda miliki. Saya tidak akan membahas terlalu detail tentang cara mengkonfigurasi NTP. Sebagai gantinya saya hanya akan membahas dasar-dasarnya.

Contoh file `ntp.conf` dasar akan terlihat seperti:

```
# KONFIGURASI UMUM
server aaa.bbb.ccc.ddd
server 127.127.1.0
fudge 127.127.1.0 strata 10

# Berkas melayang.

driftfile /etc/ntp/drift
```

File `ntp.conf` paling dasar hanya akan mencantumkan 2 server, salah satu yang ingin disinkronkan, dan IP semu alamat untuk dirinya sendiri (dalam hal ini 127.127.1.0). IP semu digunakan jika terjadi masalah jaringan atau jika:

server NTP jarak jauh mati. NTP akan menyinkronkan dirinya sendiri hingga NTP dapat mulai menyinkronkan dengan server jarak jauh lagi. Disarankan agar Anda mencantumkan setidaknya 2 server jarak jauh yang dapat Anda sinkronkan. Satu akan bertindak sebagai server utama dan yang lainnya sebagai cadangan.

Anda juga harus mencantumkan lokasi untuk file drift. Seiring waktu NTP akan "mempelajari" tingkat kesalahan jam sistem dan secara otomatis menyesuaikannya.

Opsi batasi dapat digunakan untuk memberikan kontrol dan keamanan yang lebih baik atas apa yang dapat dilakukan NTP, dan siapa yang dapat memengaruhinya. Sebagai contoh:

```
# Melarang akses umum ke layanan ini.
batasi default abaikan
```

Panduan Administrator Sistem Linux

```
# Izinkan sistem di jaringan ini untuk menyinkronkan dengan ini
# layanan waktu. Tapi tidak mengubah
waktu kita. batasi aaa.bbb.ccc.ddd
nomodify
```

```
# Izinkan akses tak terbatas berikut ke ntpd
```

```
membatasi aaa.bbb.ccc.ddd
```

```
batasi 127.0.0.1
```

Disarankan agar Anda menunggu hingga NTP berfungsi dengan baik sebelum menambahkan opsi batasi. Anda dapat secara tidak sengaja membatasi diri Anda dari sinkronisasi dan membuang waktu untuk men-debug alasannya.

NTP perlahan mengoreksi waktu sistem Anda. Sabar! Tes sederhana adalah mengubah jam sistem Anda 10 menit sebelum Anda pergi tidur dan kemudian memeriksanya saat Anda bangun. Waktunya harus tepat.

Banyak orang mendapatkan ide bahwa alih-alih menjalankan daemon NTP, mereka hanya perlu menyiapkan pekerjaan cron job untuk menjalankan perintah ntpdate secara berkala. Ada 2 kelemahan utama menggunakan metode ini.

Yang pertama adalah bahwa ntpdate melakukan metode "brute force" untuk mengubah waktu. Jadi jika waktu komputer Anda mati 5 menit saya, itu segera memperbaikinya. Di beberapa lingkungan, ini dapat menyebabkan masalah jika waktu berubah secara drastis. Misalnya, jika Anda menggunakan perangkat lunak keamanan yang sensitif terhadap waktu, Anda dapat secara tidak sengaja mematikan akses seseorang. Daemon NTP perlahan mengubah waktu untuk menghindari gangguan semacam ini.

Alasan lainnya adalah bahwa daemon NTP dapat dikonfigurasi untuk mencoba mempelajari penyimpangan waktu sistem Anda dan kemudian secara otomatis menyesuaikan.

14.7. Perangkat NTP

Ada sejumlah utilitas yang tersedia untuk memeriksa apakah NTP melakukan tugasnya. Perintah ntpq p akan mencetak status waktu sistem Anda saat ini.

```
#ntpq p
  refid st t jarak jauh saat polling mencapai delay offset jitter
-----
*cudns.cit.corne ntp0.usno.navy. 2 u 832 1024 377      43.208    0.361    2.646
LOKAL(0) LOKAL(0) 10 1 13 64 377          0,000    0,000    0,008
```

ntpdc c loopinfo akan menampilkan seberapa jauh waktu sistem dalam hitungan detik, berdasarkan waktu terakhir server jauh dihubungi.

```
# ntpdc c loopinfo
offset: 0.004479 s frekuensi:
133.625 ppm poll menyesuaikan: 30
pengatur waktu pengawas: 404 s
```

ntpdc c kerninfo akan menampilkan koreksi yang tersisa saat ini.

```
#ntpdc c kerninfo
pll offset: 0,003917 s frekuensi
pll: 133,625 ppm kesalahan
maksimum: 0,391414 d perkiraan
```

Panduan Administrator Sistem Linux

```
kesalahan: 0,003676 d status:
0001 pll pll konstanta waktu: 6
presisi: 1e-06 s toleransi
frekuensi: 512 ppm pps
frekuensi: 0,000 ppm
stabilitas pps: 512.000 ppm
jitter pps: interval kalibrasi
0,0002 detik: siklus kalibrasi 4
detik: 0
jitter terlampaui: 0
stabilitas terlampaui:
0 kesalahan kalibrasi:
0
```

Versi ntpdc yang sedikit lebih berbeda c kerninfo adalah ntptime

```
# ntptime
ntp_gettime() mengembalikan kode 0 (OK)
waktu c35e2cc7.879ba000 Kam, 13 Nov 2003 11:16:07.529, (.529718),
kesalahan maksimum 425206 kami, kesalahan perkiraan 3676 kami
ntp_adjtime() mengembalikan kode 0 (OK)
mode 0x0 (),
offset 3854.000 us, frekuensi 133.625 ppm, interval 4 s,
error maksimum 425206 us, estimasi error 3676 us,
status 0x1 (PLL),
konstanta waktu 6, presisi 1.000 us, toleransi 512 ppm,
frekuensi pps 0,000 ppm, stabilitas 512.000 ppm, jitter 200.000
us, interval 0, jitter melebihi 0, stabilitas melebihi 0, error 0.
```

Cara lain untuk melihat seberapa baik NTP bekerja adalah dengan perintah ntpdate d. Ini akan menghubungi

server dan menentukan perbedaan waktu tetapi tidak mengubah waktu sistem Anda.

```
# ntpdate d 132.236.56.250
13 Nov 14:43:17 ntpdate[29631]: ntpdate 4.1.1c- rc1@1.836 Kam 13 Feb 12:17:20 EST 2003 (1)
mengirimkan(132.236.56.2
50)
menerima(132.236.56.250)
mengirimkan(132.236.56.2
50)
menerima(132.236.56.250)
mengirimkan(132.236.56.2
50)
menerima(132.236.56.250)
mengirimkan(132.236.56.2
50)
menerima(132.236.56.250)
mengirimkan(
132.236.56.250)
server 132.236.56.250, port 123
stratum 2, presisi 17, lompatan 00, kepercayaan 000
refid [192.5.41.209], delay 0,06372, dispersi 0,00044
ditransmisikan 4, di filter 4
waktu referensi: c35e5998.4a46cfc8 Kam, 13 Nov 2003 14:27:20.290
stempel waktu awal: c35e5d55.d69a6f82 Kam, 13 Nov 2003 14:43:17.838
stempel waktu pengiriman: c35e5d55.d16fc9bc Kam, 13 Nov 2003
14:43:17.818 penundaan filter : 0,06522 0,06372 0,06442 0,06442
0,00000 0,00000 0,00000 0,00000
filter offset: 0,000036 0,001020 0,000527 0,000684
0,000000 0,000000 0,000000 0,000000
penundaan 0,06372, dispersi 0,00044
mengimbangi 0,001020
```

Panduan Administrator Sistem Linux

```
13 Nov 14:43:17 ntpdate[29631]: menyesuaikan waktu server 132.236.56.250 offset 0.001020 detik
```

Jika Anda ingin benar-benar menonton sinkronisasi sistem, Anda dapat menggunakan `ntptrace`.

```
#ntptrace 132.236.56.250
cudns.cit.cornell.edu: stratum 2, offset 0.003278, jarak sinkronisasi 0.02779
truetime.ntp.com: stratum 1, offset -0.014363, jarak sinkronisasi 0.00000, refid
'ACTS'
```

Jika Anda ingin waktu sistem Anda segera disinkronkan, Anda dapat menggunakan `ntpdate remote-servername` untuk memaksa sinkronisasi. Tidak menunggu!

```
#ntpdate 132.236.56.250
13 Nov 14:56:28 ntpdate[29676]: menyesuaikan waktu server 132.236.56.250 offset 0.003151 detik
```

14.8. Beberapa server NTP yang dikenal

Daftar server NTP publik dapat diperoleh dari: <http://www.eecis.udel.edu/~mills/ntp/servers.html>. Harap baca informasi penggunaan pada halaman sebelumnya sehingga menggunakan server. Tidak semua server memiliki bandwidth yang tersedia untuk memungkinkan sejumlah besar sistem melakukan sinkronisasi terhadap mereka. Oleh karena itu, sebaiknya hubungi administrator sistem sebelum menggunakan servernya untuk layanan NTP.

14.9. Link NTP

Informasi lebih rinci tentang NTP dapat diperoleh dari beranda NTP: <http://www.ntp.org>. Atau

dari <http://www.ntp.org/ntpfaq/NTP-a-faq.htm>
