

Studi Kasus:

GNU/Linux

Perangkat Lunak Bebas

Konsep bebas

- ❖ Analogi “kebebasan berbicara” vs “bir gratis”, konsep bebas lebih didasarkan pada kebebasan mengembangkan, mendistribusikan daripada sekedar “gratis”

Public Domain

- ❖ Perangkat lunak tanpa hak cipta pada badan atau pihak tertentu

OpenSource

- ❖ Bebas untuk menggunakan, menyalin, mendistribusikan, modifikasi, menjual, dan seterusnya
- ❖ Source kode mesti tersedia

Copyleft

- ❖ Tidak diperkenankan mengubah aturan awal pendistribusiannya sebagai opensource

GPL

- ❖ Kumpulan aturan untuk membuat sebuah distribusi jadi copyleft

GNU

- ❖ Didirikan untuk menghasilkan distribusi yang mirip Unix tapi sepenuhnya bebas

Software Semi Bebas

- ❖ Bebas digunakan, disalin, distribusi dan modifikasi
- ❖ Hanya dapat digunakan untuk tujuan non profit

Software berpemilik

- ❖ Ada keterbatasan dalam memodifikasi, mendistribusi, ataupun menggunakan

Freeware

- ❖ Bukan opensource
- ❖ Bebas untuk digunakan, didistribusikan tapi tidak untuk dimodifikasi

Shareware

- ❖ Bukan opensource
- ❖ Bebas didistribusikan, tetapi pemakai harus bayar *fee* (biaya)
- ❖ *Source code* tidak tersedia

Software Komersial

❖ Untuk profit

Sejarah dan Rancangan Dasar GNU/Linux

Sejarah

- ❖ Merupakan sistem operasi mirip UNIX
- ❖ Pada awalnya diciptakan oleh Linus Benedict Torvalds pada tahun 1991
- ❖ Dalam perkembangannya, Linux dewasa ini merupakan hasil kerjasama dari berbagai pihak

Kernel Linux

- ❖ Versi 0.01 (Mei 1991), hanya dapat berjalan pada prosessor Intel kompatibel-80386 dan pada hardware PC, mempunyai device-driver support yang terbatas
- ❖ Linux 1.0 (Maret 1994)
- ❖ Versi 1.2 (Maret 1995)
- ❖ Versi 2.0 (Juni 1996)

Sistem Linux

- ❖ Menggunakan tool yang dikembangkan sebagai bagian dari Berkeley's BSD operating system, MIT's Window System, dan Free Software Foundation's GNU project
- ❖ Main system libraries pada awalnya diciptakan oleh proyek GNU dan kemudian dikembangkan oleh komunitas Linux

Distribusi Linux

- ❖ Standar, himpunan paket yang belum dikompililasi atau distribusi termasuk sistem dasar Linux, sistem instalasi dan *management utilities*, dan paket siap *install* dari *tool* UNIX
- ❖ Distribusi pertama mengatur paket-paket dengan memberikan cara bagaimana meletakkan semua file ke dalam tempat yang tepat.; distribusi modern berisi manajemen paket lanjutan
- ❖ Distribusi awal yang ada adalah SLS dan Slackware, sedangkan Red Hat distribusi yang populer dari sumber yang komersial, dan Debian merupakan distribusi yang populer dari sumber non komersial

Lisensi Linux

- ❖ Kernel Linux didistribusikan di bawah lisensi GNU General Public License (GPL) di bawah naungan the Free Software Foundation
- ❖ Linux bukan public-domain software
- ❖ Linux adalah free-software
- ❖ Software yang dikeluarkan oleh GPL tidak dapat didistribusikan kembali sebagai produk binary-only

Rancangan Dasar GNU/Linux

- ❖ Linux adalah sebuah sistem yang *multiuser*, *multitasking* dengan *tools* kompatibel UNIX yang lengkap
- ❖ Sistem file-nya mengikuti semantic UNIX tradisional dan mengimplementasikan secara penuh standar model network UNIX
- ❖ Tujuan utama adalah kecepatan, efisiensi, dan standarisasi

Komponen Sistem Linux (1)

- ❖ Seperti kebanyakan implementasi dari UNIX, Linux terdiri dari tiga bagian utama kode.
- ❖ Perbedaan yang paling besar adalah antara Kernel dan komponen-komponen lainnya
- ❖ Kernel bertanggungjawab untuk menjalankan sistem operasi

Komponen Sistem Linux (2)

- ❖ Librari sistem merupakan himpunan standar dari fungsi-fungsi melalui mana aplikasi-aplikasi berinteraksi dan yang mengimplementasikan sebagian besar dari fungsi-fungsi sistem operasi yang tidak membutuhkan hak-hak dari kode kernel
- ❖ Perlengkapan sistem melakukan tugas-tugas manajemen tertentu secara individual

Manajemen proses

Pendahuluan

- ❖ Setiap aplikasi yang dijalankan di Linux mempunyai pengenal yang disebut sebagai process identification number (PID). PID disimpan dalam 32 bit dengan angka berkisar dari 0-32767 untuk menjamin kompatibilitas dengan UNIX. Dari nomor PID inilah Linux dapat mengawasi dan mengatur proses-proses yang terjadi didalam system. Proses yang dijalankan ataupun yang baru dibuat mempunyai struktur data yang disimpan di `task_struct`

Proses dan thread

- ❖ Proses dan thread di dalam Linux merupakan sesuatu yang dapat dikatakan sama. Perbedaannya hanya terletak dalam proses pembuatannya. Proses dibuat dengan *fork system call* sedangkan thread dibuat dengan *clone system call*

Penjadualan

- ❖ Penjadual adalah suatu pekerjaan yang dilakukan untuk mengalokasikan CPU Time untuk tasks yang berbeda-beda
 - Task dari proses yang sedang jalan
 - Task dari device driver

Sinkronisasi kernel

- ❖ Normal kernel code
- ❖ Processor's interrupt control hardware

Penjadualan proses

❖ Algoritma

- time-sharing algorithm
- real-time algorithm

Symmetric multiprocessing

- ❖ SMP di Kernel Linux sebelum 2.0
- ❖ SMP di Kernel linux 2.0
- ❖ SMP di Kernel linux setelah 2.0

Manajemen Memori

- ❖ Memori manajemen pada Linux mempunyai dua komponen:
 - Berurusan dengan mengalokasikan dan membebaskan *pages*, kumpulan *pages*, dan kumpulan blok kecil dari memori
 - Berurusan dengan pengaturan memori virtual, dimana memori dipetakan kedalam alamat ruang dari proses yang berjalan

Manajemen memori fisik

- ❖ Tabel Kernel *pages* memetakan sebanyak mungkin memori fisik kedalam suatu *range* alamat. Alamat fisik yang ditempati oleh kode kernel dan data telah dipesan dan tidak akan pernah dialokasikan untuk tujuan lain
- ❖ Penggunaan dan manajemen dari memori fisik diatur oleh Zone allocator
- ❖ Allocator menggunakan algoritma buddy-heap untuk menjaga dan melihat memori fisik yang ada
- ❖ Alokasi memori di kernel Linux dapat terjadi secara static (driver memesan suatu area tertentu di memori pada saat boot sistem) atau dinamik (dengan menggunakan zone allocator)

Memori Virtual (1)

- ❖ Memori Virtual menjaga agar alamat space terbaca dalam setiap proses dengan cara menciptakan pages dari memori virtual sesuai permintaan, dan mengatur pemakaian dari pages tersebut
- ❖ Manajer Memori Virtual menjaga dua sudut pandang yang berbeda dari alamat space sebuah proses:
 - Sudut pandang logika menggambarkan instruksi berkaitan dengan tatanan dari alamat space
 - Sudut pandang fisik dari tiap alamat space, yang tersimpan dalam table hardware pages untuk proses tersebut

Memori Virtual (2)

- Kernel Linux menjaga satu daerah tetap yang bergantung dari arsitektur dari alamat virtual space yang digunakan oleh tiap proses untuk penggunaannya sendiri

Load dan Eksekusi Program

- ❖ Linux membuat sebuah tabel fungsi untuk meload program, memberikan kesempatan pada tiap fungsi untuk mencoba meload file yang akan berjalan ketika sebuah *system call exec* dipanggil
- ❖ Pertama-tama page file binary ditempatkan pada virtual memori. Hanya ketika program tersebut mencoba mengakses page yang diberikan dan terjadi page fault, maka page tersebut akan diload ke memori fisik

Linking Statis dan Dinamis

- ❖ Linking statis: Sebuah program yang semua library yang digunakan ditaruh secara langsung dalam kode binary program. Kelemahannya adalah tiap program yang dihasilkan harus mengandung kopi system library yang sama
- ❖ Linking dinamis lebih efisien dalam hal memori fisik dan ruang disk, karena hanya memerlukan satu sistem librari untuk tiap program yang dihasilkan

Sistem Berkas Linux

- ❖ Sistem berkas Linux menggunakan layer abstraksi VFS (Virtual File Systems). Terdiri dari dua komponen :
 - Sekumpulan definisi yang mendefinisikan seperti apa sebuah objek berkas harus terlihat
 - Sebuah layer software untuk memanipulasi objek tersebut
- ❖ Tiga tipe objek yang didefinisikan oleh VFS
 - Inode-object
 - File-object
 - File-system-object

Berbagai macam Sistem Berkas Linux

❖ EXT2

EXT2 mendefinisikan topologi file sistem dengan memberikan arti bahwa setiap file pada sistem diasosiasikan dengan struktur data inode.

❖ Proc

Tidak menyimpan data, namun hanya berisi komputasi atas kebutuhan berdasarkan permintaan I/O pengguna.

❖ EXT3

EXT3 adalah peningkatan dari EXT2 file sistem. Kelebihannya antara lain dalam kemudahan pengecekan data setelah 'crash', integritas data, kecepatan throughput, dan migrasi.

❖ Reiser

❖ X

Bentuk Sistem berkas Linux lainnya

❖ Steganographic File System

Menyediakan sedikit perlindungan terhadap instrumen legal ataupun illegal yang mengharuskan pemilik data untuk memberikan kunci deskripsi bagi data yang disimpan sekali kehadiran data terenskripsi dari komputer yang diinspeksi terbangun.

❖ Transparent Criptographic File System

Adalah system berkas terdistribusi yang kriptografik, memungkinkan pengguna untuk mengakses berkas-berkas sensitif secara remote dengan aman.

❖ Web File System

Adalah sistem berkas linux yang baru yang menyediakan sistem berkas seperti antarmuka untuk www. Sistem berkas ini menyediakan caching remote document dan dapat memproses multiple outstanding request secara konkuren

I/O Kernel Linux

- ❖ I/O Linux ini berhubungan dengan bagaimana kernel Linux mengatur device fisik di sistem
- ❖ Linux membagi device ini ke tiga kelas
 - device karakter
 - device blok
 - device jaringan

Device Karakter

- ❖ Device karakter, device paling sederhana dari Linux, diakses sebagai berkas
- ❖ Aplikasi menggunakan system calls standar untuk membukanya, membacanya dan menulisnya dan menutupnya persis seolah devices adalah berkas
- ❖ Digunakan Line discipline sebagai interpreter bagi informasi dari device terminal

Device Blok

- ❖ Device ini pun diakses seperti berkas.
- ❖ Mekanisme untuk menyediakan perangkat operasi berkas yang benar bagi berkas khusus blok yang terbuka sama seperti devices karakter.
- ❖ Untuk kecepatan aksesnya digunakan dua komponen
 - Block Buffer Cache
 - Sebagai pool bagi buffers untuk I/O aktif dan sebuah cache untuk completed I/O
 - The Request Manager
 - Layer software yang mengatur isi buffer baca/tulis ke dan dari block-device driver.

Device Jaringan

- ❖ Device jaringan merupakan sebuah entity yang mengirimkan dan menerima paket-paket data
- ❖ Biasanya ia merupakan device fisik seperti kartu *ethernet*