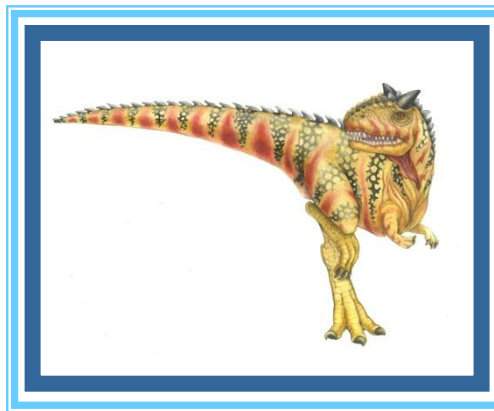


Bab 11:

Antarmuka Sistem File





Bab 11: Antarmuka Sistem File

- Konsep Berkas
- Metode Akses
- Pemasangan Sistem File
- Struktur Disk dan Direktori
- File sharing
- Perlindungan





Tujuan

- Untuk menjelaskan fungsi sistem file Untuk
- menjelaskan antarmuka ke sistem file
- Untuk membahas pengorbanan desain sistem file, termasuk metode akses, berbagi file, penguncian file, dan struktur direktori
- Untuk menjelajahi perlindungan sistem file





Konsep Berkas

- Jenis ruang alamat logis yang
- berdekatan:
 - Data
 - numerik
 - karakter
 - biner
 - Program
- Konten ditentukan oleh pembuat file
 - Banyak jenis
 - Mempertimbangkan **file teks, file sumber, file yang dapat dieksekusi**





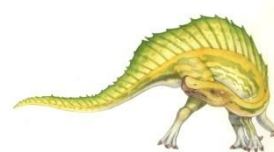
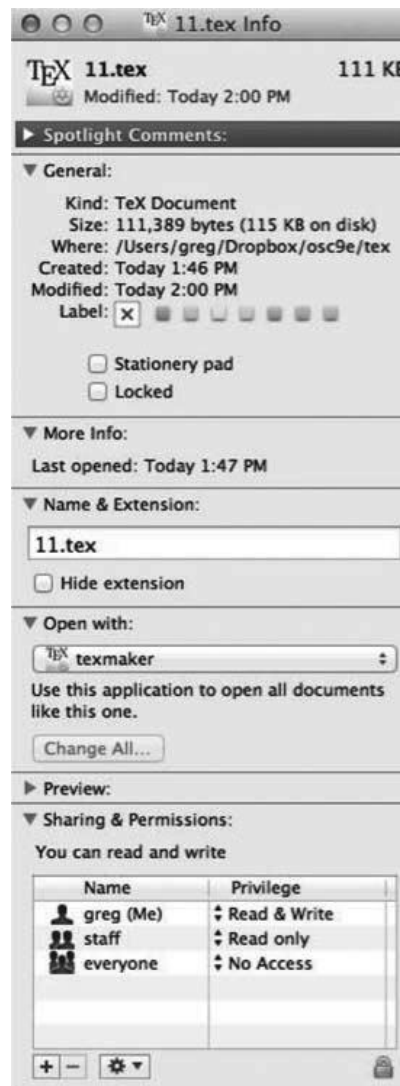
Atribut Berkas

- **Nama**—hanya informasi yang disimpan dalam bentuk yang dapat dibaca manusia
- **Pengidentifikasi**—tag unik (angka) mengidentifikasi file dalam sistem file **Jenis**—
- diperlukan untuk sistem yang mendukung berbagai jenis **Lokasi**—penunjuk ke lokasi file di perangkat **Ukuran**—ukuran file saat ini
-
- **Perlindungan**—mengontrol siapa yang dapat melakukan membaca, menulis, mengeksekusi
- **Waktu, tanggal, dan identifikasi pengguna**—data untuk perlindungan, keamanan, dan pemantauan penggunaan
- Informasi tentang file disimpan dalam struktur direktori, yang disimpan di disk
- Banyak variasi, termasuk atribut file yang diperluas seperti file checksum
- Informasi disimpan dalam struktur direktori





Informasi file di Mac OS X





Operasi Berkas

- File adalah **type data abstrak**
- **Buat**
- **Menulis** - pada **menulis penunjuk** lokasi
- **Membaca** - pada **membaca penunjuk**
- lokasi **Reposisi dalam file** - **mencari**
- **Menghapus**
- **Memotong**
- **Terbuka (F_{saya})** - cari struktur direktori pada disk untuk masuk F_{saya} , dan pindahkan konten entri ke memori
- **Tutup (F_{saya})** - memindahkan konten entri F_{saya} dalam memori ke struktur direktori pada disk





Buka File

- Beberapa bagian data diperlukan untuk mengelola file yang terbuka:
 - **Tabel file terbuka:** melacak file yang terbuka
 - **Penunjuk berkas:** menunjuk ke lokasi baca/tulis terakhir, per proses yang membuka file
 - **Jumlah pembukaan file:** penghitung berapa kali file dibuka – untuk memungkinkan penghapusan data dari tabel file terbuka ketika proses terakhir menutupnya
 - **Lokasi disk file:** cache informasi akses data
 - **Hak akses:** informasi mode akses per proses





Buka Penguncian File

- Disediakan oleh beberapa sistem operasi dan sistem file
 - Mirip dengan kunci pembaca-penulis
 - **Kunci bersama** mirip dengan kunci pembaca – beberapa proses dapat diperoleh secara bersamaan
 - **Kunci eksklusif** mirip dengan kunci penulis
- Memediasi akses ke file
- Wajib atau penasehat:
 - **Wajib** – akses ditolak tergantung pada kunci yang dipegang dan diminta
 - **Penasihat** – proses dapat menemukan status kunci dan memutuskan apa yang harus dilakukan





Contoh Penguncian File – Java API

```
import java.io.*;
import java.nio.channels.*; Contoh
Penguncian kelas publik {
    boolean final public static EKSKLUSIF = false;
    boolean akhir statis publik BERBAGI = benar;
    public static void main(String arsg[]) melempar IOException{
        FileLock sharedLock = null;
        FileLock eksklusifLock = null;
        mencoba {
            RandomAccessFile raf = new RandomAccessFile("file.txt", "rw"); //
            dapatkan saluran untuk file tersebut
            FileChannel ch = raf.getChannel();
            // ini mengunci paruh pertama berkas - Exclusive
            ExclusiveLock = ch.lock(0, raf.length()/2, EXCLUSIVE); /**
            Sekarang ubah datanya. . . */
            // lepaskan kunci
            ExclusiveLock.release();
```





Contoh Penguncian File – Java API (Lanjutan)

```
// ini mengunci paruh kedua file - dibagikan
sharedLock = ch.lock(raf.length()/2+1, raf.length(),
                    BERSAMA);

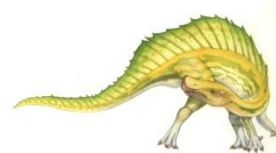
/** Sekarang baca datanya. . . */
lepaskan kuncinya
sharedLock.rilis();
} catch (java.io.IOException ioe) {
    Sistem.err.println(ioe);
} Akhirnya {
    if (exclusiveLock != null)
        ExclusiveLock.release(); if
        (sharedLock != null)
            sharedLock.release();
    }
}
```





Jenis File – Nama, Ekstensi

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information





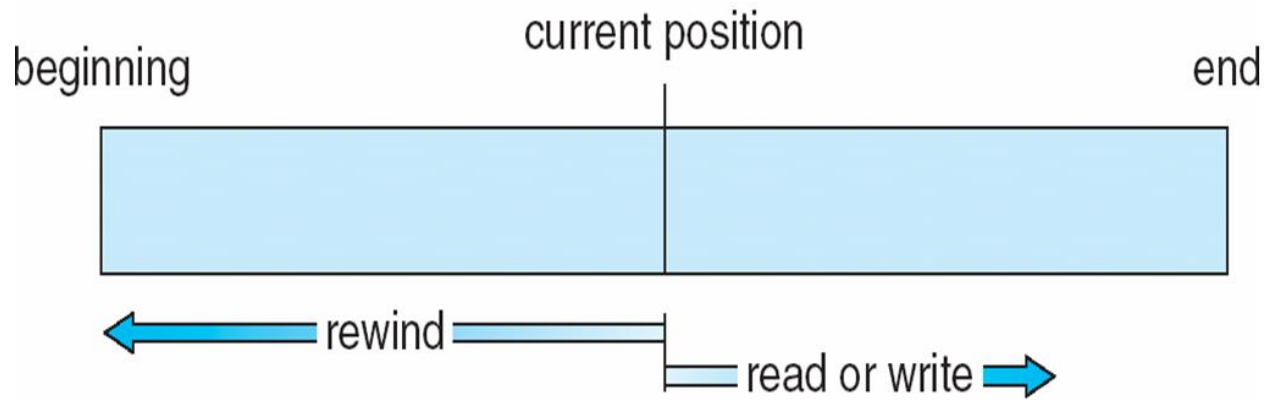
Struktur Berkas

- Tidak ada - urutan kata, byte
- Struktur rekaman sederhana
 - Garis
 - Panjang tetap
 - Panjang variabel
- Struktur Kompleks
 - Dokumen yang diformat
 - File beban yang dapat direlokasi
- Dapat mensimulasikan dua metode terakhir dengan metode pertama dengan memasukkan karakter kontrol yang sesuai
- Siapa yang memutuskan:
 - Sistem operasi
 - Program





File akses berurutan





Metode Akses

- **Akses Berurutan**

baca selanjutnya

menulis Berikutnya

mengatur ulang

tidak ada bacaan setelah tulisan terakhir

(menulis kembali)

- **Akses langsung** -file memiliki panjang tetap **catatan logis**

membaca N

menulis N

posisi ke N

membaca Berikutnya

menulis Berikutnya

menulis kembali N

N =**nomor blok relatif**

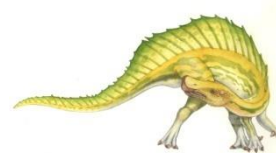
- Nomor blok relatif memungkinkan OS untuk memutuskan di mana file harus ditempatkan
- Melihat **masalah alokasi** di Bab 12





Simulasi Akses Sekuensial pada File Akses Langsung

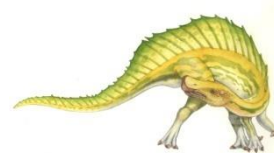
sequential access	implementation for direct access
<i>reset</i>	$cp = 0;$
<i>read next</i>	$read\ cp;$ $cp = cp + 1;$
<i>write next</i>	$write\ cp;$ $cp = cp + 1;$





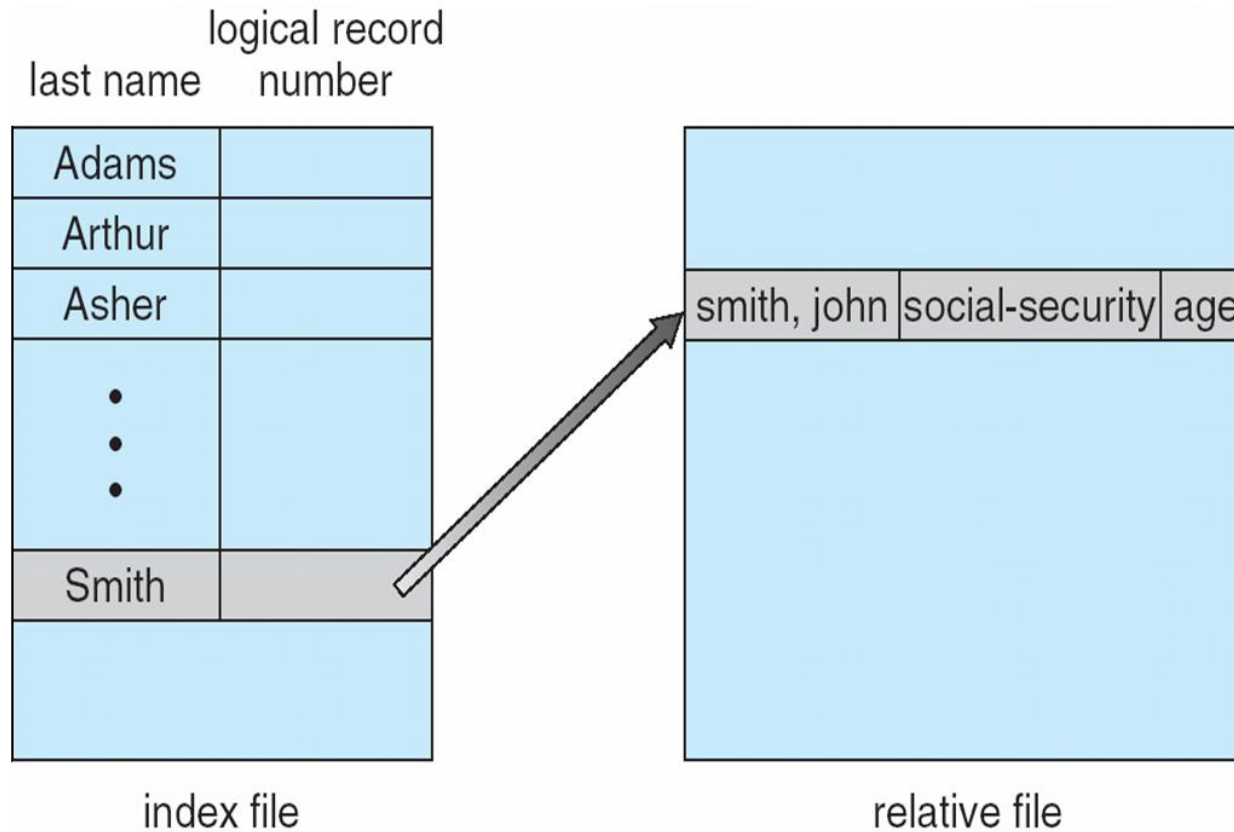
Metode Akses Lainnya

- Dapat dibangun di atas metode dasar Umum melibatkan
- penciptaan sebuah **indeks** untuk file tersebut
- Simpan indeks di memori untuk penentuan lokasi data yang akan dioperasikan dengan cepat (pertimbangkan kode UPC ditambah catatan data tentang item tersebut)
- Jika terlalu besar, indeks (dalam memori) indeks (pada disk)
- metode akses sekuensial yang diindeks IBM (ISAM)
 - Indeks master kecil, menunjuk ke blok disk indeks sekunder
 - File disimpan diurutkan pada kunci yang ditentukan.
 - Semua dilakukan oleh OS
- Sistem operasi VMS menyediakan file indeks dan relatif sebagai contoh lain (lihat slide berikutnya)





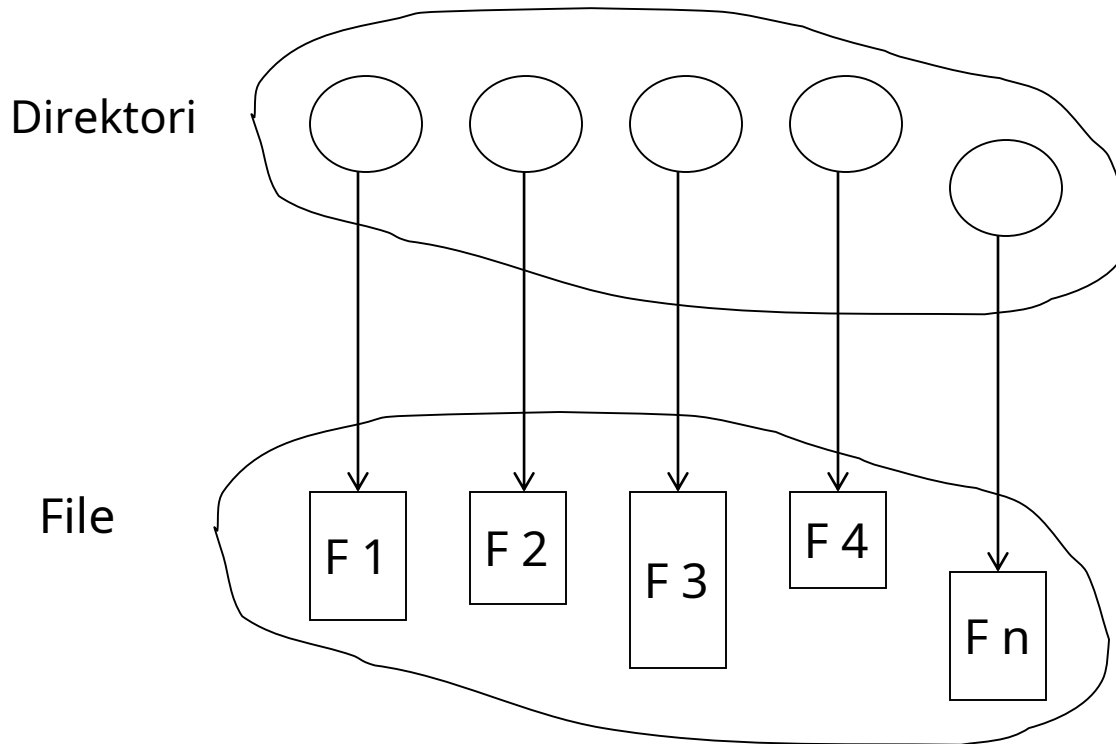
Contoh File Indeks dan Relatif





Struktur Direktori

- Kumpulan node yang berisi informasi tentang semua file



Struktur direktori dan file berada di disk





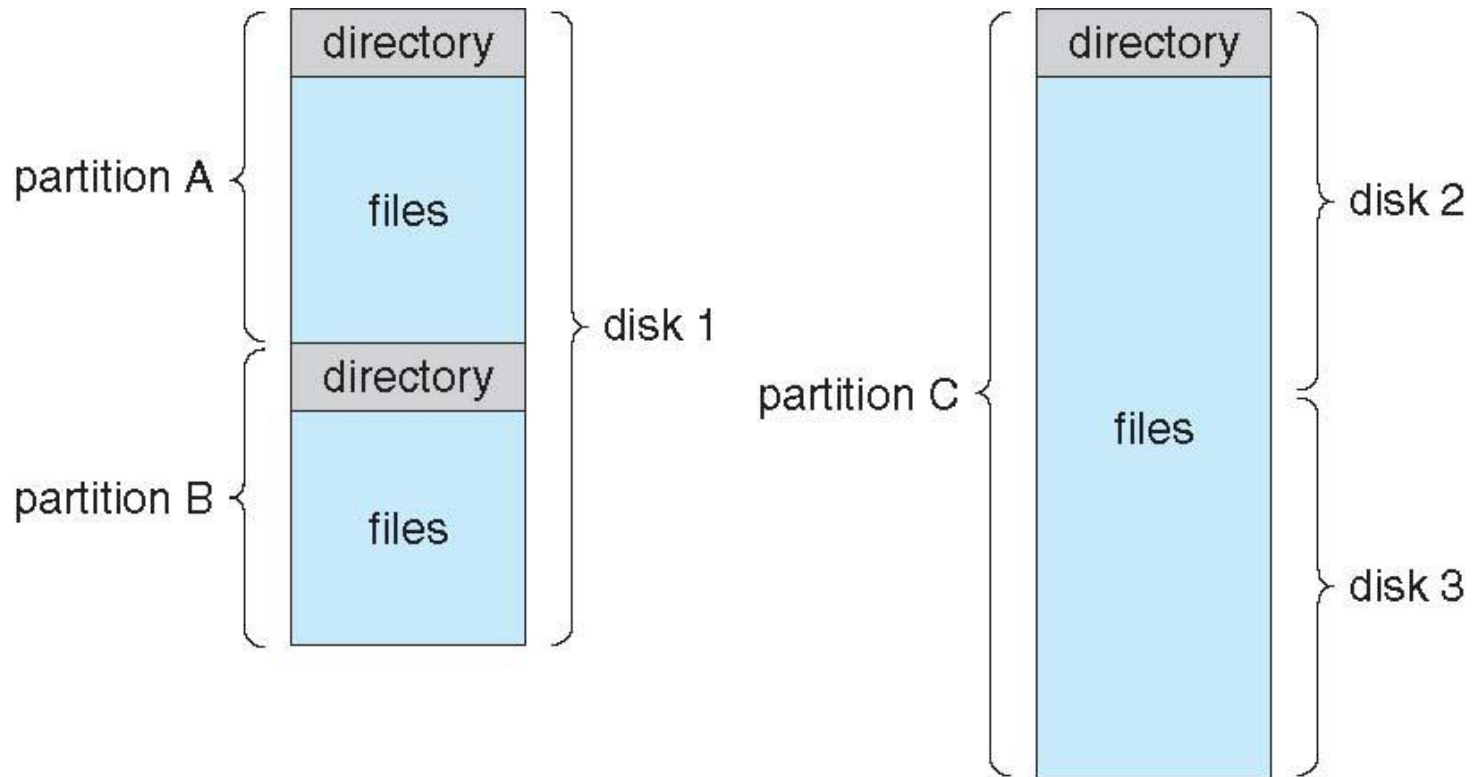
Struktur Disk

- Disk dapat dibagi lagi menjadi **partisi**
- Disk atau partisi bisa **SERANGAN** terlindungi dari kegagalan
- Disk atau partisi dapat digunakan **mentah**—tanpa sistem file, atau **diformat** dengan sistem file
- Partisi juga dikenal sebagai minidisk, irisan Entitas yang
- berisi sistem file yang dikenal sebagai **volume**
- Setiap volume yang berisi sistem file juga melacak informasi sistem file tersebut **direktori perangkat** atau **daftar isi volume**
- Sebaik **sistem file tujuan umum** ada banyak **sistem file tujuan khusus**, sering kali semuanya dalam sistem operasi atau komputer yang sama





Organisasi Sistem File yang Khas





Jenis Sistem File

- Kami kebanyakan berbicara tentang sistem file tujuan umum
- Namun sistem sering kali memiliki banyak sistem file, sebagian bersifat umum dan sebagian bertujuan khusus
- Pertimbangkan Solaris punya
 - tmpfs – FS volatil berbasis memori untuk I/O sementara yang cepat
 - objfs – antarmuka ke memori kernel untuk mendapatkan simbol kernel untuk debugging
 - ctfs – sistem file kontrak untuk mengelola daemon
 - lofs – sistem file loopback memungkinkan satu FS diakses sebagai ganti FS lainnya
 - procfs – antarmuka kernel untuk memproses
 - struktur ufs, zfs – sistem file tujuan umum





Operasi Dilakukan pada Direktori

- Cari file
- Buat berkas
- Hapus file
- Daftar direktori
- Ganti nama file
- Lintasi sistem file





Organisasi Direktori

Direktori disusun secara logis untuk diperoleh

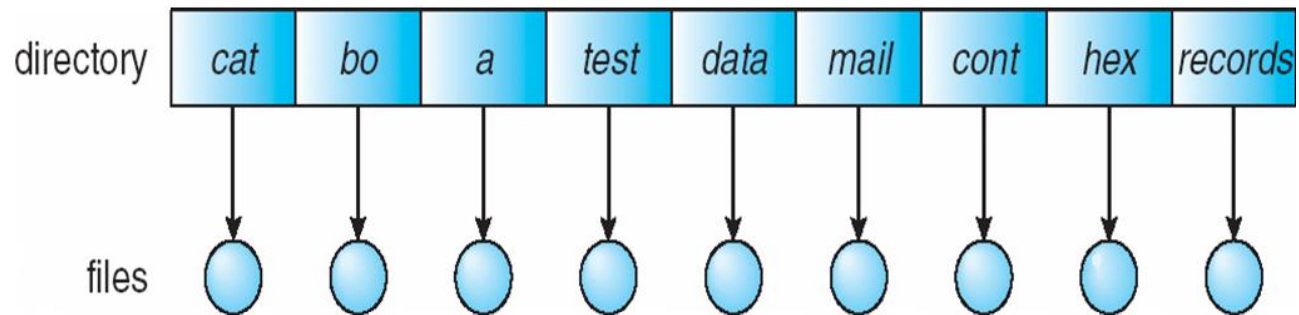
- Efisiensi – menemukan file dengan cepat.
- Penamaan – nyaman bagi pengguna
 - Dua pengguna dapat memiliki nama yang sama untuk file
 - berbeda. File yang sama dapat memiliki beberapa nama berbeda
- Pengelompokan – pengelompokan file secara logis berdasarkan properti, (misalnya, semua program Java, semua permainan,...)





Direktori Tingkat Tunggal

-Direktori tunggal untuk semua pengguna



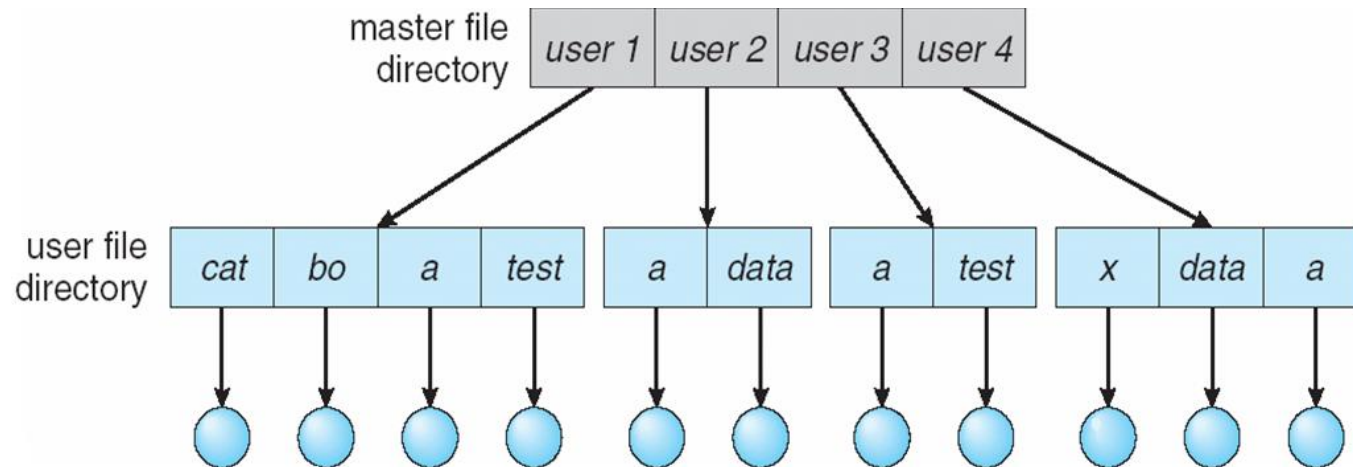
- Masalah penamaan
- Masalah pengelompokan



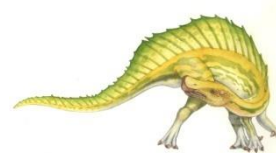


Direktori Dua Tingkat

- Direktori terpisah untuk setiap pengguna

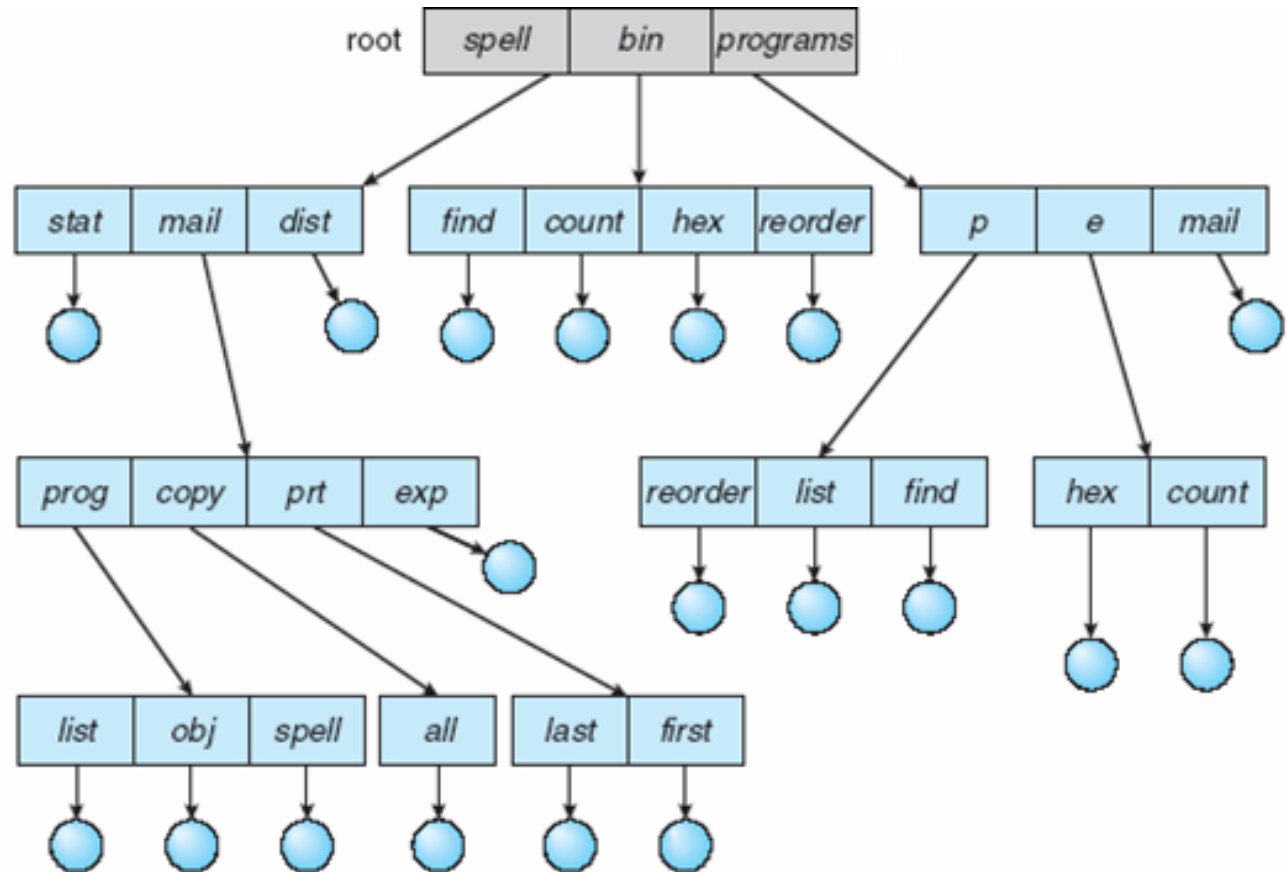


- Nama jalur
- Dapat memiliki nama file yang sama untuk pengguna yang berbeda
- Pencarian yang efisien
- Tidak ada kemampuan pengelompokan





Direktori Terstruktur Pohon





Direktori Terstruktur Pohon (Lanjutan)

- Pencarian yang efisien
- Kemampuan Pengelompokan
- Direktori saat ini (direktori kerja)
 - **cd /spell/mail/prog**
 - **ketik daftar**





Direktori Terstruktur Pohon (Lanjutan)

- **Mutlak** atau **relatif** nama jalur
- Membuat file baru dilakukan di direktori saat ini.
- Hapus file

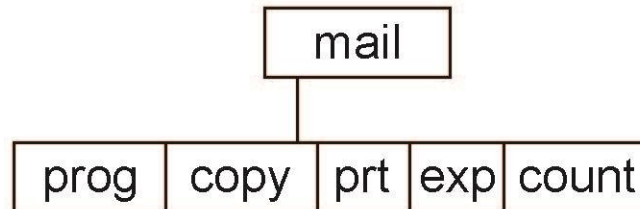
rm <nama-file>

- Membuat subdirektori baru dilakukan di direktori saat ini

mkdir <nama-dir>

Contoh: jika dalam direktori saat ini **/surat**

hitungan mkdir



Menghapus "mail" - menghapus seluruh subpohon yang di-root oleh "mail"





The diagram illustrates a neural network architecture for word processing. At the top, a root node splits into two branches: *dict* and *spell*. The *dict* branch leads to a node with four fields: *list*, *all*, *w*, and *count*. The *spell* branch leads to a node with three fields: *count*, *words*, and *list*. Both branches lead to a common node with three fields: *list*, *rade*, and *w7*. The final output is a vector of three blue circles.





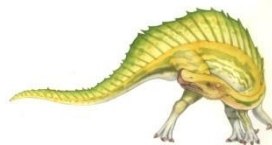
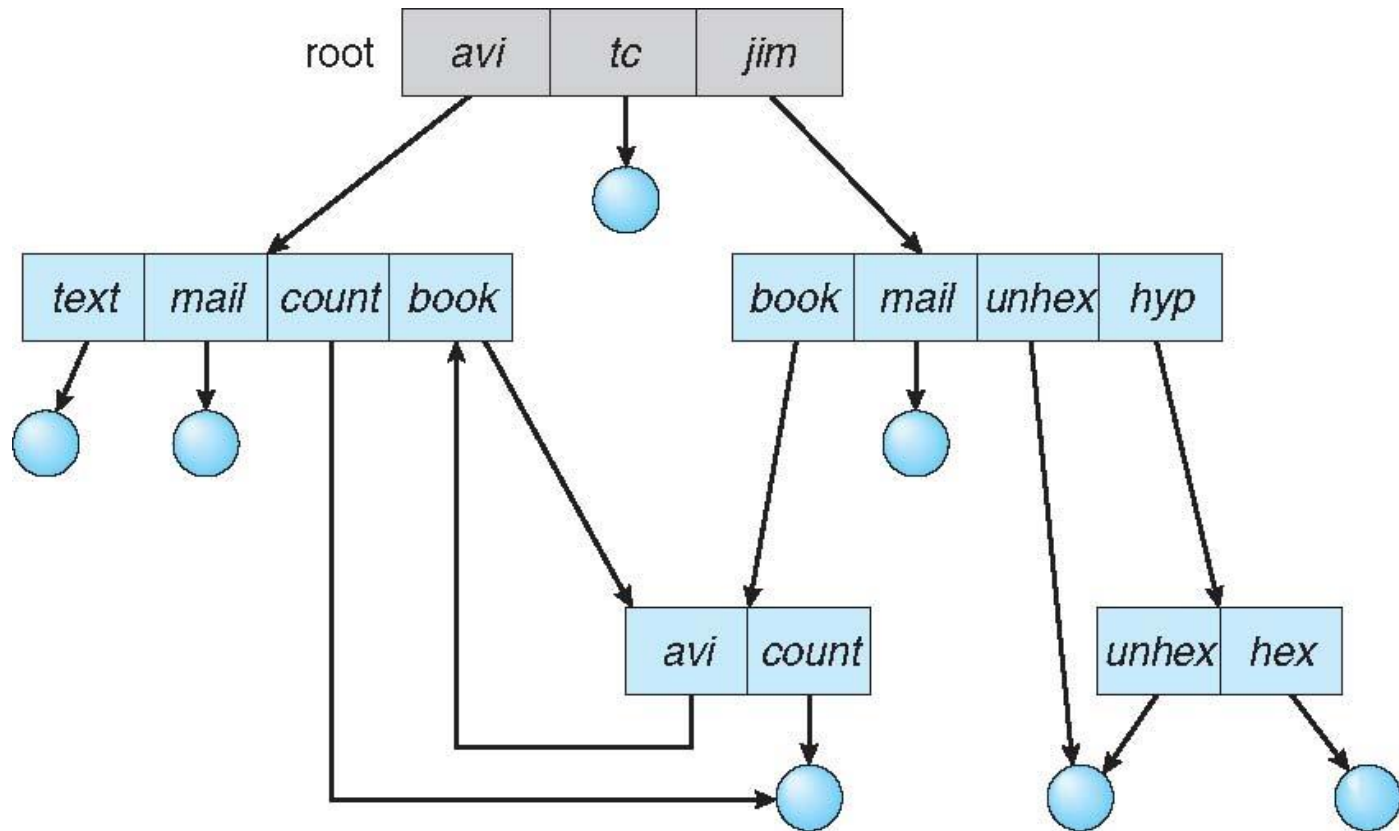
Direktori Grafik Asiklik (Lanjutan)

- Dua nama berbeda (aliasing)
- Jika **diktem** menghapus **daftar**-Solusi penunjuk menjuntai:
 - Backpointer, jadi kita bisa menghapus semua pointer. Ukuran variabel mencatat masalah
 - Backpointer menggunakan solusi Entry-hold-count
 - organisasi rantai daisy
- Jenis entri direktori baru
 - **Tautan**—nama lain (penunjuk) ke file yang ada **Selesaikan**
 - **tautannya**—ikuti penunjuk untuk menemukan file





Direktori Grafik Umum





Direktori Grafik Umum (Lanjutan)

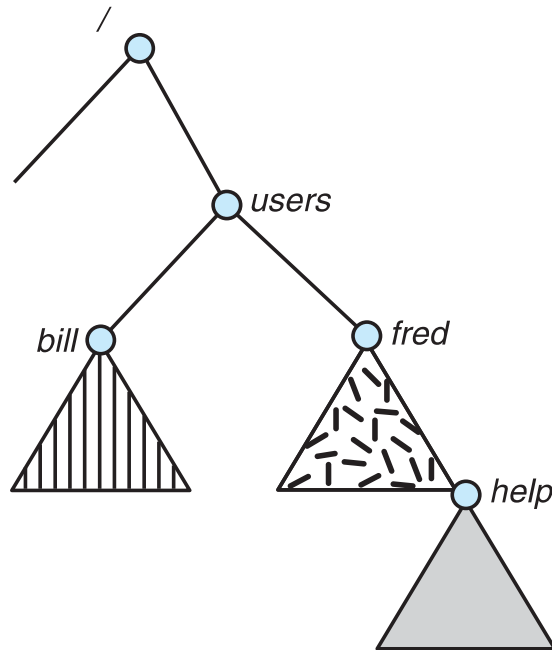
- Bagaimana kami menjamin tidak ada siklus?
 - Izinkan hanya tautan ke file, bukan subdirektori
 - **Pengumpulan sampah**
 - Setiap kali tautan baru ditambahkan, gunakan algoritma deteksi siklus untuk menentukan apakah tautan tersebut OK



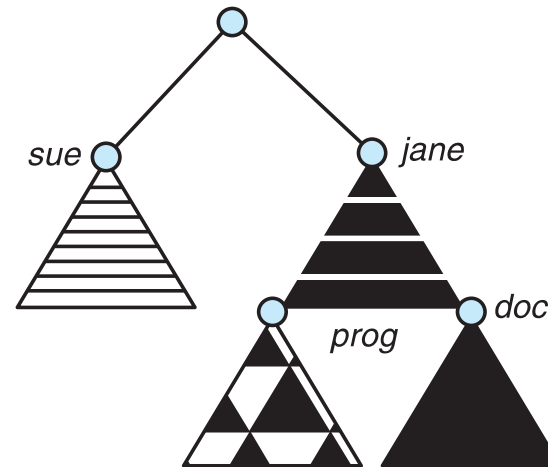


Pemasangan Sistem File

- Sistem file harus **dipasang** sebelum dapat diakses
- Sistem file yang tidak di-mount (yaitu, Gambar 11-11(b)) dipasang di a **titik pemasangan**



(a)

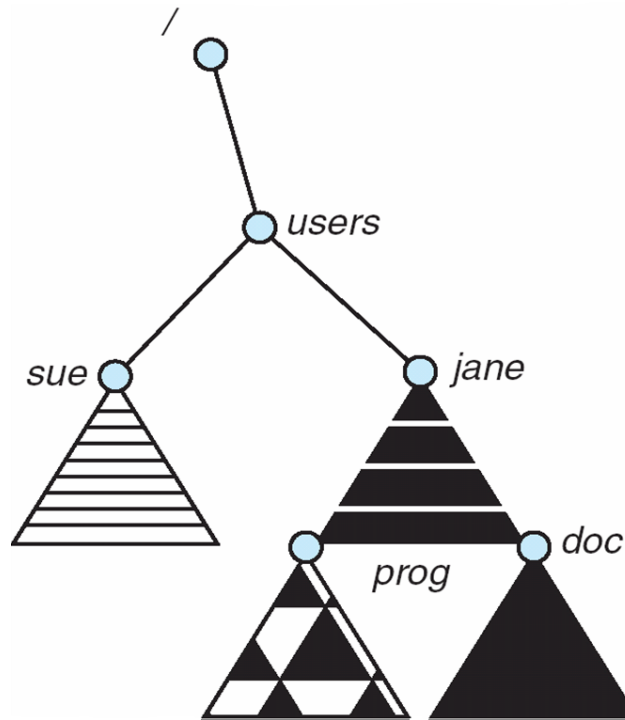


(b)





Titik Gunung





File sharing

- Berbagi file pada sistem multi-pengguna diinginkan.
- Berbagi dapat dilakukan melalui a**perlindungan**skema
- Pada sistem terdistribusi, file dapat dibagikan melalui jaringan
- Network File System (NFS) adalah metode berbagi file terdistribusi yang umum
- Jika sistem multi-pengguna
 - **ID Pengguna** mengidentifikasi pengguna, memungkinkan izin dan perlindungan untuk per pengguna
 - **ID Grup** memungkinkan pengguna untuk berada dalam grup, mengizinkan hak akses grup
 - Pemilik file/direktori
 - Kelompok file/direktori





Berbagi File – Sistem File Jarak Jauh

- Menggunakan jaringan untuk memungkinkan akses sistem file antar sistem
 - Secara manual melalui program seperti FTP
 - Secara otomatis, digunakan dengan lancar **sistem file terdistribusi** Semi otomatis melalui **World Wide Web**
- **Server klien** model memungkinkan klien untuk memasang sistem file jarak jauh dari server
 - Server dapat melayani banyak klien
 - Identifikasi klien dan pengguna-ke-klien tidak aman atau rumit **NFS**
 - adalah protokol berbagi file server-klien UNIX standar **CIFS** adalah protokol Windows standar
 - Panggilan file sistem operasi standar diterjemahkan menjadi panggilan jarak jauh
- Sistem Informasi Terdistribusi (**layanan penamaan terdistribusi**) seperti LDAP, DNS, NIS, Active Directory menerapkan akses terpadu ke informasi yang diperlukan untuk komputasi jarak jauh





Berbagi File – Mode Kegagalan

- Semua sistem file memiliki mode kegagalan
 - Misalnya kerusakan struktur direktori atau data non-pengguna lainnya, yang disebut **metadata**
- Sistem file jarak jauh menambahkan mode kegagalan baru, karena kegagalan jaringan, kegagalan server
- Pemulihan dari kegagalan bisa melibatkan **informasi negara** tentang status setiap permintaan jarak jauh
- **Tanpa kewarganegaraan** protokol seperti NFS v3 menyertakan semua informasi dalam setiap permintaan, memungkinkan pemulihan yang mudah tetapi keamanannya lebih rendah





Berbagi File – Semantik Konsistensi

- Tentukan berapa banyak pengguna yang dapat mengakses file bersama secara bersamaan
 - Mirip dengan algoritma sinkronisasi proses Ch 5
 - Cenderung tidak terlalu rumit karena I/O disk dan latensi jaringan (untuk sistem file jarak jauh)
 - Andrew File System (AFS) mengimplementasikan semantik berbagi file jarak jauh yang kompleks
 - Sistem file Unix (UFS) mengimplementasikan:
 - Menulis ke file terbuka yang langsung terlihat oleh pengguna lain dari file terbuka yang sama
 - Berbagi penunjuk file untuk memungkinkan banyak pengguna membaca dan menulis secara bersamaan
 - AFS memiliki semantik sesi
 - Menulis hanya terlihat pada sesi yang dimulai setelah file ditutup





Perlindungan

- Pemilik/pembuat file harus dapat mengontrol:
 - apa yang bisa dilakukan
 - oleh siapa
- Jenis akses
 - **Membaca**
 - **Menulis**
 - **Menjalankan**
 - **Menambahkan**
 - **Menghapus**
 - **Daftar**





Akses Daftar dan Grup

- Mode akses: baca, tulis, jalankan Tiga
- kelas pengguna di Unix / Linux

			RWX
A)akses pemilik	7	-	1 1 1
			RWX
B)akses grup	6	-	1 1 0
			RWX
C)akses publik	1	-	0 0 1

- Minta manajer untuk membuat grup (nama unik), ucapkan G, dan tambahkan beberapa pengguna ke grup.
- Untuk file tertentu (misalnya *permainan*) atau subdirektori, tentukan akses yang sesuai.

owner	group	public
chmod	761	game

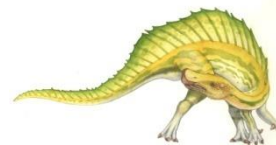
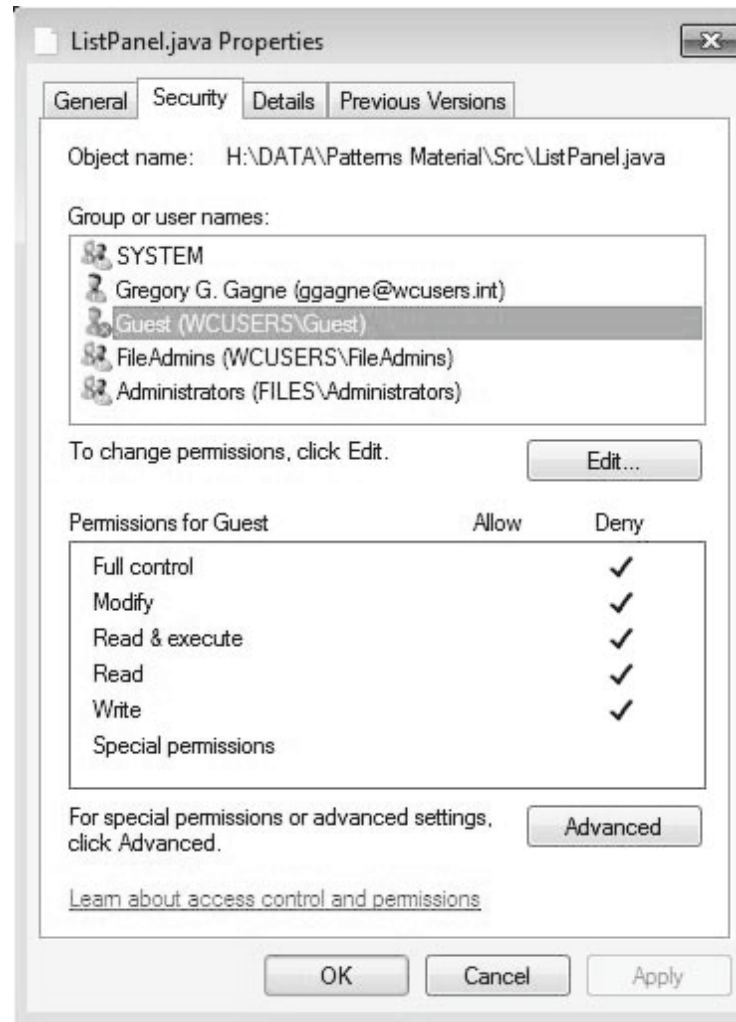
Lampirkan grup ke file

chgrp	G	permainan
-------	---	-----------





Manajemen Daftar Kontrol Akses Windows 7





Contoh Daftar Direktori UNIX

-rw-rw-r--	1 pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5 pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2 pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2 pbg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1 pbg	staff	9423	Feb 24 2003	program.c
-rwxr-xr-x	1 pbg	staff	20471	Feb 24 2003	program
drwx--x--x	4 pbg	faculty	512	Jul 31 10:31	lib/
drwx-----	3 pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3 pbg	staff	512	Jul 8 09:35	test/



Akhir Bab 11

