

Bab 5. File Service

5.1 Pendahuluan

File adalah kumpulan informasi yang berhubungan dan tersimpan dalam *secondary storage*.

Tipe File :

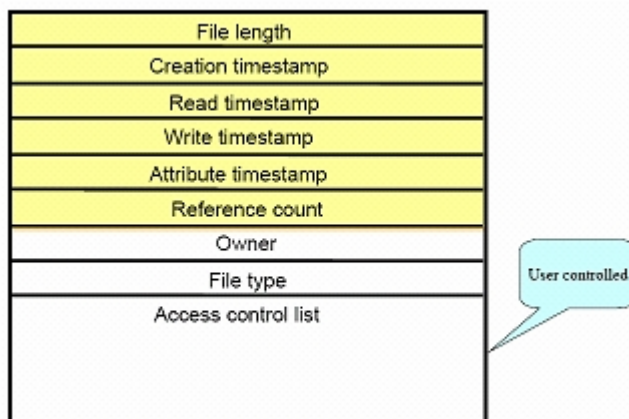
- Data (*character, numeric, binary*)
- Program

Atribut File

- Nama yaitu menentukan nama file yang dimaksud
- Tipe

KARAKTER	ARTI
-	File biasa
d	Direktori
l	Symbolic Link
b	Block Special File
c	Character Special File
s	Socket Link
p	FIFO

- Lokasi, yaitu merupakan pointer atau penunjuk ke device dan lokasi file pada device tersebut berada
- Ukuran, yaitu ukuran file pada saat itu, baik dalam byte, huruf ataupun blok
- Waktu pembuatan dan identitas pembuat, yaitu menentukan kapan file terakhir dimodifikasi
- Proteksi, berisi informasi mengenai kontrol akses, misalnya siapa saja yang boleh membaca, menulis dan mengeksekusi file
- Informasi lain tentang file
Informasi ini biasanya disimpan untuk :
 - Pembuatan file
 - Modifikasi terakhir yang dilakukan pada file
 - Penggunaan terakhir file



Operasi pada File

- Membuat (Create)
 - Menemukan free space
 - Entry baru dibuat dalam tabel direktori yang mencatat nama dan lokasi
 - Ukuran yang diinisialisasi 0
- Menulis (Write)
 - OS melihat ke direktori untuk mencari lokasinya dalam disk
 - Melakukan transfer dari memori ke lokasi dalam disk (suatu pointer digunakan sebagai penunjuk lokasi penulisan berikutnya)
 - Entry dalam direktori di update
- Membaca (Read)
 - OS melakukan hal yang sama dengan penulisan file kecuali operasinya membaca dari lokasi dalam disk ke dalam memori
- Menghapus
 - OS melihat ke direktori mencari entry dengan nama yang dimaksud
 - Membebaskan space yang teralokasi
 - Menghapus entry
- Reposition dalam file
 - OS melihat ke direktori untuk mencari entry yang dimaksud,
 - Pointer di set dengan harga (lokasi) tertentu yang diberikan
- Menghapus dengan menyisakan atribut (Truncate)
 - sama dengan menghapus file kecuali entry tidak dihapuskan tapi ukuran file diisi 0

Contoh Tipe File

file type	usual extension	function
executable	exe, com, bin or none	read 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, rrf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	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	binary file containing audio or A/V information

5.2 Layanan File (*File Service*)

File service adalah suatu perincian atau pelayanan dari file sistem yang ditawarkan pada komputer client.

File System merupakan struktur logika yang digunakan untuk mengendalikan akses terhadap data yang ada pada disk. File System menyediakan mekanisme untuk penyimpanan data dan program yang dimiliki oleh sistem operasi serta seluruh pengguna dari sistem computer

File System terdiri dari dua bagian:

- Kumpulan file yang masing-masingnya menyimpan data-data yang berhubungan
- Struktur direktori yang mengorganisasi dan menyediakan informasi mengenai seluruh file dalam system

Masing-masing Sistem Operasi menggunakan cara yang berbeda dalam mengatur dan mengendalikan akses data dalam disk. Cara pengaturan dan pengendalian ini tidak bergantung pada spesifikasi dari perangkat keras. Misalnya suatu hard disk dengan spesifikasi yang sama dapat menggunakan file system yang berbeda. Struktur logika dari suatu hard disk memiliki pengaruh yang besar terhadap kinerja, daya tahan, dan pengembangan dari suatu disk. Penetapan file system dalam suatu disk dilakukan pada saat disk tersebut di **format**.

File System merupakan interface yang menghubungkan sistem operasi dengan disk. Ketika program aplikasi yang sedang dijalankan memerlukan pembacaan file dari hard disk, sistem operasi meminta file system untuk membuka file yang diinginkan. File system harus mengetahui lokasi penyimpanan file yang dibaca. Setelah menemukan lokasinya, file system membaca data yang ada dan mengirimkan data tersebut pada sistem operasi.

Karakteristik File System

File system bertugas dalam organisasi, storage, retrieval (memperbaiki), penamaan, sharing, dan proteksi terhadap files. File dikirim ke dalam disk atau non-volatile storage media lainnya. File meliputi data dan atribut. File system didesain untuk menyimpan dan mengatur banyak dan besar file dengan fasilitas untuk membuat, memberi nama dan menghapus file. File system juga bertanggung jawab untuk pengontrolan dari akses file, akses terbatas ke file oleh user yang berhak dan tipe-tipe dari akses yang diminta.

Struktur File System

Directory module:	relates file names to file IDs
File module:	relates file IDs to particular files
Access control module:	checks permission for operation requested
File access module:	reads or writes file data or attributes
Block module:	accesses and allocates disk blocks
Device module:	disk I/O and buffering

Kebutuhan File System Terdistribusi

1. *Transparency*

Keseimbangan antara fleksibilitas dan skalabilitas terhadap kompleksitas dan performansi dalam desainnya. Client tidak menyadari adanya lingkungan terdistribusi. Transparency dalam File System Terdistribusi:

- *Access Transparency*
Suatu client tidak mengetahui bahwa file-file di distribusikan dan dapat mengaksesnya dengan cara yang sama seperti file lokal yang diakses.
- *Location Transparency*
Suatu nama konsisten mencakup lokal seperti halnya file yang jauh. Nama dari suatu file tidak memberi tahu lokasinya.
- *Concurrency Transparency*
Semua client mempunyai pandangan yang sama dari suatu status file sistem. Ini berarti bahwa jika satu proses sedang memodifikasi suatu file, proses lain yang berada pada suatu sistem yang sedang mengakses file tersebut akan dapat melihat modifikasi tersebut di dalam suatu cara yang jelas.
- *Failure Transparency*
Suatu client dan client program perlu beroperasi dengan tepat setelah adanya suatu kegagalan server.
- *Heterogeneity*
File service harus dapat menyediakan layanan walaupun berbeda hardware dan operating system platform.
- *Scalability*
Suatu file sistem harus dapat bekerja dengan baik di dalam lingkungan yang kecil dan juga dapat bekerja dalam skala yang besar.
- *Replication Transparency*
Untuk mendukung skalabilitas, kita mungkin dapat membuat tiruan suatu file menyebrangi beberapa server.
- *Migration Transparency*
Suatu file harus dapat berpindah tanpa sepengetahuan clientnya.

2. *Concurrent File Updates*

Perbaruan file serentak bersamaan antara file server dan client. Kebanyakan arah sistem mengikuti standar UNIX dalam memberikan pelayanan advisory atau mandatory file atau record file level locking.

3. *File Replication*

Replikasi dapat untuk share load, untuk mempertinggi fault tolerance, dan untuk mempertinggi scalability. Kebanyakan sistem yang ada dapat melayani caching dengan replication terbatas. Sebagian dapat melayani full replication.

4. *Hardware dan Operating Systems*

Heterogenitas atau kesamaan adalah kebutuhan yang sangat penting dalam melayani keterbukaan.

5. *Fault Tolerance*

Service harus terus menerus beroperasi walaupun terjadi kesalahan atau error pada client ataupun server

6. *Consistency*

7. *Security*

Semua sistem melayani mekanisme akses kontrol berdasarkan daftar akses kontrol (access control lists)

8. *Efficiency*

Sistem harus dapat melayani perbandingan performance apakah lebih baik atau tidak.

Pilihan Desain dalam File Service

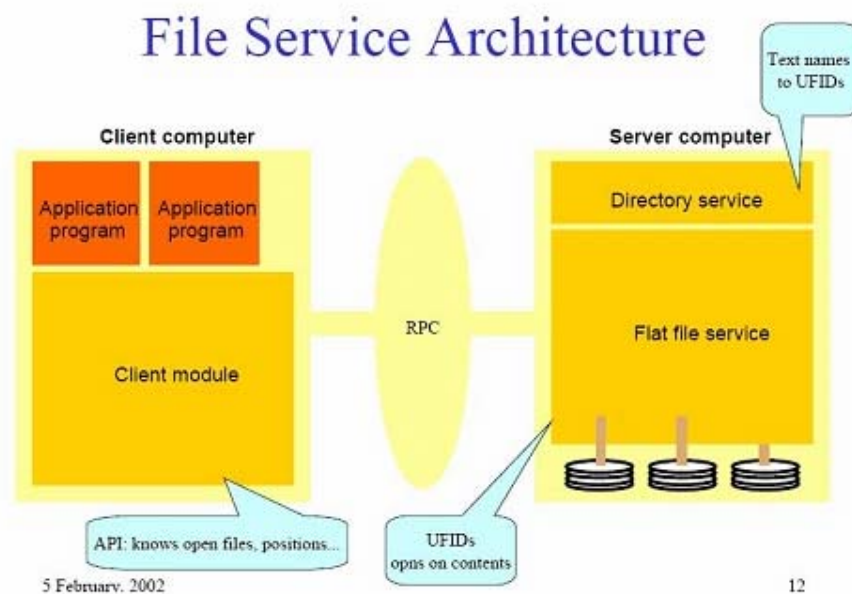
1. Stateful

- server menyimpan informasi tentang file yang terbuka, dan posisi file sekarang (current position).
- Open (dibuka) sebelum access dan kemudian ditutup
- Menyediakan file locks

2. Stateless

- server tidak menyimpan state informasi
- operasi file harus mengandung semua yang diperlukan (memuat pesan yang lengkap).
- Dapat dengan mudah di-recovery apabila terjadi client ataupun server crash
- Membutuhkan extra lock server untuk mempertahankan state

Arsitektur File Service



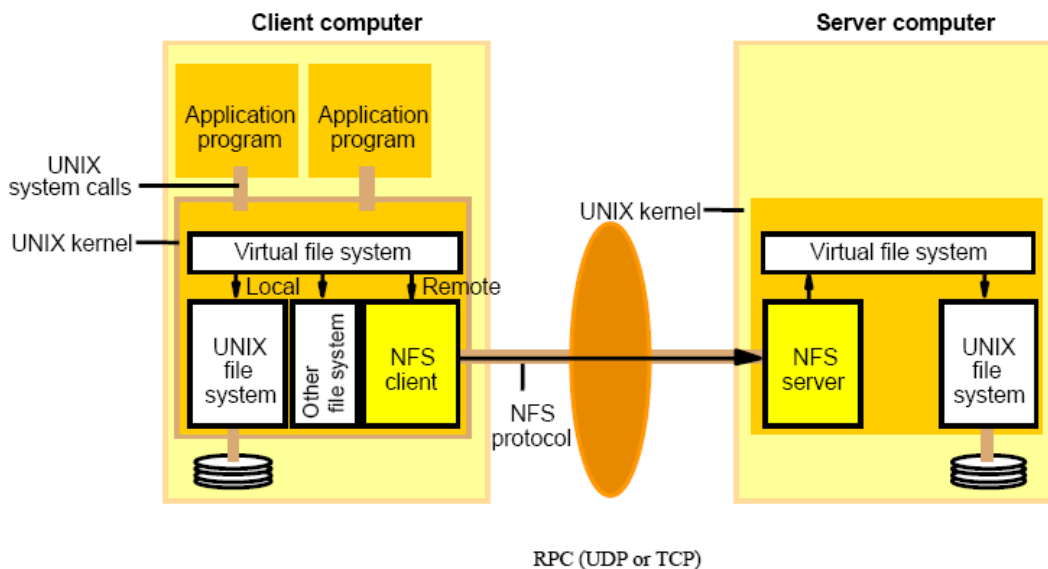
- Layanan Flat File (*Flat file service*)
 - Beroperasi pada isi file
 - Unique File Identifier (UFID)
- Layanan Direktori (*Directory Service*)
 - Menyediakan pemetaan antara nama sebagai teks ke UFIDs
- Modul Client
 - Menyediakan aplikasi untuk mengakses layanan file jarak jauh secara transparan
 - Menyimpan status (state): *open files, positions*
 - Mengetahui lokasi jaringan dari flat file dan directory server

Contoh File System

▪ NFS (Network File System)

Network File System (NFS) merupakan sebuah protokol yang dikembangkan oleh Sun Microsystems pada tahun 1984 dan NFS didefinisikan dalam RFC 1094, 1813 dan 3530 sebagai “DFS” yang mengizinkan sebuah komputer untuk mengakses file melalui network serasa akses file di disk local. NFS merupakan protokol yang sangat mendukung dalam pengaplikasian suatu file system yang terdistribusi.

NFS architecture



Tujuan dari NFS adalah untuk memungkinkan terjadinya pertukaran sistem berkas secara transparan antara mesin-mesin bebas tersebut. Hubungan yang terjadi di sini didasarkan pada hubungan *client-server* yang menggunakan perangkat lunak *NFS server* dan *NFS client* yang berjalan diatas *workstation*.

NFS didesain agar dapat beroperasi di lingkungan ataupun jaringan yang heterogen yang meliputi mesin, platform, sistem operasi, dan arsitektur jaringan. Ketidaktergantungan ini didapat dari penggunaan RPC primitif yang dibangun diatas protokol *External Data Representation (XDR)*.

Jika misalnya terjadi sebuah pertukaran sistem berkas antara *server* dan *client*, maka pertukaran sistem berkas yang terjadi disini harus dipastikan hanya berpengaruh pada tingkat *client* dan tidak mempengaruhi sisi *server*, karena server dan client adalah mesin yang berbeda dan sama-sama bebas. Untuk itu, mesin *client* harus melakukan operasi *mount* terlebih dahulu agar *remote directory* dapat diakses secara transparan.

Protokol NFS

NFS umumnya menggunakan protokol *Remote Procedure Call (RPC)* yang berjalan di atas UDP dan membuka *port* UDP dengan *port number* 2049 untuk komunikasi antara *client* dan *server* di dalam jaringan. *Client* NFS selanjutnya akan mengimpor

sistem berkas *remote* dari *server* NFS, sementara *server* NFS mengeksport sistem berkas lokal kepada *client*.

Mesin-mesin yang menjalankan perangkat lunak NFS *client* dapat saling berhubungan dengan perangkat lunak NFS *server* untuk melakukan perintah operasi tertentu dengan menggunakan *request* RPC. Adapun operasi-operasi yang didukung oleh NFS adalah sebagai berikut:

- a. Mencari berkas di dalam direktori.
- b. Membaca kumpulan direktori.
- c. Memanipulasi link dan direktori.
- d. Mengakses atribut berkas.
- e. Membaca dan menulis berkas.

Perlu diketahui bahwa server NFS bersifat *stateless*, yang artinya setiap *request* harus mengandung argumen yang penuh dan jelas sebab server NFS tidak menyimpan sejarah informasi *request*. Data yang dimodifikasi harus di *commit* ke *server* sebelum hasilnya di kembalikan ke *client*. NFS protokol tidak menyediakan mekanisme *concurrency-control*.

Beberapa **manfaat NFS** diantaranya ialah

- Lokal workstations menggunakan ruang disk lebih kecil
- Pemakai tidak harus membagi direktori home pada setiap mesin di jaringan
- Direktori home dapat di set up pada NFS server dan tersedia melalui jaringan
- Device penyimpanan seperti floppy disk, CDROM drives, dll dapat digunakan oleh mesin lainnya

Kerugian /Kelemahan NFS

- Desain awal hanya untuk jaringan yang lokal dan tertutup
- Security
- Congestion (Traffic yang tinggi bisa menyebabkan akses lambat)

▪ **AFS (Andrew File System)**

Ada persamaan dan perbedaan antara AFS dan NFS.

Persamaan:

AFS sama seperti NFS menyediakan akses menuju shared files secara transparan. Akses menuju ke file AFS adalah melalui normal unix file primitives (tanpa modifikasi atau rekompilasi). AFS kompatibel dengan NFS.

Perbedaan:

Secara desain dan implementasi, AFS berbeda dengan NFS, perbedaan utama, NFS didesain untuk dapat menangani active users dengan jumlah yang lebih banyak daripada distributed file systems yang lain. Kunci AFS untuk menangani jumlah active users yang besar terletak pada kemampuan caching seluruh files pada client node.

Karakter AFS

AFS mempunyai dua karakter desain yang unik :

- *Whole file serving*
Seluruh konten dari direktori dan file dikirim kepada komputer client oleh AFS servers (di AFS3, file yang ukurannya lebih dari 64kbytes ditransfer dalam 64kbyte chunks).

- *Whole file caching*
Copy dari sebuah file chunk yang telah ditransfer di komputer client disimpan dalam cache pada local disk. Cache disini mengandung ratusan file yang sering dipakai di dalam sebuah komputer.

5.3 Access Control

- Dalam sistem file UNIX
 - hak akses diperiksa berdasarkan mode akses (*read, write, execute*) ketika membuka file
 - identitas user diperiksa ketika login
- Dalam sistem file terdistribusi
 - hak akses harus diperiksa pada server
 - *RPC unprotected*
 - dimungkinkan melakukan pemalsuan identitas, tapi ini merupakan resiko keamanan
 - user ID biasanya dikirim dengan setiap request
 - stateless

5.4 Layanan Direktori

Tugas utama dari layanan direktori adalah memetakan nama teks ke UFIDs

Operasi pada layanan direktori

<i>Lookup(Dir, Name) -> FileId</i> — throw <i>NotFound</i>	Locates the text name in the directory and returns the relevant UFID. If <i>Name</i> is not in the directory, throws an exception
<i>AddName(Dir, Name, File)</i> — throw <i>NameDuplicate</i>	If <i>Name</i> is not in the directory, adds (<i>Name, File</i>) to the directory and updates the file's attribute record. If <i>Name</i> is already in the directory: throws an exception.
<i>UnName(Dir, Name)</i> — throw <i>NotFound</i>	If <i>Name</i> is in the directory: the entry containing <i>Name</i> is removed from the directory. If <i>Name</i> is not in the directory: throws an exception.
<i>GetNames(Dir, Pattern) -> NameSeq</i>	Returns all the text names in the directory that match the regular expression <i>Pattern</i> .

Referensi:

1. Implementasi Sistem File, http://lecturer.eepis-its.edu/~arna/Diktat_SO/10.Implementasi%20Sistem%20File.pdf
2. File Sistem, <http://bebas.vlsm.org/v06/Kuliah/SistemOperasi/BUKU/bahan/bahan-bab6.pdf>
3. Adang Suhendra, Sistem File Terdistribusi, <http://staffsite.gunadarma.ac.id/adang/index.php?stateid=download&id=4855&part=files>
4. Ayu Anggriani dkk., Tugas Kuliah Pengantar Sistem Terdistribusi, 2008.