SUMMARY
FILE SYSTEM IN THE OPERATING SYSTEM

Make To A Half Condition Participate Of

Final Examination Semester 7th

With Lecturer Ir. Bana Handaga, MT., Ph.D

Created by :

Didik Maryono

L200144017

INFORMATICS DEPARTMENT

FACULTY OF COMMUNICATION AND INFORMATION

UNIVERSITY OF MUHAMMADIYAH SURAKARTA

REASON TOPIC

1. Understanding of System File
2. File Concept
3. Method of Access
4. Structure of Directory
5. System file to Mounting
6. Protection
7. References

1. Understanding of System File

A file system is clearly-defined method that computer's operating system uses to store, catalogue, and retrieve files.

2. File Concept

File is a saving unit of logic abstraction by operating system from storage. File has been information saved on the seconder storage (like magnetic disk, magnetic tape and optical disk). The information on the file definition by that creator. A file has structure order by that type. File type following good data from numeric data, character or binary, and program like source program, object program and executable program.

2.1. File Attribute

A file has different attribute between the operating system with other, but general order following like this :

a. Name : saved information in to model that has been read people.
b. Type : needs system to support different type
c. Location : pointer to the file location on the ware
d. Size : file size this session
e. Protection : controlling who has been read, write and executions
f. Time, date and user identifier : data to monitoring protection, security and uses.

File information save on the directory structure has been control by disk.

2.2. Operating in the File

As Type abstract data, it has need to be defined to Operating that has been being model by file. It has six basic Operating develop as call system following :

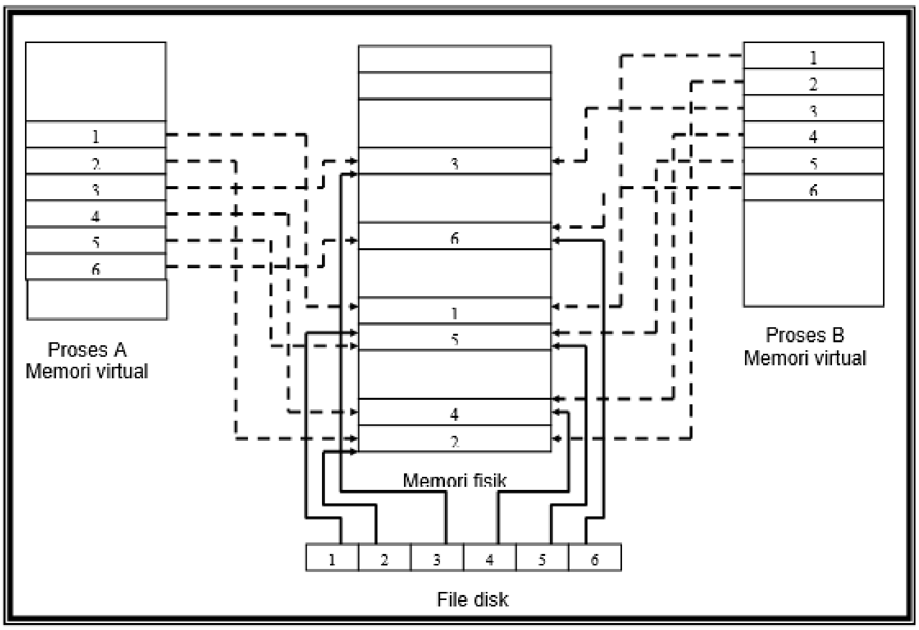
a. Create
b. Write
c. Read
d. File Seek
e. Delete
f. Truncate
g. Open (Fi) find structure of directory for Fi entry and moved the contain of entry in to memory.
h. Close (Fi) moved entry contain Fi in the memory to structure of directory on the disk.

Additional Operating usually done to file is :

a. Append to new information on the last file exist
b. Rename file exist
c. Copy File

More file Operating involve finding the directory for input communicate with the file. To avoid fixed search, some system will be open the file if that file first time active. Operating System save little table that information contain about all of open-file table. If the file not uses again, then it done closing by process and operating system move that file from open-file table. Some information related with opening file following :

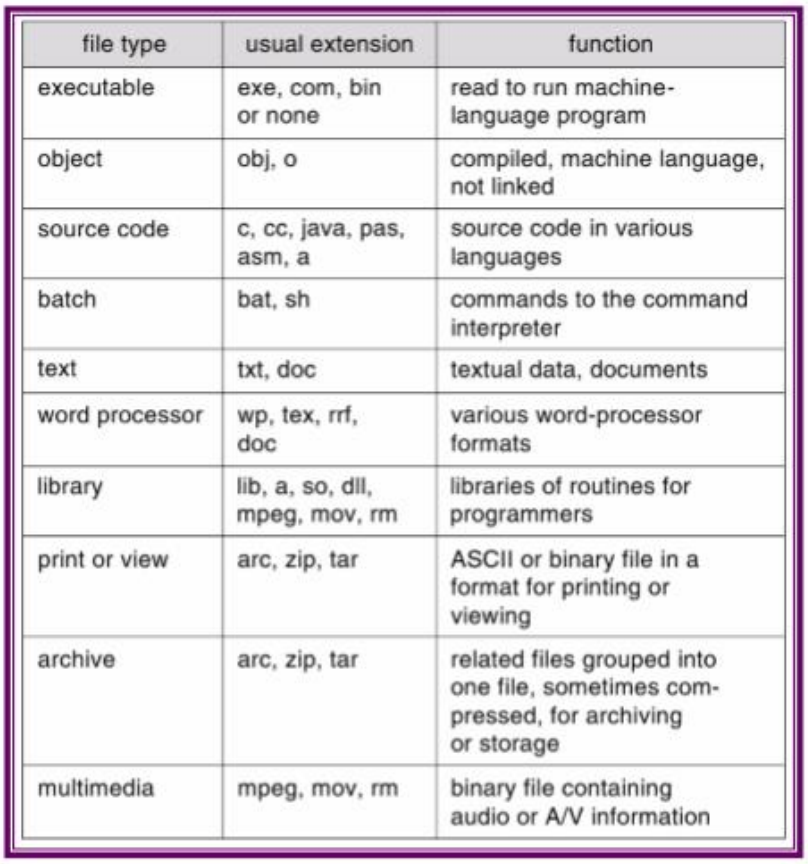
a. File Pointer
b. Sum file opened
c. Location of file on the disk



Gambar 9-1 : Pemetaan file ke memori

2.3. File Type

One important consideration in designing the file system and overall operating system is whether the operating system recognizes and supports the file system. When the operating system recognizes the type of a file, it can be performed operations against the file in a rational way. For example a user trying to print an executable file can be prevented by the operating system because the file is a binary program. A common technique for implementing file types is to enter a file type as part of a file name. The file name is divided into two parts: name and extension (as in MS-DOS) as in Figure 9-2. Each file has a creator attribute containing the name of the program that created it (as in MS Windows / Apple Macintosh). This attribute is set by the operating system when using the system call create. When the user opens the file by double-clicking the mouse on the icon of the file, the program is automatically displayed.



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

Gambar 9-2 : Tipe File

UNIX uses the magic number stored at the beginning of the file to indicate the file type of an executable program, batch file (shell script), postscript file and so on. Not all files have magic numbers, so type information can not be described. UNIX does not store the name of the program maker. UNIX also allows the extension name of hidden files, so that the user can specify the file type itself and not depend on the operating system.

2.4. Structure File

The file type is also used to show the internal structure of the file. Certain files must confirm to the required structure understood by the operating system. For example the operating system requires executable files that have a special structure that can determine where the location of memory and the location of the first instruction. Some operating systems use a set of file system support systems with a number of special operations for file manipulation with these structures. This is a weakness in operating systems that support more than one file structure. If the operating system determines 10 different file structures, it is necessary to include code to support the file structure. Each file needs to be defined as one of the file types supported by the operating system. Some operating systems such as UNIX and MS-DOS only support a number of file structures. UNIX specifies each file to be an 8 bit byte bit and that bit is not translated by the operating system. This scheme has maximum flexibility, but little support. Each application program must include its own code to translate input files into the proper structure. At least all SO must support at least one executable file structure so that the system can load and run the program.

2.5. Internal Structure of File

Internally, the disk system has a block size determined by the size of a sector. All I / O disks are formed in a single-size block of physical units. The physical record size is incorrect with the length of the logical record. Logical records vary in length. The solution is to send a number of logical records to a physical block. Logical record sizes, physical block size and delivery techniques determine how many logical records are in physical blocks. Delivery can be done by user application program or operating system. File is a row of blocks. All the basic functions of I / O are operated on the blocks. Conversion from logical records to physical blocks is related to simple software.

3. Method Of Access

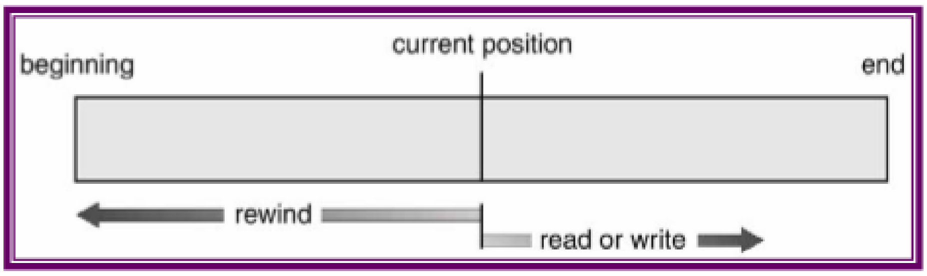
The file saves the information. When used, the information must be accessed and read to memory. There are several ways to access information in the file that is sequential access, direct access or relative access and other access methods.

3.1. Sequential Access

Sequential access is the simplest access method. The information in the file is processed in sequence, one record is accessed after another record. This access method is based on a tape model of a file that works with sequential access or random-access devices. Operations in sequential access consist of :

a. Read next
b. Write next
c. Reset
d. No read after last write (rewrite)

The read operation reads the next part of the file and automatically adds a pointer file that tracks the location of the I / O. The write operation adds to the end of the file and to the end of the new reading material (new end of file). The file can be reset to the start and a program to jump forward or back to n record.



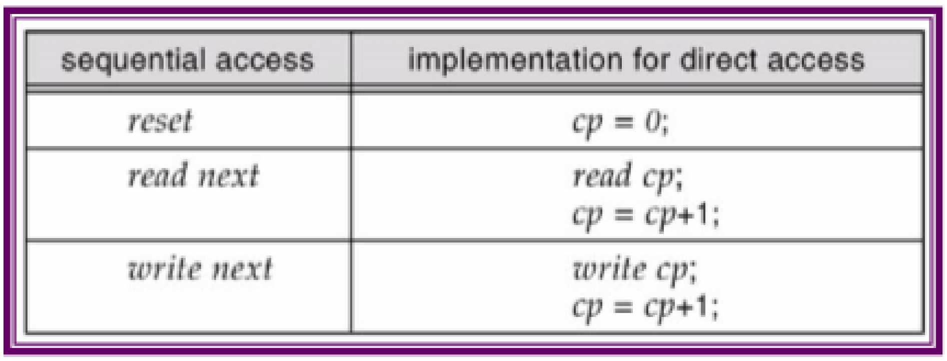
Gambar 9-3 : Akses file berurutan

3.2. Direct Access

The file is a fixed-length logical record that allows the program to read and write records quickly in no particular order. The direct access method is based on the disk model of a file, allowing random to any file block, allowing the random block to be read or written. Operations on direct access consist of :

a. Read n
b. Write n
c. Position to n
    Read next
    Write next
d. Rewrite n

File operations are modified to include block numbers as parameters. Block number specified user which is a relative block number, eg index relative to the beginning of the file. The first relative block of the file is 0, although the actual absolute disk address of the block is for example 17403 for the first block. This method allows the operating system to specify where the file is placed and prevent the user from accessing the position of the file system that is not part of the file. Not all operating systems use either sequential access or direct access to files. Some systems only use sequential access, some other systems use direct access. To change sequential access to direct access is not something difficult as in Figure 9-4.



sequential access	implementation for direct access
reset	*cp = 0;*

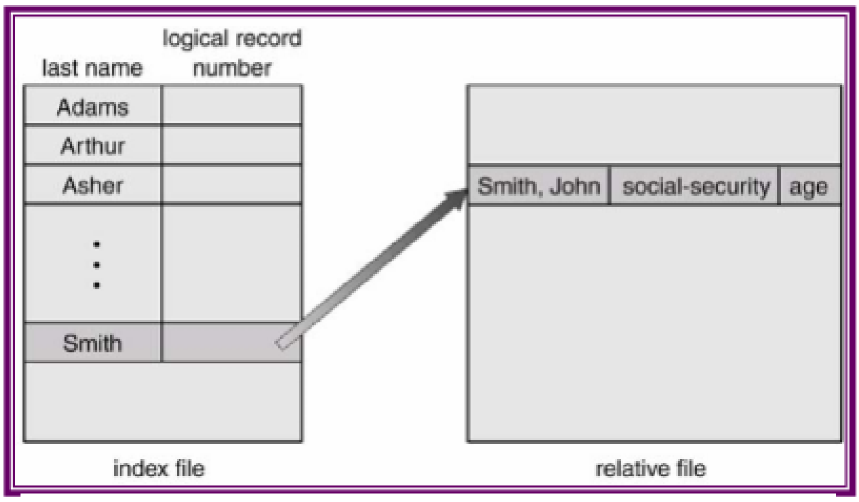
read next	*read cp;*
	cp = cp+1;

write next	*write cp;*
	cp = cp+1;

Gambar 9-4 : Mengubah akses berurutan menjadi akses langsung

3.3. The other method

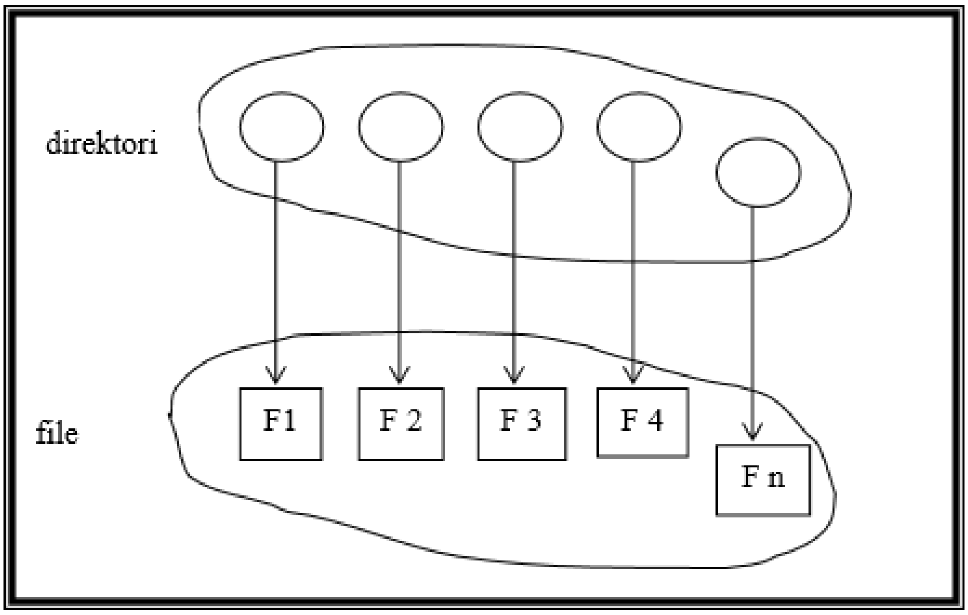
Other access methods can be constructed based on direct access method. This additional method usually involves the construction of an index for a file. The index, like the index at the end of the book, contains a pointer to certain blocks. To specify an input in a file, first searched the index, and then use the pointer to access the file directly and find the appropriate input. Index files can be stored in memory. When the file is large, the index file also becomes too large to be stored in memory. One solution is to create an index for the index file. The primary index file contains a pointer to the secondary index file, which points to the actual item data. An indexing access form is illustrated in Fig. 9-5.



Gambar 9-5 : Contoh indeks dan file relatif

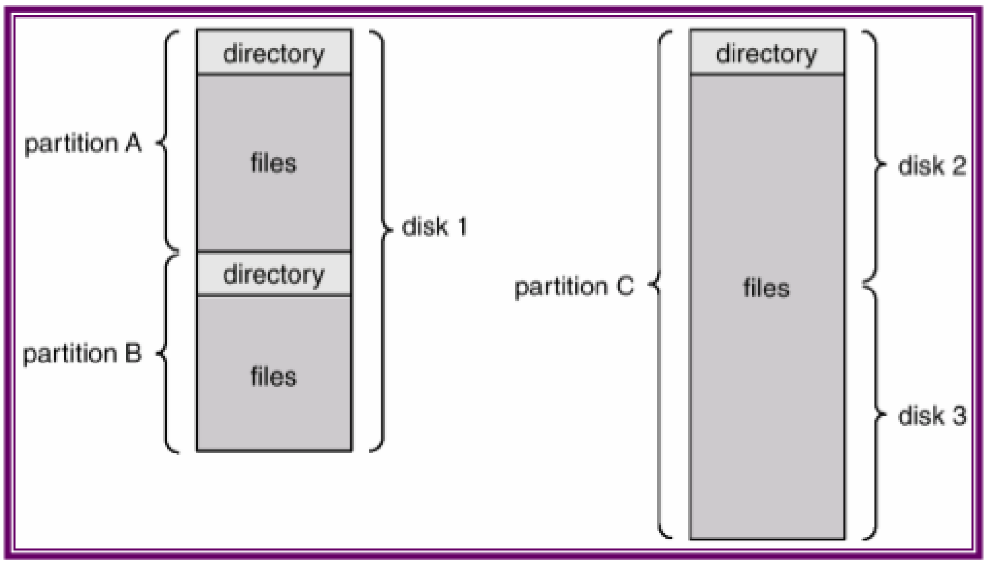
4. Structure of Directory

A directory is a collection of dots that contain information about all the files (Figure 9-6). Some systems store hundreds of files on disks of hundreds of gigabytes. To organize all data using the organization done in two parts.



Gambar 9-5 : Direktori

First, the file system is broken down into partitions, also called "minidisk" (on IBM machines) or "volumes" (on PC and Macintosh machines). Each disk on the system contains at least one partition, a low-level structure in which files and directories reside. Sometimes, partitions are used to define multiple separate areas on a single disk, which are treated as separate storage devices. Other systems use larger partitions of a disk to group disks into a logical structure. Second, each partition contains information about the files in it. This information is stored in an entry in the "device directory or volume table of contents". The directory (or directory) device stores information such as the name, location, size and type for all files of the partition. The common file organization can be seen in Figure 9-7.



Gambar 9-7 : Organisasi sistem file

The information contained in the directory is :

a. Name
b. Type
c. Address
d. Length this time
e. Maximum Length
f. The last date accesses
g. The last data changes
h. ID Owner
i. Protection Information

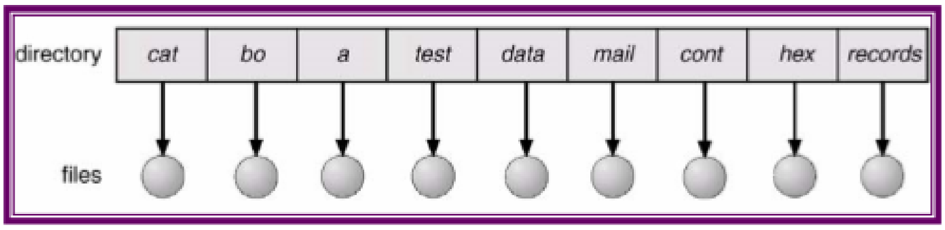
Some operating contained in the directory is :

a. Search
b. Create
c. Delete
d. List
e. Rename
f. Traverse

The suggested file and directory organization is as efficient as possible so it can place files quickly. Also in naming files and directories should be convenient for the user. Two users can provide the same file name. The same file can have multiple names. In the organization of files and directories also need to be grouped files by property, for example all Java programs, games and others.

4.1. Directory one level

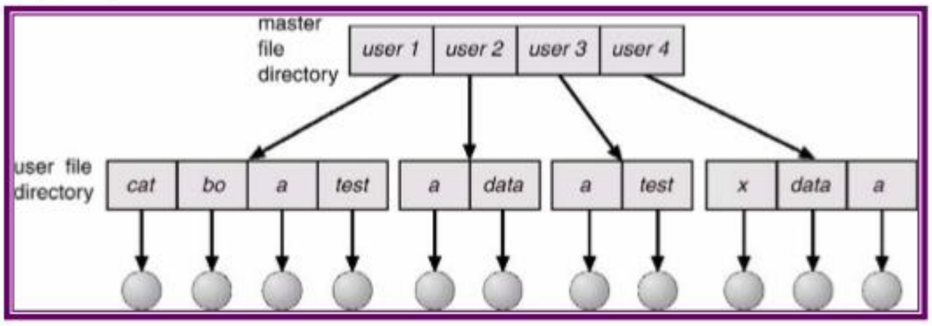
This directory consists of only one directory for each user (Figure 9-8). In this type of directory there is a problem of naming and grouping by user.



Gambar 9-8 : Direktori satu level

4.2. Directory two level

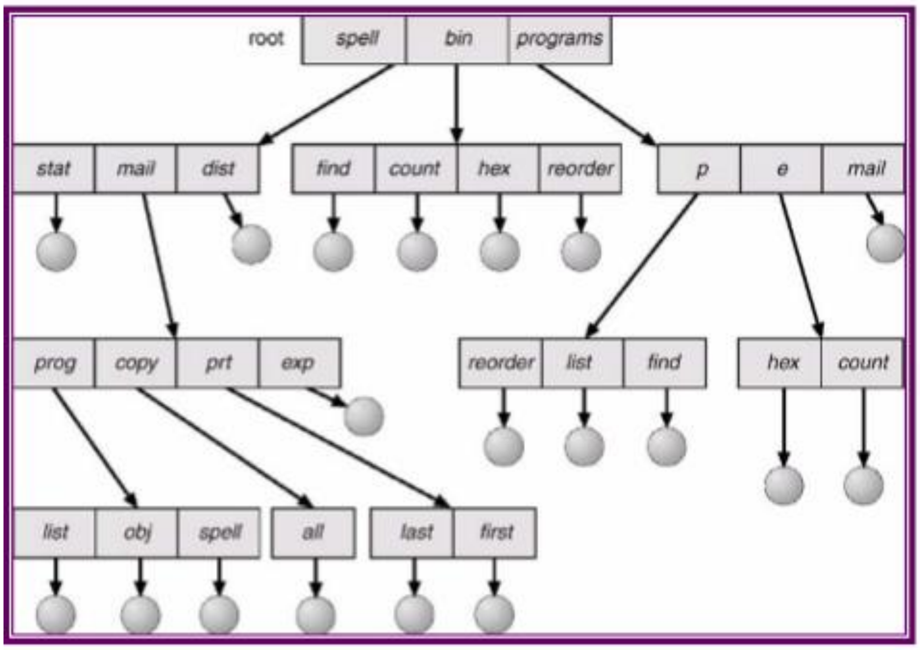
This directory consists of two levels separating the directory for each user (Figure 9-9). Each file named path, can have the same file name for different user, has search capability, but has not done grouping.



Gambar 9-9 : Direktori dua level

4.3. Tree Structured Directory

The tree-structured directory is a commonly used directory structure. The tree has a root directory. Each file on the system has a unique path name (Figure 9-10).

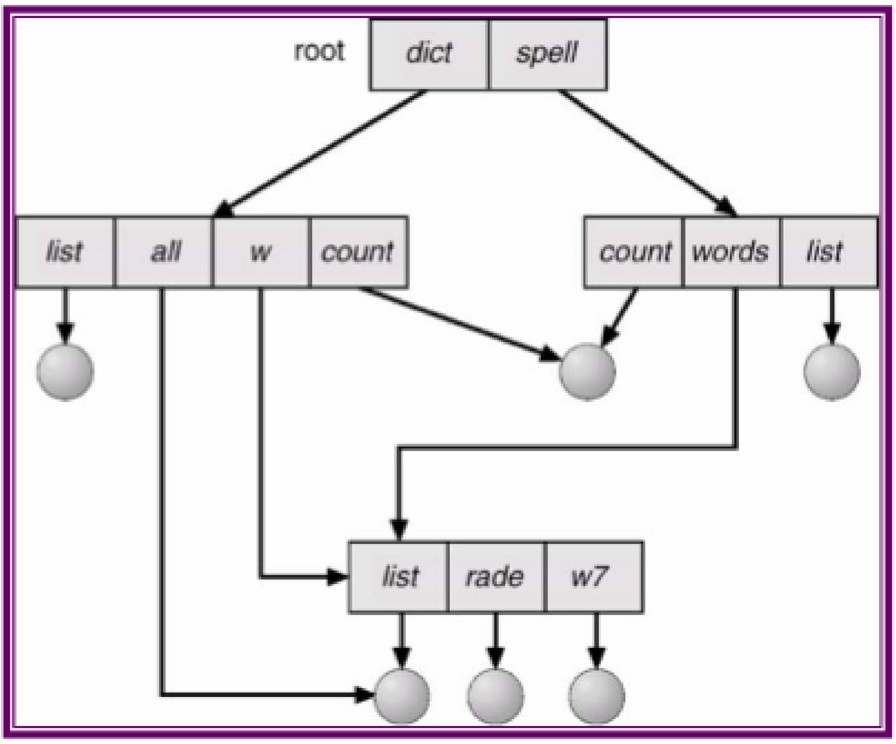


Gambar 9-10 : Direktori berstruktur pohon

In this directory search files and directories more efficiently, grouping files and can access directories and subdirectories. A directory or subdirectory contains a collection of files or subdirectories. A directory is a file that is treated in a special way. All directories have the same internal format. One bit in each directory entry is an input as file (0) or as a subdirectory (1). A special calling system is used to create and delete directories. The path name can be divided into two types namely "absolute" and "relative" pathname. At the moment create a new file will be done in current directory. Similarly when creating a new directory.

4.4. Directory Acyclic Graph

The tree structure prohibits the use of files and directories. In the acyclic graph directory allows directories to have subdirectories and files used together. The same files and subdirectories may reside in two different directories (Figure 9-11).

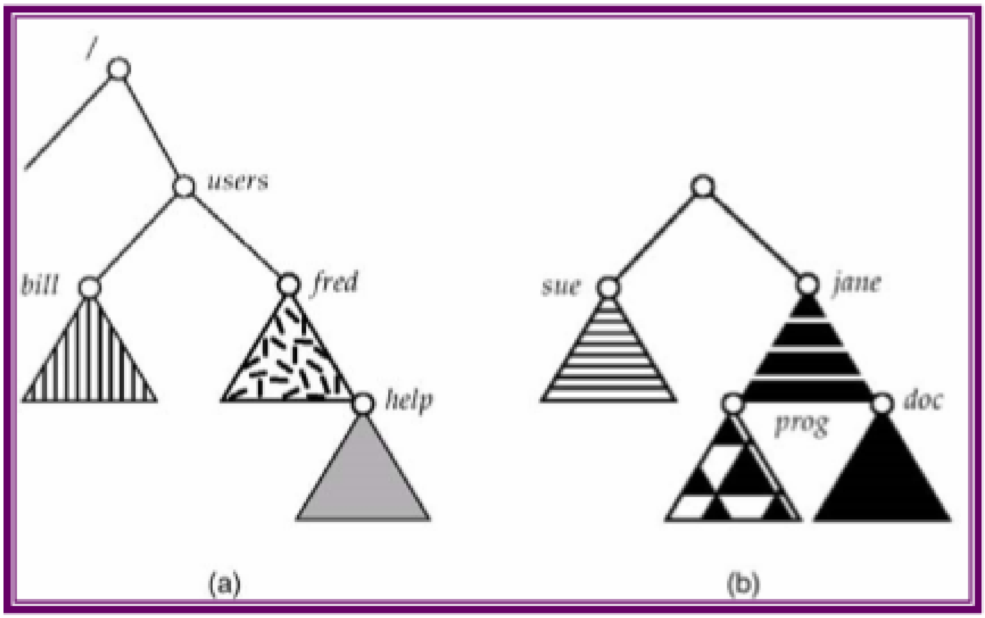


Gambar 9-11 : Direktori acyclic graph

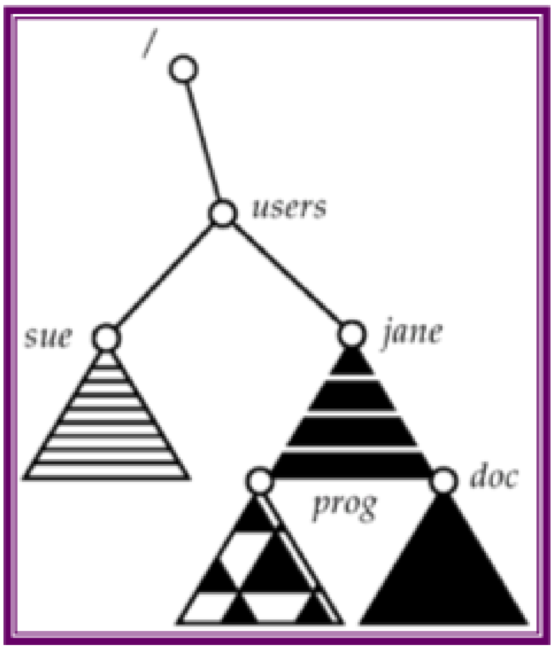
The acyclic graph directory is implemented in several ways. A common way, on some UNIX systems, is to create a new directory entry called a link. A link is a pointer to another file or subdirectory. A link can be implemented as an absolute or relative path name (a symbolic link). A link is different from the original directory. The link is identified by the format in the entry directory and gives the pointer name indirectly. Another emphasis is by duplicating all the information in the shared directory so that both entries are identical and identical. This method causes original information and duplication is indistinguishable The fundamental problem is maintaining consistency if the file is modified. Acyclic-graph directory structure is more flexible than tree structure, but more complex. A file may have more than one path name, consequently, different file names should refer to the same file. If trying to traverse the entire file system (eg for statistical accumulation on all files) the sharing structure should not be crossed more than once. Other issues involve deletion. When the allocated space for shared files can be reallocated and used again. Another approach to deletion is to provide the file until all references are deleted.

5. File System Mounting

A file system must be mounted before it is accessed. Unsold files like Figure 9-12 will be mounted on the mount point (Figure 9-13)



Gambar 9-12: (a) Sistem eksis (b) partisi yang tidak di-mount



Gambar 9-13 : Mount point

6. Protection

Information stored in the computer system must be protected from physical damage (reliability) and improper access (protection). Reliability is usually done by duplicating copies of files. Some computer systems have systems that automatically (or through computer operator intervention) duplicate files to tape regularly from file systems that are suddenly deleted. Protection, by contrast, can be done in several ways.

6.1. Access Type

A protection mechanism with limited file access types that can be created. Access is allowed or not depends on several factors, one of which is access type request. Some operations are provided:

a. Read from file
b. Write to file
c. Execute the file
d. Append contain of file
e. Delete file

Other operations, such as naming, copying or modifying files, must also be controlled. For some reason, higher level functions (such as copying) are implemented by system programs that use lower level call systems. Protection is provided only at a lower level. For example, copying files is implemented with a row of reading requests. In this case a user with read access can cause files to be copied, printed and others.

6.2. Access List and Group

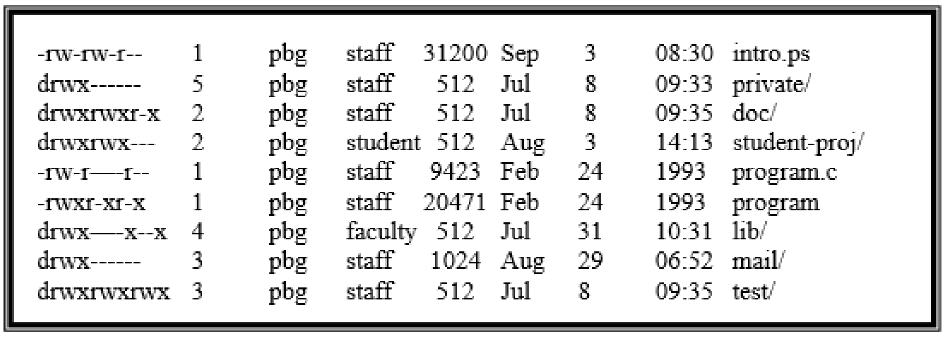
A common approach to protection issues is to make access dependent on user identification. A common scheme for the implementation of identity-dependent access by linking each file and directory with an access list that specifies the user name and the type of access that is allowed for each user. When a user requests access to a specific file, the operating system checks the access list. If the user is registered, access is allowed, otherwise protection violation is prohibited from accessing the file. The main problem with the access list is size. If you want to allow the user to read the file, you must register all users with read access. This technique has two consequences: building a list may be difficult and directory entry which previously has a fixed size is now a varying size, resulting in space management issues. This problem is solved by tightening the access list. Some systems introduce three user classifications :

- Owner : Users who create a file
- Group file : A collection of users who use files together and need the same access
- Universe : All other users in the system.

In order for the above system to work properly, group membership must be strictly controlled. For example, in UNIX systems, groups can be created and modified only by the manager (super user).

6.3. Protection Example : UNIX

On UNIX systems, directory protection is handled the same as file protection, for example, is associated with each subdirektory using owner, group and universe (others) as 3 bits RWX. The information contained in the file from left to right consists of file or directory protection, number of links to file, owner name, group name, file size in bytes, creation date, file name (Figure 9-14).



```
-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    1993   program.c
-rwxr-xr-x    1    pbg   staff    20471  Feb   24    1993   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/
```

Gambar 9-14 : Proteksi file dan direktori pada UNIX

7. References

http://arna.lecturer.pens.ac.id/ (Politeknik Elektronika Negeri Surabaya)