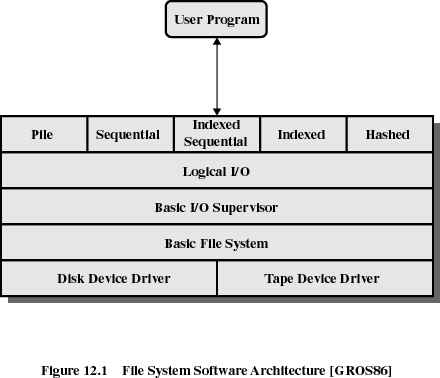
**T0316 – Operating Systems**

Assignment #2

By: Baskoro Indrayana / 1701290311

1. **Describe the file management layer in an operating system.**

**Answer:**



The architecture file management in an operating system can be represented in the form of a stack of layers as the figure above depicts. The lowest layer directly deals with the initiation of and communication within I/O operations on actual the device. Whereas the topmost layer abstracts the I/O operations, in a way that the user can understand.

Following are the layers of the architecture: (from bottom to top)

* *Device driver:* directly communicates with the peripheral devices or their controllers/channels, and is also responsible for initiating the I/O operations on the devices (usually in the form of disk drives);
* *Basic file system:* concerned with the placement of blocks of data in the secondary storage (i.e. the hard disk drive) and buffering of those blocks in the main memory (i.e. the RAM);
* *Basic I/O supervisor:* responsible for all the I/O initiation and termination, scheduling (to optimize performance), and file status;
* *Logical I/O:* enables users and applications to access file records. It provides general-purpose I/O capabilities and maintains basic data about the files; and
* *Access method:* concerned with how the file in the I/O device can be accessed (by the users/applications).

1. **Describe three different types of directory structure.**

**Answer:**

* *List of entries (one entry per file):*

The structure can be represented in the form of a simple, sequential file. The name of the file serves as the key. It is inadequate for use by a shared system of multiple users, each of whom stores a lot of files, and is incapable of structuring files based on the user’s preferences (organize data by name, date, type, etc.).

* *Two-level scheme:*

There are two kinds of directories in this structure:

* + The master directory: stores an entry for each user directory, providing address and control information.
  + The user directory: the actual files/data owned by the user.
* *Hierarchical / tree-structure directory:*

The directories in the computer are stored in the form of a tree. A single master directory becomes the root of the directory tree, containing every user’s directory along with their personal subdirectories and files within.

1. **Describe the following file allocation method**

**Answer:**

* 1. ***Contiguous Allocation***

A single set of blocks is allocated to a file when it is created. Every file has a single entry in the file allocation table. The allocated blocks have their own starting point, and have a length , depending on how large the file is. So if we are to access the -th block of the file (, assuming we use zero-based indices), we have to access the -th block. The disadvantage of this method is on the possibility of external fragmentation in the filesystem.

* 1. ***Chained Allocation***

Instead of directly allocating a set of blocks to a file (within its creation), chained allocation focuses on allocating the file on a basis of individual blocks. Again, the file allocation table requires a single entry for each file, showing the starting point and length of the blocks. In the file, each block has a pointer to the next block, like that of a linked list. This method does solve the external fragmentation issue. However, it does not accommodate the principle of locality (since the blocks can be scattered anywhere in the storage, regardless of which file they belong to).

* 1. ***Indexed Allocation***

Indexed allocation addresses problems of contiguous and chained allocation. In this method, the file allocation table contains a separate one-level index for each file; and the index has one entry for each portion allocated to the file. The file indexes are not physically stored as part of the file allocation table, but in a separate block, so that the file allocation table points to that block in order to refer to the file. The allocation may be on the basis of either fixed-size blocks (eliminates external fragmentation) or variable-size portions (improves locality).

1. **Explain two ways on how to keep track of free blocks in File Management System.**

**Answer:**

* *Bit tables:* stores one bit for each block in the disk. Bit 0 indicates an empty / free block, while bit 1 indicates a block in use. These bits are arranged in the form of a bitmap. It’s easy to find contiguous group of free blocks by using bit tables, and applicable in the file allocation methods mentioned above. However, it is uneconomical in terms of memory usage if the disk size and filesystem block size is too large.
* *Pointers to free space/blocks/portions:* the free portions of the disk can be chained by using a pointer and length value in each free portion. However, this method causes big overhead on the memory space, especially if there are many contiguous free portions spread across the disk. After some use of this method, the disk will become quite fragmented and there will be many single-block-long free pointers within the disk.

1. **Suppose there are two files occupying block as described below:**

File A: Block 5 🡪 block 10 🡪 block 2🡪 block 15

File B: Block 8 🡪 block 9 🡪 block 1 🡪 block 13 🡪 block 4

**Represent the above files using Linked List Tables.**

**Answer:** (in the next page)

**Answer to question #5:**

|  |  |
| --- | --- |
| **Physical block** |  |
| 0 |  |
| 1 | 13 |
| 2 | 15 |
| 3 |  |
| 4 | -1 |
| 5 | 10 |
| 6 |  |
| 7 |  |
| 8 | 9 |
| 9 | 1 |
| 10 | 2 |
| 11 |  |
| 12 |  |
| 13 | 4 |
| 14 |  |
| 15 | -1 |

**End of file B**

**Start of file A**

**Start of file B**

**End of file A**