

Assignment 3

Saad Saiyed
200415258

SAS162@UREGINA.CA

1. What are the advantages of using dynamic loading? (6 marks)

Ans: Only the data which we need for the program should be stored on the memory, in a relocated load format on disk, rather than storing all the data and even the program all at the same time. It would ultimately waste system resources.

To utilize the perfect memory space properly and efficiently, we use Dynamic loading. For dynamic loading to happen, we load the routines from a secondary source, such as a drive or a disk, rather than loading all the routines on the memory. When the routine is needed by the program, the routine is called and then relocated by the source into memory. After that, the memory controls the routine.

Basically, dynamic loading does not require any special operating system support. The programmer designs the program in such a way that it can utilize the library routine provided by the operating system.

2. Explain the basic method for implementing paging. (8 marks)

Ans: Paging is a memory management scheme which permits the physical address of a process to be noncontiguous. An example of this is what happens in segmentation. Moreover, it avoids external fragmentation.

This method will break up the memory into fixed-sized blocks called frames by implementing pages as frames. It also breaks up the logical memory in the blocks of the same size called pages. Due to this arrangement, when we run the process, its pages are loaded into any available frames inside the physical memory. This method is useful as the size of the logical address is not limited by the size of the physical memory.

Overlooking the advantages of this setup, it is actually important that each logical address maps correctly to the right physical address. To get that running, we first look at the address space. It is made up of 2 parts. The first one for the page number and the second one for the page offset. With the right page number of the logical address, one can use the page table to map that very page number to a correct frame number. When we combine that frame number with the offset, it allows us to find the right physical address.

3. Briefly describe the segmentation memory management scheme. How does it differ from the paging memory management scheme in terms of the user's view of memory? (8 marks)

Ans: Segmentation is a way of managing the memory allocation, where the management scheme mimics the user's view of memory. With this process, the logical address space is made up of segments with each of the segments having its base and its length. When we map from the logical to the physical address, we make sure that the offset provided fall within the segment's base address, and even the segment's limit. If this does not happen,

then we would surely figure out that the values written were wrong and the process was illegally trying to access a part of the memory which it was now allowed access to.

4. Explain the distinction between a demand-paging system and a paging system with swapping. (8 marks)

Ans: There are two ways to load process into memory, the first one being demand paging system and the second one being paging system with swapping. The first one works by adding the process content in fragments, just like page by page, and it does not load the entire process at once. The paging system with swapping works in a way where first of all, the system creates a backing store to swap out the process memory in the CPU for adding the original process in the data. Backing store is nothing but a memory which is created just to swap two processes to run simultaneously.

The demand paging system process is more convenient and preferable, as it loads the data piece by piece and not all at once, according to the wish of the user and it is even preferable as we are not swapping the entire process. We actually are swapping processes too in the demand paging system, but the swapping is done in fragments, and not altogether.

5. How does the second-chance algorithm for page replacement differ from the FIFO page replacement algorithm? (8 marks)

Ans: The FIFO replacement algorithm basically means which process we actually have to swap. In FIFO, the process which enters in the very beginning, is swapped with the process which comes when all the space is filled. In second chance algorithm, all the processes are given individual reference bits and initially their value is 0. When the process is running, the reference bit value changes to 1. When all the processes have the value of 1 in their reference bit, the process with value 0 becomes the victim. Basically it tries to remove the least recently used frame. But this process requires hardware which only a few computers can support.

6. Explain how copy-on-write operates. (8 marks)

Ans: There are two types of related processes, one is the parent process and the other is the child process. Eventually, the child process does not utilize all the pages from its parent process, rather it makes a copy or say a duplicate of its parent's necessary pages when it needs to write or make changes to it. So this process of copying the actual page for writing into it is called copy to-write.

The pages which are utilized by the child process by this way basically should have two characteristics. First, it should be modifiable, because the pages which cannot be modified cannot be shared. Second, all the pages must have a tag of "copy to-write" with them, for them to be edited further.

7. If you were creating an operating system to handle files, what are the six basic file operations that you should implement? (8 marks)

Ans: The six basic file operations used to handle files in operating systems are as follows. Creating, writing, reading, deleting, truncating and re-positioning of files. Creating a file requires both finding a space in a file system, and it should be allowed access in the file directory using the name and attribute of the file. In order to write a file, we need the file

name and the data that is meant to be written. However, we need the writer pointer in order to keep track of where we actually have to begin after writing.

Reading requires knowing the file name as well the block of the memory from which file is being read. Exactly as in writing, we use a read pointer to keep track of the next beginning. Instead of using one read pointer and one write pointer, we can use one pointer for determining the beginning of the next, and it will be called current file position pointer. Unlike reading and writing files, we just use the name of the file in order to delete it, and it allocates a free space for further use in the file structure. When we just want to delete the content of the file and not the actual file, we use truncating.

To change the position of the current file position pointer, we just call the name of the file and the value of the address where we actually want to re-position to.

8. To create a new file, an application program calls on the logical file system. Describe the steps the logical file system takes to create a file. (8 marks)

Ans: When we create a new file, the file system creates an FCB on the place in the file system where the actual file has to be created. We can create the FCB as per the file system's convenience, initially or whenever it is necessary to create it. For creating a file, of course, the name is necessary. When it is created, the file name as well as the attributes are necessary to be introduced in the file directory.

9. How is a hash table superior to a simple linear list structure? What issue must be handled by hash table implementation? (8 marks)

Ans: The simple linear list structure is user friendly and easy to implement, moreover, it has a natural appeal. The only consequence we have to face during its usage is that it is a little bit slow during its runtime, which causes noticeable lagging. To explain the slow usage, we take the example that even for adding a file, creating a file or deleting a file, we have to search the query in the linear list structure, and the reason for the lag is the way the data is stored. The actual behind the scenes happen in the cache, which helps in the search queries to store recently used files, for them to be restored easily.

A hash table is an algorithm which stores data at the location related to its value. It narrows down the query of searching. This helps in reducing the search time hence it is more preferable. It converts the file names into hash values, each of them having a pointer assigned to a file name. The only problem which is prevalent in this algorithm is that which occurs when two file names are the same.

10. What are the factors influencing the selection of a disk-scheduling algorithm? (8 marks) Ans: The disk-scheduling algorithm is based on two parameters, one being the number of requests and the next being the type of requests. If there is only one request, then all the algorithms will be identically efficient. When the number of requests increase, the movement should be managed so as to be as efficient as possible. The types of requests determine the efficiency of that very request.

When the data is stored adjacent to each other, it reduces the amount of the disk head movement pointer.

11. Explain the disadvantage(s) of the SSTF scheduling algorithm. (8 marks)

Ans: A scheduling algorithm is something which schedules the next process in the queue efficiently. The SSTF stands for Shortest Seek Time First algorithm. It relies on priority, given to the process which has the shortest seek time. In order to explain this, we have an example. The read head is at 90. The processes have values 120, 100 and 94. So, the algorithm will obviously choose 94 rather than the first value 120, because this selected process has the lowest seek time. After that, it will choose the next lowest seek time, which is 100. Then the last priority will be given to the value having the seek time of 120. This is how the priority system SSTF for processes work.

Still, there are some flaws in this algorithm. The first one is starvation of requests. The priority in the processes of this algorithm does not give priority to higher seek times even though they arrive first, so, if there is a process of an extreme high seek time in one set of processes, it will be delayed, because the next set of processes will arrive, and they will obviously have reduced seek time. For example, the read head is at 90. The seek time for one set of processes are 94, 99 and 213. The processes having seek times 94 and 99 will be selected by this algorithm as priority, but before that process of seek time 213 is selected, the algorithm is introduced with a different set of processes, this time with 95, 102 and 110. So even before the process having time 213 is selected, the process having time 95 and 102, and even 110 will be selected.

12. Explain the concepts of a bus and a daisy chain. Indicate how these concepts are related. (8 marks)

Ans: The bus network refers to devices such as CDs, Rom, scanners, mouse or keyboards, even printers, connected to the same set of wires. A bus can determine how instructions are transferred. They are of different kinds, one of them includes PCI and the other includes expansion of the bus. The PCI is used for fast device connections and the latter is used for slow device connections.

Discussing daisy chain, we can say that even in this connection, the devices are connected to one another, wired or wireless, but the tail of these devices is connected to the computer through a port. Scalability is an exclusive feature of daisy chain. We can add loads of devices to this chain system, like a linked list.

13. What are the three reasons that buffering is performed? (6 marks)

Ans: Buffer is actually playing the role of a middle man, between data transfer of two devices to facilitate the transfer. The three main reasons are described as follows. 1) Difference in speed of sending device and receiving device. When the sending device is transferring data slower than the receiving device, the data is stored in the buffer, and when the data is finished sending, the buffer sends the data altogether to the receiving end. 2) Difference of data transfer size in both the sending and receiving devices. There is rarely a chance where both sender and receiver has same transfer size and that way buffer is used to send or collect data in buckets where all the data is gathered in a fashion to produce a complete effect. 3) A buffer can be utilized for efficient data integrity. It is useful as all the data transferred by buffer is stored in system kernel, one of the safest place to keep data.

14. Why is it important to distinguish between mechanisms of protection and policies of protection? (4 marks) Ans:

15. What is an access matrix, and how can it be implemented? (4 marks) Ans:
16. How does a virus differ from a worm? (4 marks) Ans:
17. What is the difference between symmetric encryption and asymmetric encryption?
(4 marks) Ans:
18. What are the two main varieties of authentication algorithms? (4 marks) Ans: