Technical Papers UNIX Concepts: Unix Memory Management

1. What is the difference between Swapping and Paging? Swapping: Whole process is moved from the swap device to the main memory for execution. Process size must be less than or equal to the available main memory. It is easier to implementation and overhead to the system. Swapping systems does not handle the memory more flexibly as compared to the paging systems. Paging: Only the required memory pages are moved to main memory from the swap device for execution. Process size does not matter. Gives the concept of the virtual memory. It provides greater flexibility in mapping the virtual address space into the physical memory of the machine. Allows more number of processes to fit in the main memory simultaneously. Allows the greater process size than the available physical memory. Demand paging systems handle the memory more flexibly.

2. What is major difference between the Historic Unix and the new BSD release of Unix System V in terms of Memory Management? Historic Unix uses Swapping entire process is transferred to the main memory from the swap device, whereas the Unix System V uses Demand Paging only the part of the process is moved to the main memory. Historic Unix uses one Swap Device and Unix System V allow multiple Swap Devices.

3. What is the main goal of the Memory Management? It decides which process should reside in the main memory, Manages the parts of the virtual address space of a process which is non-core resident, Monitors the available main memory and periodically write the processes into the swap device to provide more processes fit in the main memory simultaneously.

4. What is a Map? A Map is an Array, which contains the addresses of the free space in the swap device that are allocatable resources, and the number of the resource units available there. This allows First-Fit allocation of contiguous blocks of a resource. Initially the Map contains one entry address (block offset from the starting of the swap area) and the total number of resources. Kernel treats each unit of Map as a group of disk blocks. On the allocation and freeing of the resources Kernel updates the Map for accurate information.

5. What scheme does the Kernel in Unix System V follow while choosing a swap device among the multiple swap devices? Kernel follows Round Robin scheme choosing a swap device among the multiple swap devices in Unix System V.

6. What is a Region? A Region is a continuous area of a process's address space (such as text, data and stack). The kernel in a Region Table that is local to the process maintains region. Regions are sharable among the process.

7. What are the events done by the Kernel after a process is being swapped out from the main memory? When Kernel swaps the process out of the primary memory, it performs the following: Kernel decrements the Reference Count of each region of the process. If the reference count becomes zero, swaps the region out of the main memory, Kernel allocates the space for the swapping process in the swap device, Kernel locks the other swapping process while the current swapping operation is going on, The Kernel saves the swap address of the region in the region table.

8. Is the Process before and after the swap are the same? Give reason. Process before swapping is residing in the primary memory in its original form. The regions (text, data and stack) may not be occupied fully by the process, there may be few empty slots in any of the regions and while swapping Kernel do not bother about the empty slots while swapping the process out. After swapping the process resides in the swap (secondary memory) device. The regions swapped out will be present but only the occupied region slots but not the empty slots that were present before assigning. While swapping the process once again into the main memory, the Kernel referring to the Process Memory Map, it assigns the main memory accordingly taking care of the empty slots in the regions.

9. What do you mean by u-area (user area) or u-block? This contains the private data that is manipulated only by the Kernel. This is local to the Process, i.e., each process is allocated a u-area.

10. What are the entities that are swapped out of the main memory while swapping the process out of the main memory? All memory space occupied by the process, process's u-area, and Kernel stack are swapped out, theoretically. Practically, if the process's u-area contains the Address Translation Tables for the process then Kernel implementations do not swap the u-area.

11. What is Fork swap? fork () is a system call to create a child process. When the parent process calls fork () system call, the child process is created and if there is short of memory then the child process is sent to the read-to-run state in the swap device, and return to the user state without swapping the parent process. When the memory will be available the child process will be swapped into the main memory.

12. What is Expansion swap? At the time when any process requires more memory than it is currently allocated, the Kernel performs Expansion swap. To do this Kernel reserves enough space in the swap device. Then the address translation mapping is adjusted for the new virtual address space but the physical memory is not allocated. At last Kernel swaps the process into the assigned space in the swap device. Later when the Kernel swaps the process into the main memory this assigns memory according to the new address translation mapping.

13. How the Swapper works? The swapper is the only process that swaps the processes. The Swapper operates only in the Kernel mode and it does not uses System calls instead it uses internal Kernel functions for swapping. It is the archetype of all kernel process.

14. What are the processes that are not bothered by the swapper? Give Reason. Zombie process: They do not take any up physical memory. Processes locked in memories that are updating the region of the process. Kernel swaps only the sleeping processes rather than the ready-to-run'processes, as they have the higher probability of being scheduled than the Sleeping processes.

15. What are the requirements for a swapper to work? The swapper works on the highest scheduling priority. Firstly it will look for any sleeping process, if not found then it will look for the ready-to-run process for swapping. But the major requirement for the swapper to work the ready-to-run process must be core-resident for at least 2 seconds before swapping out. And for swapping in the process must have been resided in the swap device for at least 2 seconds. If the requirement is not satisfied then the swapper will go into the wait state on that event and it is awaken once in a second by the Kernel.

16. What are the criteria for choosing a process for swapping into memory from the swap device? The resident time of the processes in the swap device, the priority of the processes and the amount of time the processes had been swapped out:

17. What are the criteria for choosing a process for swapping out of the memory to the swap device? The process's memory resident time, Priority of the process and The nice value.

18. What do you mean by nice value? Nice value is the value that controls { increments or decrements} the priority of the process. This value that is returned by the nice () system call. The equation for using nice value is: Priority = (recent CPU usage/constant) + (base-priority) + (nice value) Only the administrator can supply the nice value. The nice () system call works for the running process only. Nice value of one process cannot affect the nice value of the other process.

19. What are conditions on which deadlock can occur while swapping the processes? All processes in the main memory are asleep. All ready-to-run'processes are swapped out. There is no space in the swap device for the new incoming process that are swapped out of the main memory. There is no space in the main memory for the new incoming process.

20. What are conditions for a machine to support Demand Paging? Memory architecture must based on Pages, The machine must support the restartable instructions.

21. What is the principle of locality? It's the nature of the processes that they refer only to the small subset of the total data space of the process. Ie, the process frequently calls the same subroutines or executes the loop instructions.

22. What is the working set of a process? The set of pages that are referred by the process in the last n, references, where n is called the window of the working set of the process.

23. What is the window of the working set of a process? The window of the working set of a process is the total number in which the process had referred the set of pages in the working set of the process.

24. What is called a page fault? Page fault is referred to the situation when the process addresses a page in the working set of the process but the process fails to locate the page in the working set. And on a page fault the kernel updates the working set by reading the page from the secondary device.

25. What are data structures that are used for Demand Paging?

Kernel contains 4 data structures for Demand paging. They are:

Page table entries.

Disk block descriptors.

Page frame data table (pfdata).

Swap-use table.