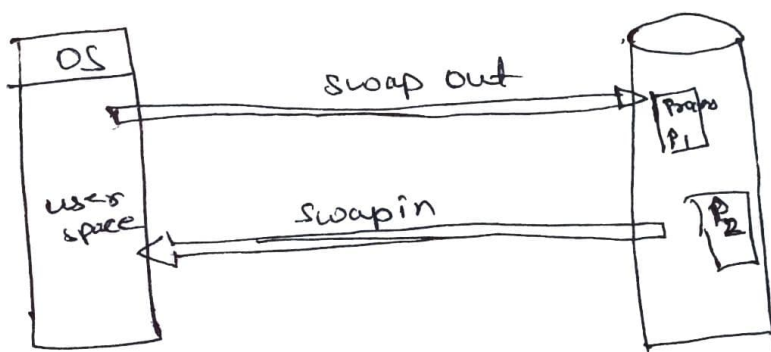


Q1) Solⁿ Swapping

- A process must be loaded into memory to execute.
- If there is not enough memory available to keep alternating process in memory at the same time the same process who are not using the CPU may have their memory swapping out to a first local disk called backing store.



This is a Technique of temporarily removing a process from the memory of a Computer System.

Q2) Solⁿ Caches are useful when two or more component need to exchange data and the component perform transfers at differing speeds caches solve the transfer problem by providing a buffer of intermediate speed between the component. If the first device finds the data it needs in the cache it need not wait for the slower device. The data cache must be kept consistent with the data in the components. If a component has a data value change and the datum is also in the caches, the cache must be

updated. This is specially a problem on multiprocessor system where more than one process may be eliminated by an equal sized cache but only if

i) The cache and the component have equivalent state capacity.

ii) The cache is affordable because faster storage tends to be more expensive

Solution (3) Yes it is possible to develop a new Command interpreter using the system call interface on those OS where the interpreter is not tightly integrated into the system.

Solution (4) Monolithic Kernel - All parts of a kernel like the Scheduler, file system, memory management, Networking stacks, Device Drivers etc, are maintained in one unit within the kernel in monolithic kernel.

→ It is faster processing.

→ Crash Insecure porting Inflexibility.

Kernel size explosion.

Ex:- MS-DOS, UNIX, LINUX.

Micro Kernel :- Only the very important parts like IPC kernel process (communication) basis scheduler basic memory handling, basis I/O primitive etc. are put into the kernel communication happen via message passing other are maintained as server process in user space

- Crash Resistant, Portable, smaller size
- slower processing due to additional message passing

Ex windows NT

⑤ Every Program assigned a priority.
The CPU is always allocated to the highest priority program that wishes to use it.

- A low priority program executing on the CPU preempted if a higher-priority program wishes to use the CPU.

CPU always executing the highest priority program that needs it.

The high priority program completed the kernel would immediately switch the CPU to the high priority program.

