

***Assignment***

***CIA -3***

**(Operating Systems)**

**Date: 26 / 09 / 2022**

**Submitted To: Submitted By:**

*Dr. Senthilnathan T Nandini*

*Associate Professor 1 - MCA A (Department of Computer Science) 2247153*

***CIA - 3***

System program and Process management

1. Write a brief Note About “ Virtual Memory Management”.

2. Explain the concept of "Demand Paging".

3. "Demand Paging can significantly affect the performance of a computer System". Explain .

1. ***VIRTUAL MEMORY MANAGEMENT***

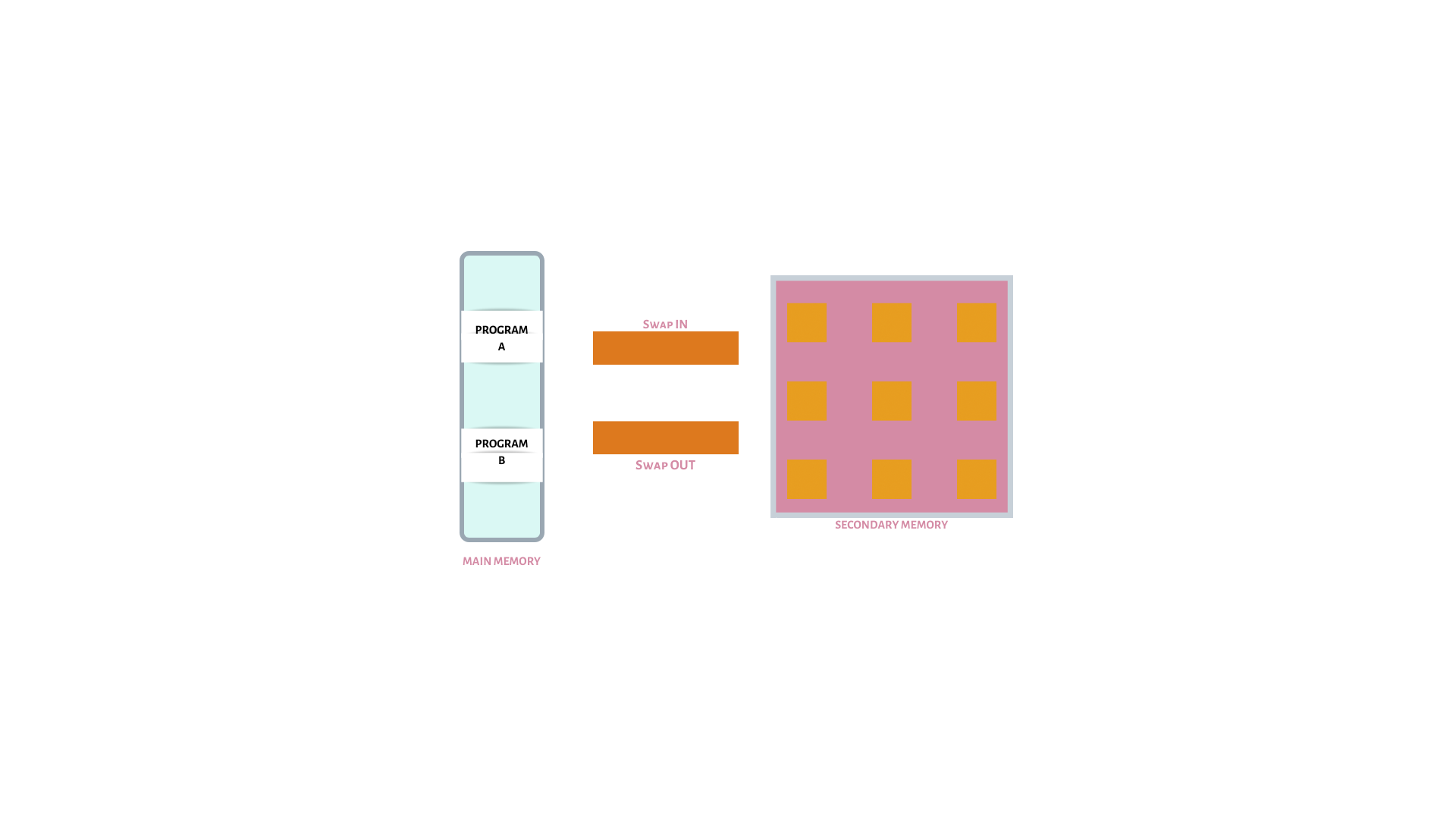
* Virtual memory is a technique for memory management where the secondary memory is made available for use as if it were a part of our system’s main memory.
* Virtual memory itself is a management technique which allows the execution of the processes that are not completely present in the main memory.
* We need virtual memory as it helps in improving the system performance,supports multitasking along with the usage of large programs.
* Basically whenever there are any memory shortages in the computer system, the virtual memory uses both, the hardware and the software, to cope up with that.
* It transfers the data from the Random Access Memory(RAM) to the disk storage , on a temporary basis.
* With the help of virtual memory, a process becomes capable of creating a region of memory that can be shared with another processes.
* But how is the virtual memory implement ?

It is implemented using two basic techniques:

1. Demand Paging 2. Demand Segmentation

* Demand paging refers to a technique where the pages are loaded into the main memory from the logical address space only when demanded by the user.
* A logical address space refers to the logical view of how a process is stored in the memory.
* Demand segmentation is just demand paging replaced with segments.

In this, the process is divided into smaller segments which are thereafter loaded into the main memory only when demanded by the user. When the user demands each part/segment is allocated to a process.



Here we completely don’t swap the entire process, rather we use a lazy swapper that swaps the page to the memory only when it is needed.

EXAMPLE :

A student who has multiple assignments to do might try to switch between two processes - google docx and browser. Now to make these programs run efficiently , the computer might need to adjust its memory usage. The system will start searching for a page table containing the virtual address for the process that will run the program. After it is located, the operating system can move the application to RAM to open it.

1. ***DEMAND PAGING***

* When a program is to be executed, we need to load it into the memory from the disk. One way of doing this is by loading the entire program at the time of execution.
* But what if the user has options to choose from and he doesn’t go with this one?

Thus to avoid that from happening we use a strategy known as demand paging which allows us to load only the pages that are necessary/ we can say are required by the user.

* With this technique, the pages that are already present there in the virtual memory are loaded into the main memory only when they are demanded at the time of execution.
* But sometimes it may undergo a page fault.

Page Fault is an exception made by the system when there is an issue with the loading of the page/ in execution when a user demands it.

* When the cpu makes an attempt to get the access to the page that has not yet been loaded/ made available , then it gives an interrupt. That interrupt is the memory access fault.
* The process so interrupted is put to a blocking state and is not executed until the operating system has loaded the required page into the memory.
* The page demanded is thus brought from the logical address space to the physical memory and made available to the user.

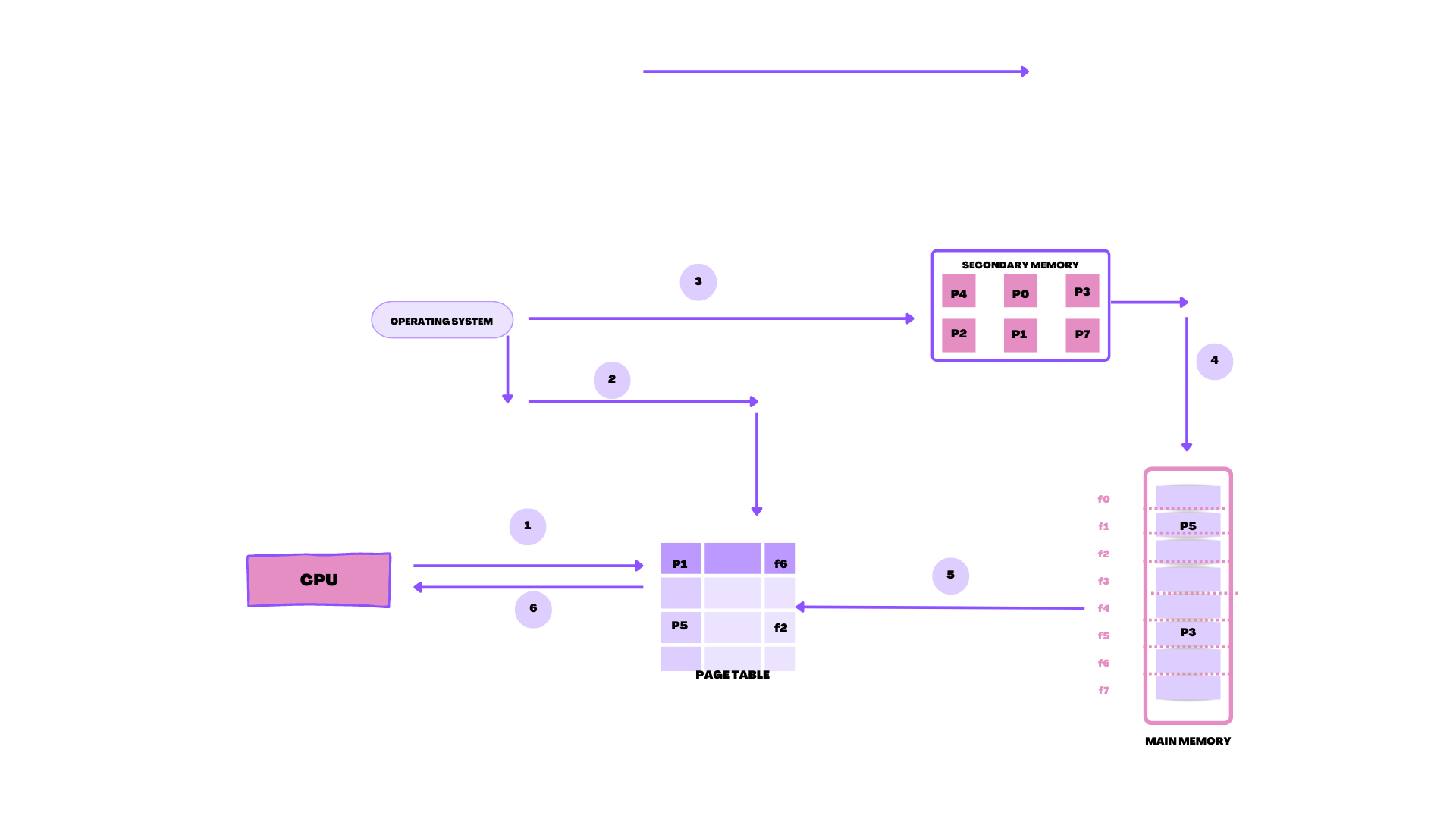
ADVANTAGES OF DEMAND PAGING:

* The very first one is that it allows us to load only those pages into the main memory that are required by the process in execution.
* Now since there is more space available in the main memory, more processes can be loaded into it. This helps us to minimize resource utilization and reduction in context switching.
* It is cost effective since main memory is more expensive than secondary memory.
* Easy to swap pages.
* Easy to share all pages.

DISADVANTAGES OF DEMAND PAGING :

* Takes more time for memory access.
* Along with memory and registers we need a page map table also.
* It requires the kernel to keep a track of all the programs that are being used by applications and which are not, thus this can sometimes make the system work a little bit slower than usual.

EXAMPLE :



* Imagine the CPU is being instructed to fetch the page P4 of an ongoing process. The CPU will go and search for it in the page table.
* If the page table doesn’t have the required page we will get a page fault.
* The operating system will claim the process to a blocking state .
* Now the OS will look for it in the secondary memory.
* Once the page P4 is found, it will be loaded into the main memory so that the user can access it.

1. ***“ Demand paging can significantly affect the performance of the system”:***

* The demand paging functioning is somewhat related to memory access time and effective access time. The memory access time which is denoted by : ma, ranges from 10 to 200 nanoseconds.
* When the whole process is going smoothly that is there is no page fault, the memory access time is almost equal to effective access time.
* Let,

Memory access time = ma;

Probability of a page fault = p; (0<p<1)

Thus, the effective access time = ( 1-p )\*ma + p\*page\_fault\_time

* The three major components of page-fault service time are:
* Service the page-fault interrupt
* Read in the page
* Restart the process.
* As we know that page switch time taken by the system is almost approximately equal to 8 milliseconds.
* Thus we can use the effective access time formula to draw the conclusions:

If,

Memory access time =200 ns

Total paging time= 8ms

Effective access time = (1-p)\*200 + p\*(8)

= 200+7999800 \* p

* Effective access time - proportional to- probability of the page fault.
* If out of 1000 page faults only one can actually cause it, then the effective access time is 8.2 microseconds.
* Because of the demand paging the factor will be 40 and thus slows down our computer.
* As the effective access time increases the process execution becomes slower hence we need to keep the page fault rate low in a demand paging system.