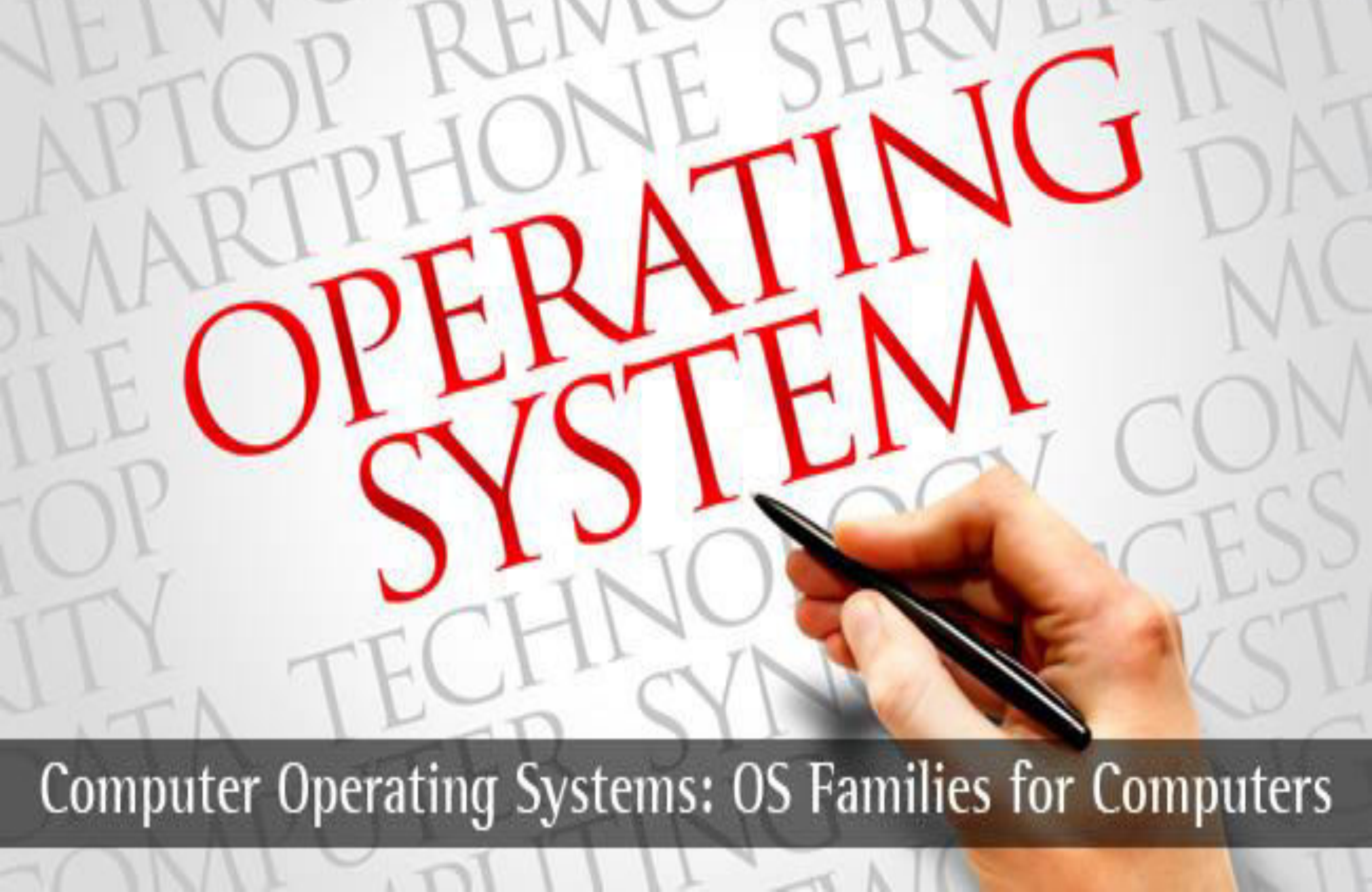
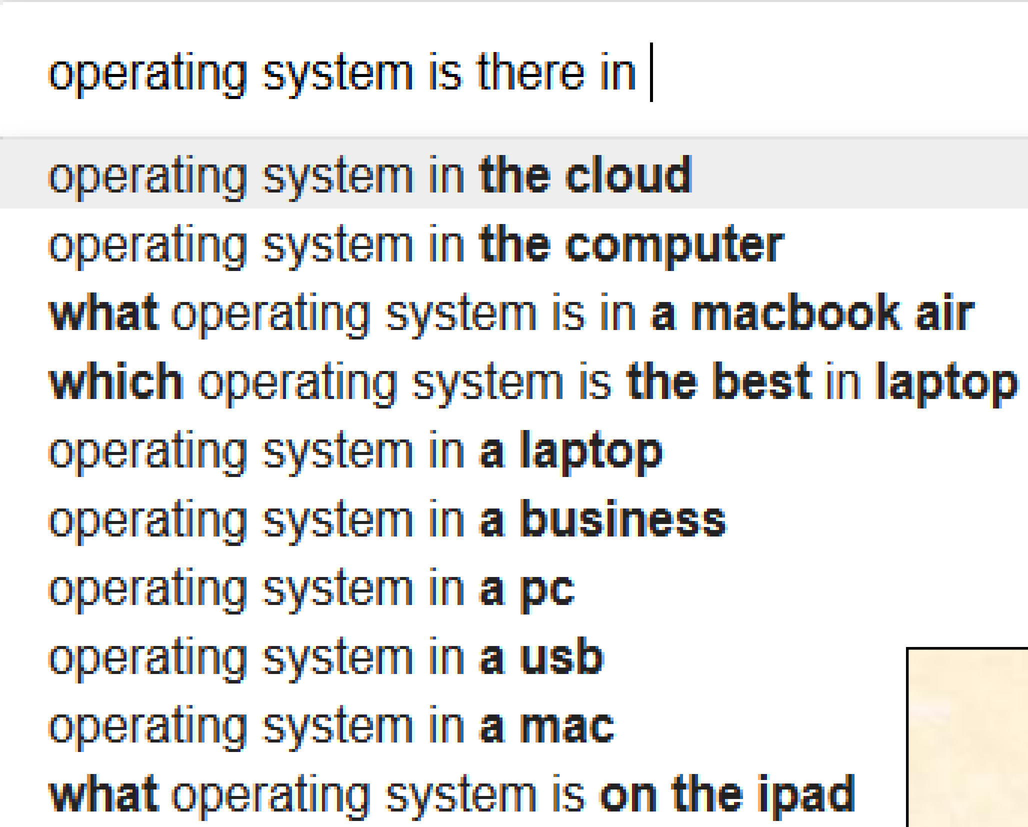
CS-3103 : Operating Systems : Sec-A (NB) : Introduction



Set of Operating Systems Running…..

What is an OS we carry with us every day (may be 24/7) ?



operating system is there in

operating system in **the cloud**

operating system in **the computer**

what operating system is in **a macbook air**

which operating system is **the best** in **laptop**

operating system in **a laptop**

operating system in **a business**

operating system in **a pc**

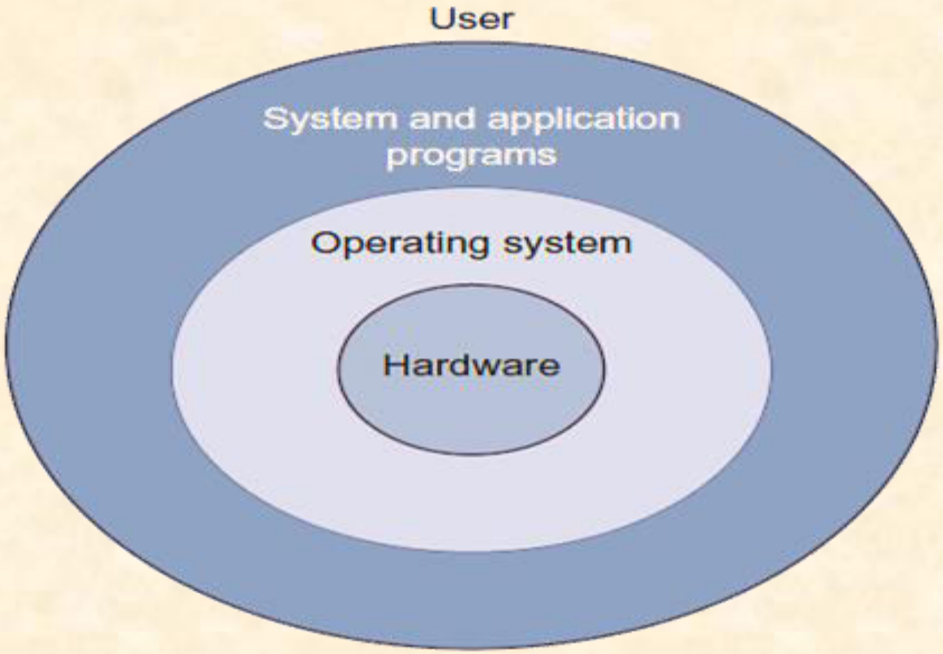
operating system in **a usb**

operating system in **a mac**

what operating system is **on the ipad**

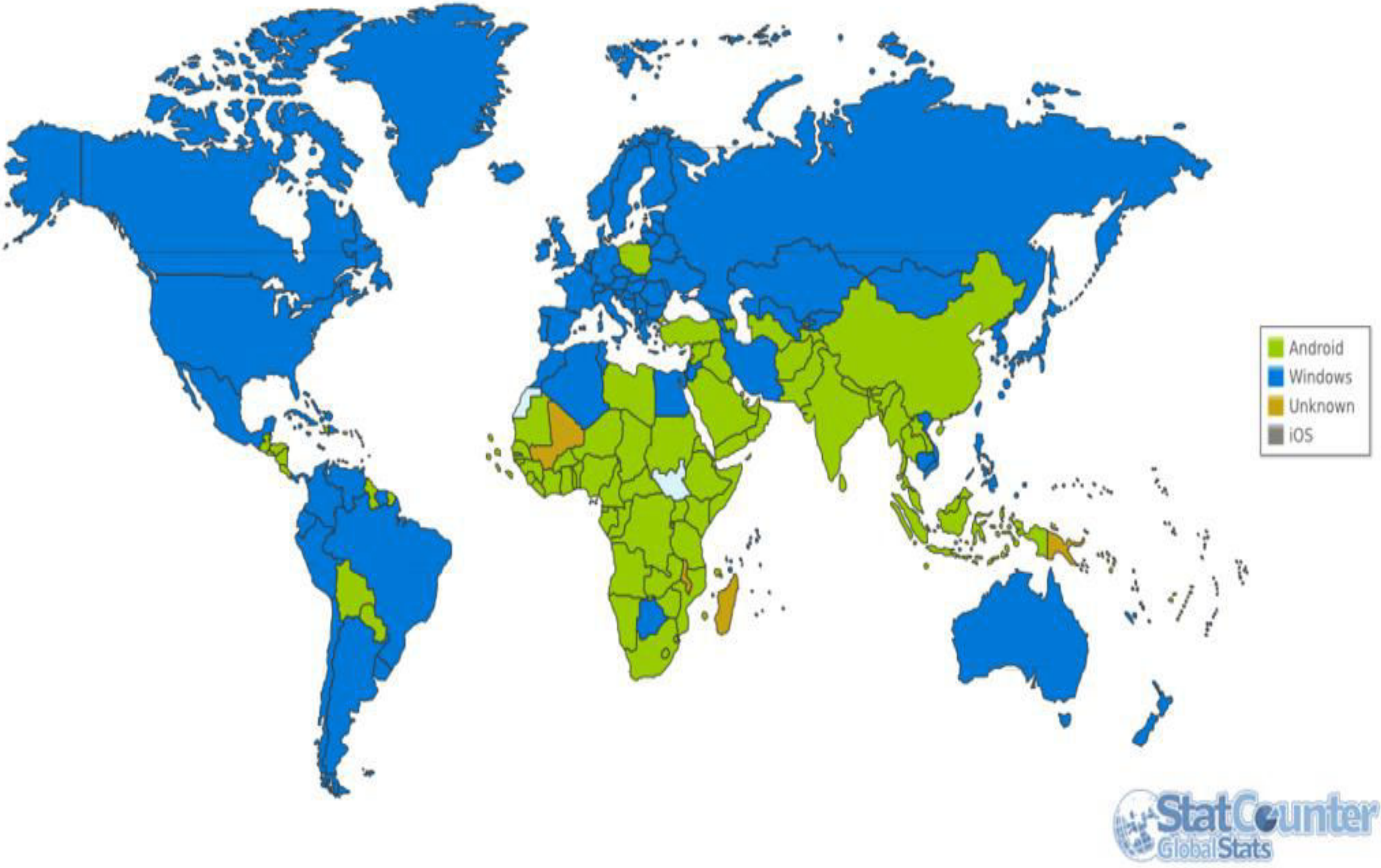
Presents an environment for the user so that he or she can easily work without worrying about the hardware resources.

Manages all the resources in an efficient manner.



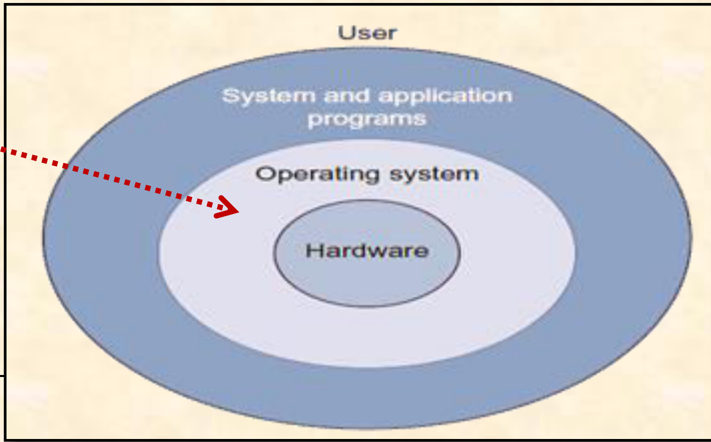
StatCounter Global Stats

Operating System Market Share Worldwide, Apr 2017



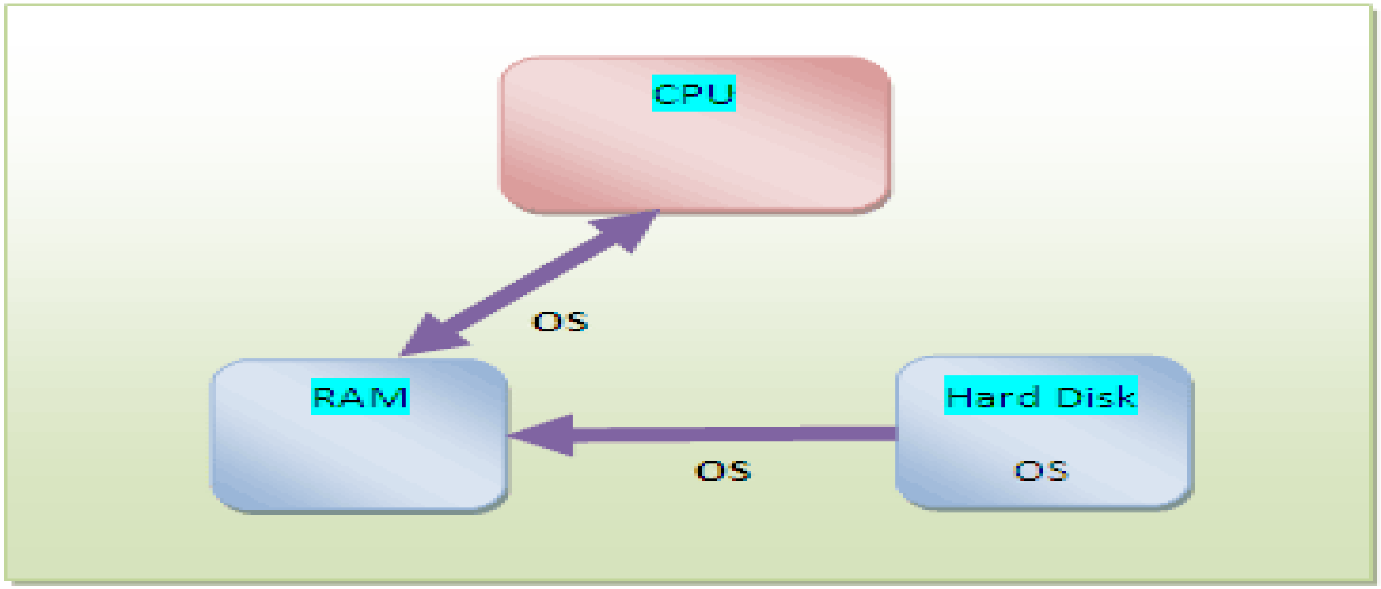
What is an OS we carry with us every day (may be 24/7) ?

- When a computer is turned on, the **operating system (OS)** is loaded into its main memory and it takes control of the system.
- It provides an interface to the user for interaction.
- The user does not work with the actual machine but with a *virtual machine* that is offered by the OS.
- ***Set of programs that acts as an interface between the user and the machine.***
- ***It manages and allocates the resources available in the computer system, thus, OS can be considered as a resource manager.***
- Examples: Windows, Mac OS, Unix, Linux, etc.



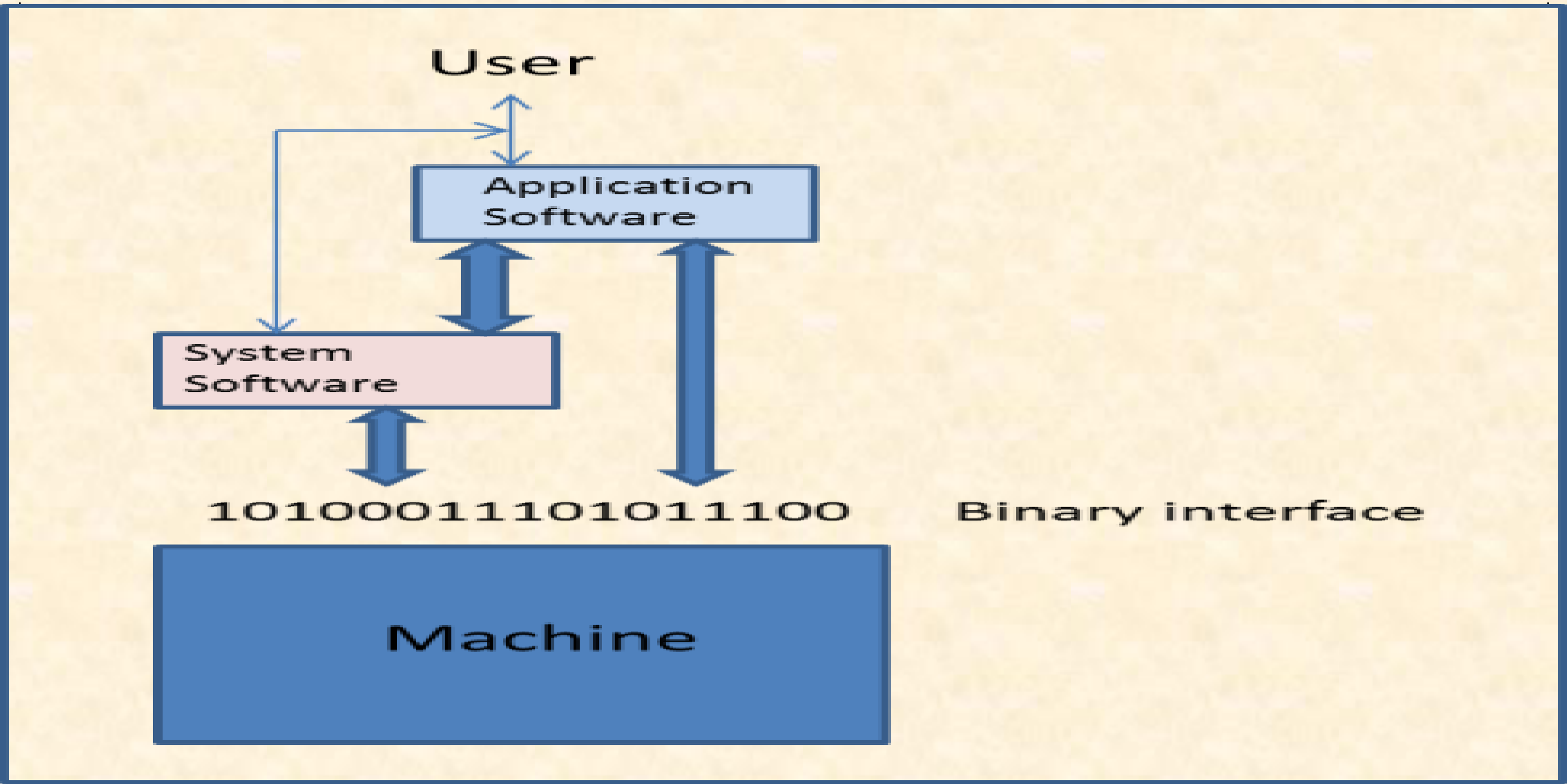
Objectives of an Operating System?

- User-friendly
- Interprets commands
- Creates an environment for other programs to execute on the CPU
- Allocates resources to the executing program as and when required
- Harnesses the full power of the machine



Computer System Components : Categories of Software

- Software can be divided into two major categories:
  - – Application software
  - – System software



Application Software

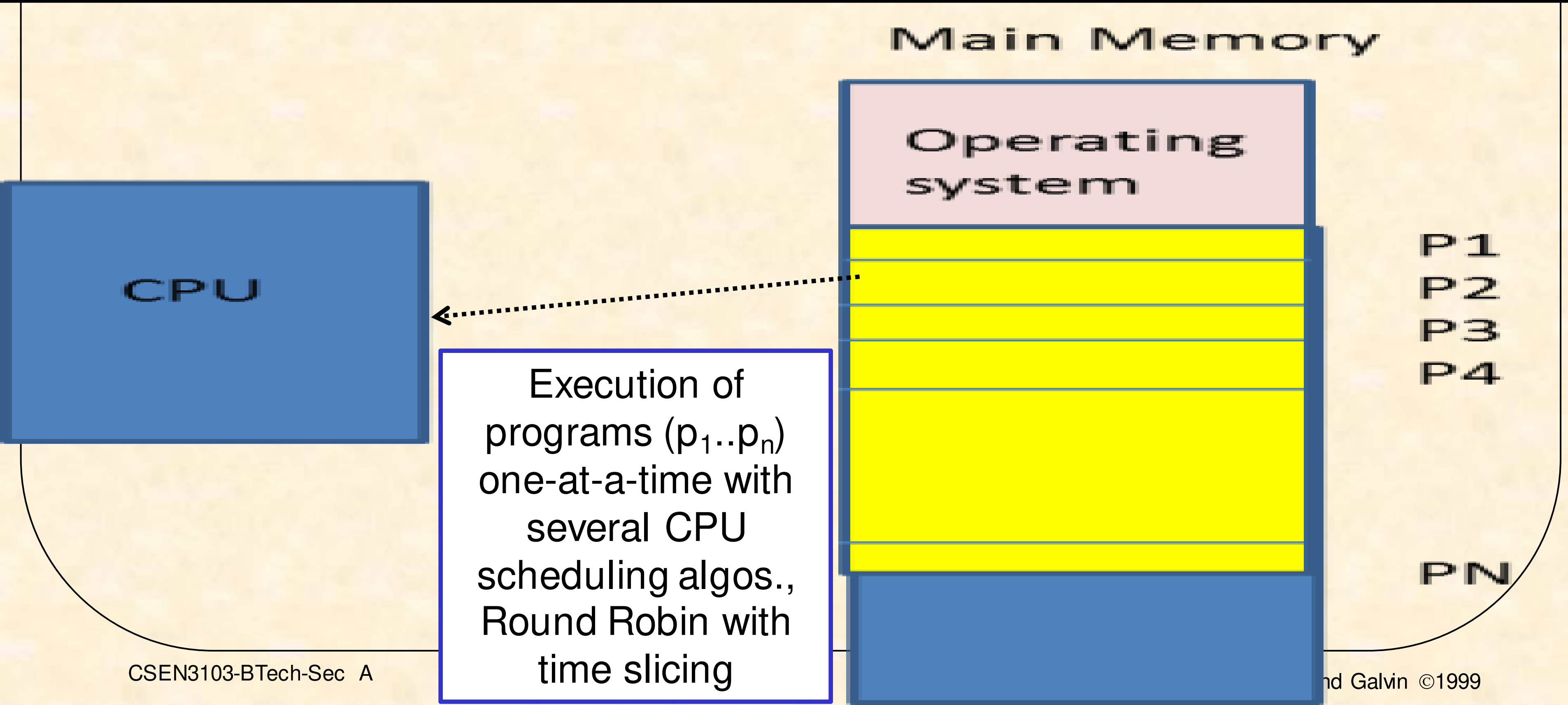
- Application software is bought by the user and loaded on the computer system to satisfy the user’s application needs.
- Example: An engineer might buy applications like python, netbeans, eclipse, oracle, codeblocks, MS-office, winzip et. al.

System Software

- This is a layer between application software and the machine.
- It provides services and resources to application software at one end and harnesses the power of the machine at the other end.
- Examples: Operating system, compilers, interpreters, assemblers, etc.

Little bit idea of some keywords we would use in this subject….

Any executable program must get space in memory before it is executed, instruction by instruction, by CPU. The OS loads the program into main memory (MM) and dispatches it to the CPU for execution. OS also provides other resources, like, CPU time, memory space, files, input-output devices, interrupts can be considered as instances of resources…..



Execution of programs ($p_1..p_n$) one-at-a-time with several CPU scheduling algos., Round Robin with time slicing

What is an Operating System….

1.What is operating system?

a) collection of programs that manages hardware resources

b) system service provider to the application programs

c) link to interface the hardware and application programs

d) all of the mentioned

Ans : (d)

2. Which one of the following error will be handled by the operating system?

a) power failure

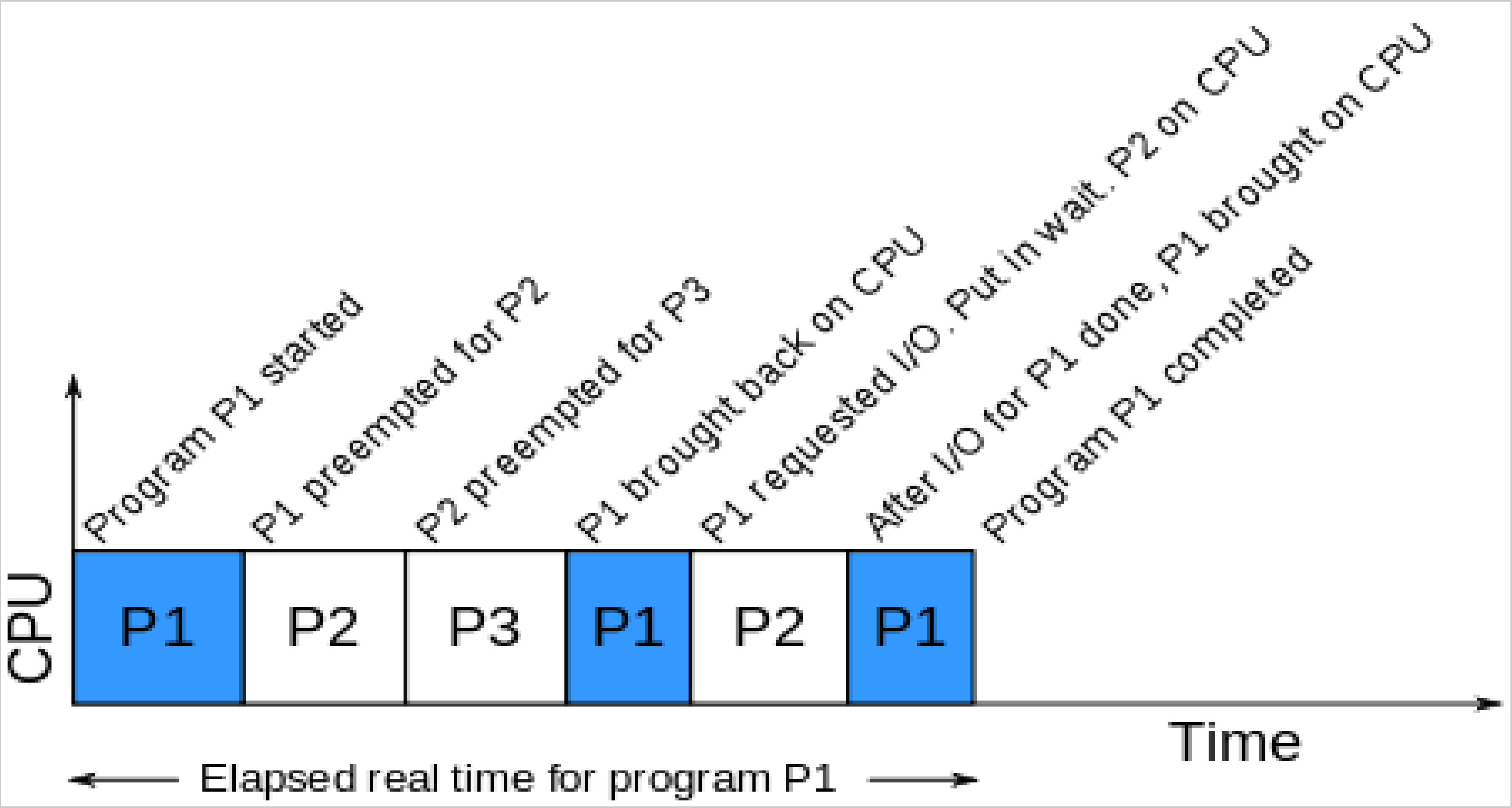
b) lack of paper in printer

c) connection failure in the network

d) all of the mentioned

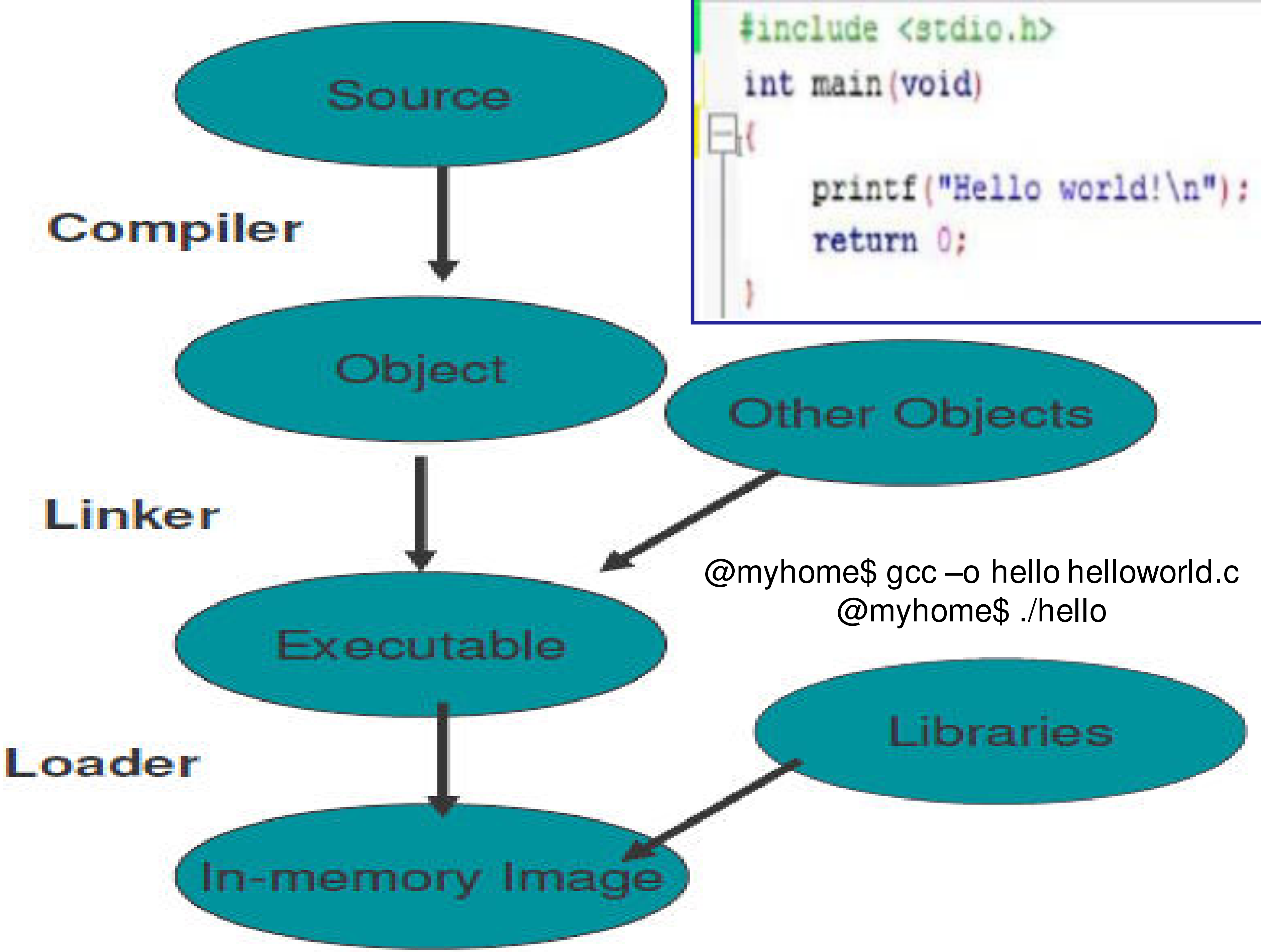
Answer: d

CPU scheduling



CPU Time on Single CPU Multi Tasking System

🟦 CPU color time for program P1



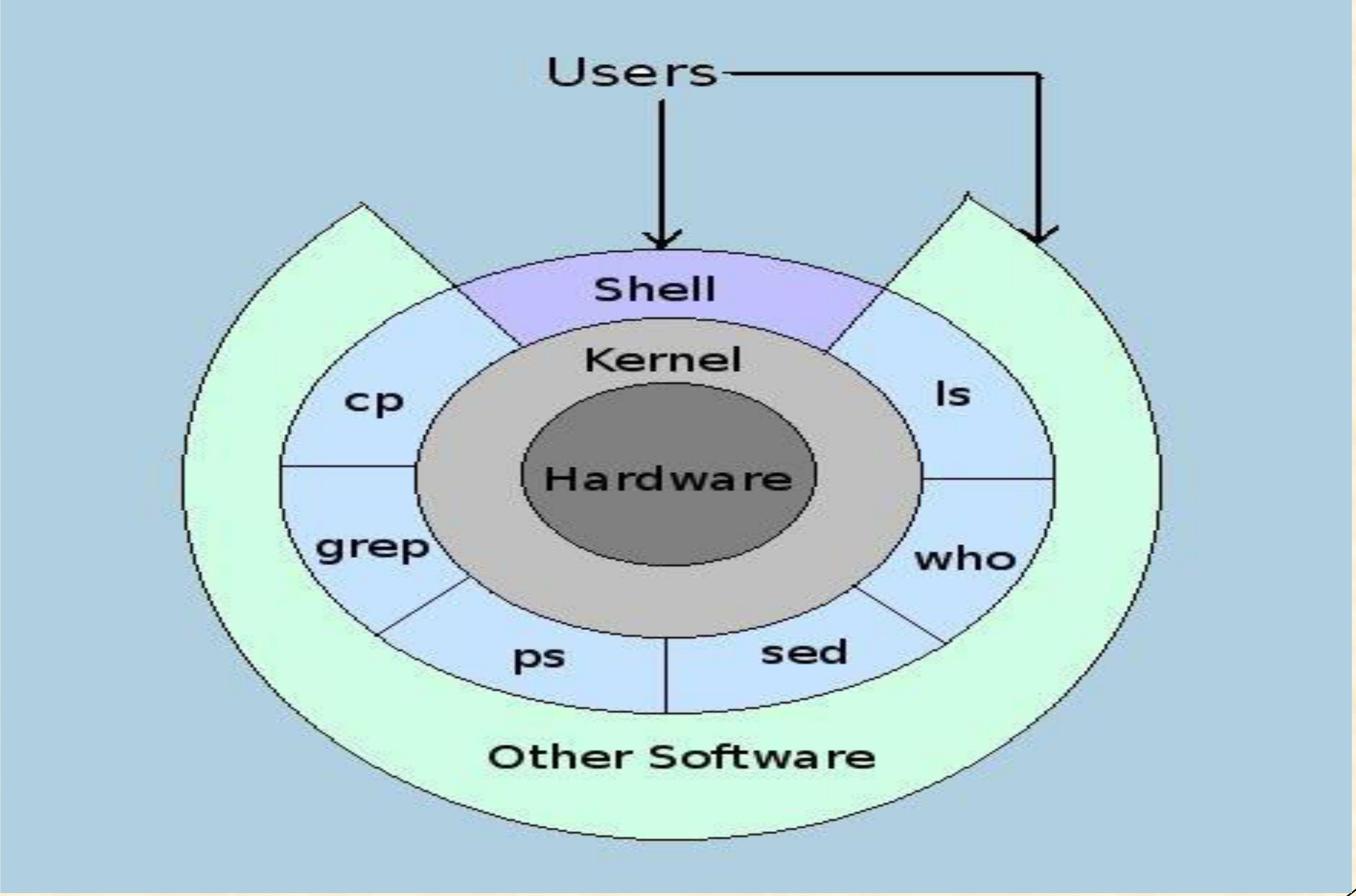
```c
#include <stdio.h>
int main(void)
{
    printf("Hello world!\n");
    return 0;
}
```

```
@myhome$ gcc –o hello helloworld.c
@myhome$ ./hello
```

Division of Labor: Kernel and Shell

- There are two agencies: the ***kernel* and *shell***
- The ***kernel*** interacts with the ***machine's hardware***, and the ***shell with the user***.
- The **kernel** is the core of the operating system-a collection of routines mostly written in C.
- Kernel is loaded into memory when the system is booted and communicates directly with the hardware.
- User programs (the applications) that need to access the hardware (like the hard disk or terminal) use the services of the kernel that performs the job on the user.

The Kernel-Shell Relationship



SOME POPULAR COMMANDS …..

- ***ps***: process status. *ps* displays the processes owned by the user running the command.

```
$ps
  PID     TTY            TIME        CMD
  291      console       0:00        bash   // the login shell of this user
```

Ignore the header. Each line shows the PID (process ID, unique per process), the terminal (TTY) with which the process is associated (the controlling terminal), the cumulative processor time (TIME) that has been consumed since the process has been started (CPU usage is negligible), and the process name (CMD: user is running the bash shell, our shell can be different).

- ***cp (copying a file):*** This command copies a file or a group of files. It creates an exact image of the file on disk with a different name.

```
  $cp  csen3103  csen3113
```

SOME POPULAR COMMANDS …..

- ***cp*** (***copying a file***): If the destination file (csen3113) doesn't exist, it will first be created before copying takes place.

```
$cp  csen3103  seca/csen3113  // csen3103 copied to csen3113 under seca directory
$cp  csen3103  seca           // csen3103 retains its name under seca
```

- ***grep***: filters using regular expressions, grep, that is used to search a file for a pattern and display the lines containing the pattern.

```
$grep "sales" emp.lst
2233| a.k. shukla     | g.m.       |sales  |12/12/15  |6000
1006| r. das          |  manager   |sales  |03/09/14  |6700
1006| s.n. dasgupta   | manager    |sales  |26/10/14  |7700
```

- ***sed***: the stream editor. Example: Line addressing

```
$sed '2q'  emp.lst
```

SOME POPULAR COMMANDS …..

- $*sed ‘2q’ emp.lst   // Quits after line number 2*

```
2233| a.k. shukla       | g.m.         |sales  |12/12/15  |6000
1006| r. das             |  manager  |sales  |03/09/14  |6700
```

- ***who: who are the users?***

```
$who
$who –Hu   // H→Header; u → list of users
$who am i
```

- ***ls: Listing directory contents***
- ***ls, ls –l, ls –al etc.***

More about Kernel…

- Providing support to User Programs is one job of Kernel.
- The kernel has a great deal of housekeeping, like, managing System's memory, scheduling processes, deciding the priority of processes.
- The Kernel is often called ***the*** operating system – ***a program's gateway to the computer's resources***.

The File and Process

- Two simple entities support the UNIX system – the *file* and *process*.
- ***Files have places* and *processes have life***.
- A file is just an array of bytes and can contain virtually anything.
- UNIX doesn't really care to know the type of file you are using.
- It considers even directories and devices as members of the file system.
- The dominant file type is text, and the behavior of the system is mainly controlled by text files.

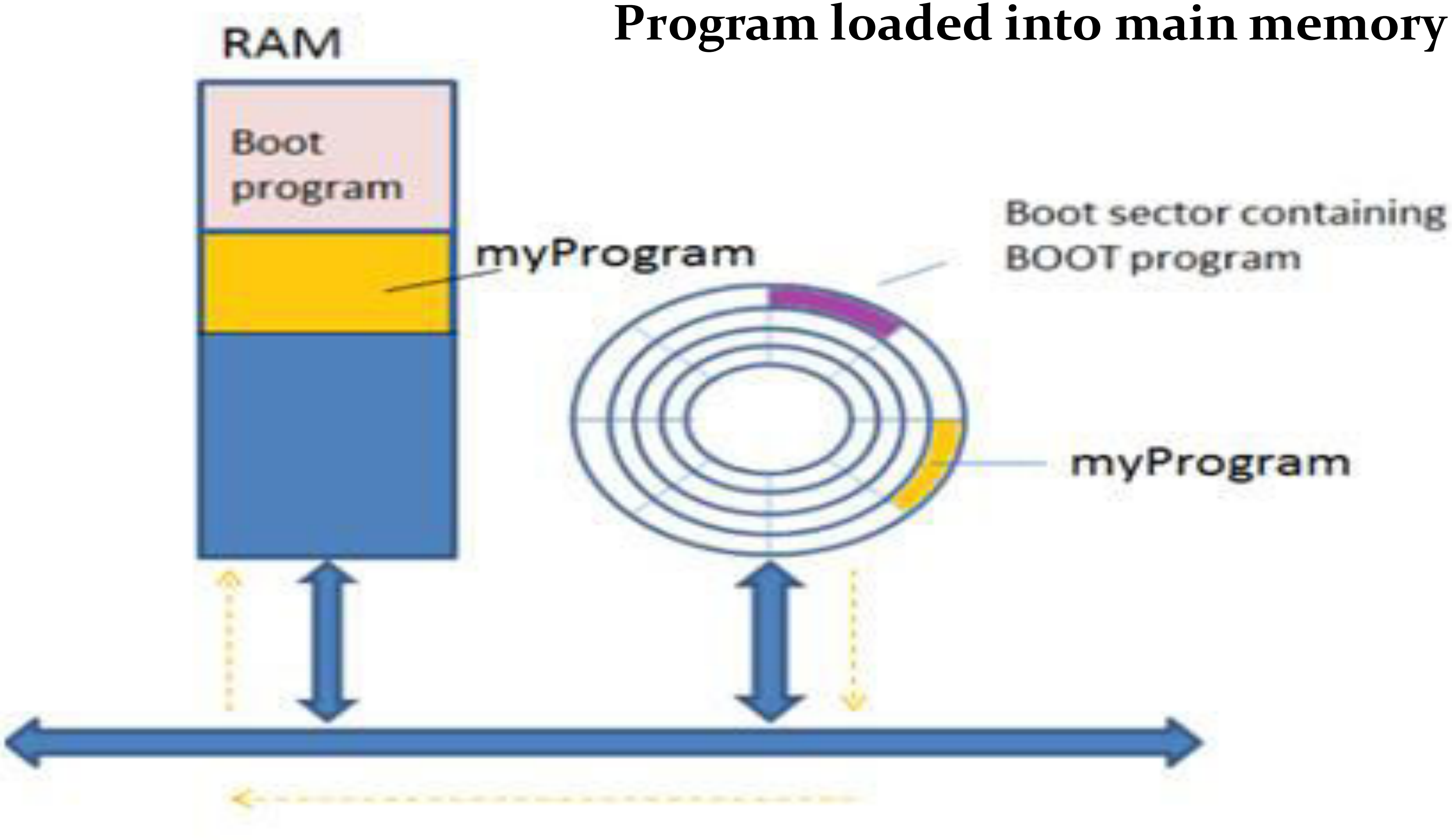
Phases of Evolution of Operating Systems

- Single user system
- Multiuser system
- Multiprogramming
- Multitasking
- Single user multitasking
- Batch processing
- Multiprocessing
- Parallel processing
- Network operating system

Single user system

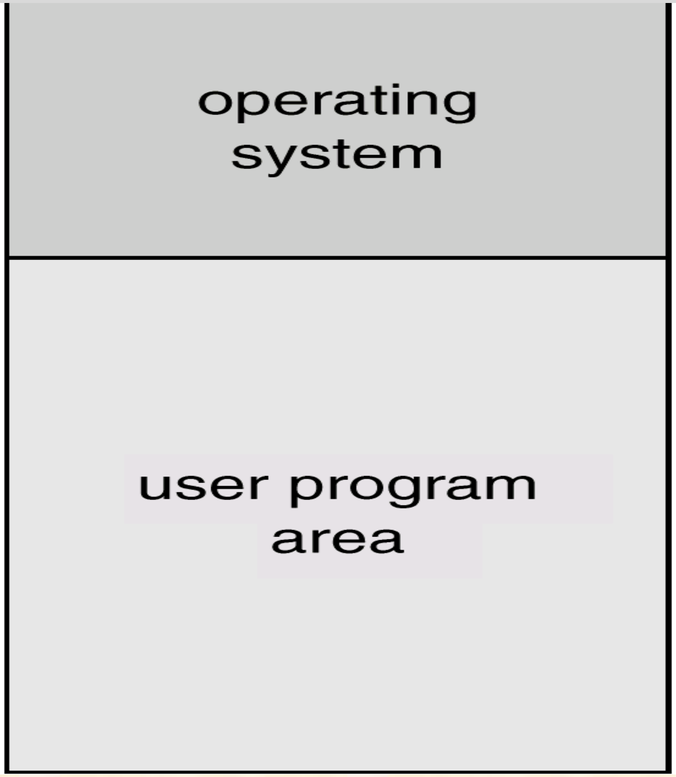
- The OS takes control of the computer. On a user request, it loads a program from the hard disk into main memory.

Program loaded into main memory



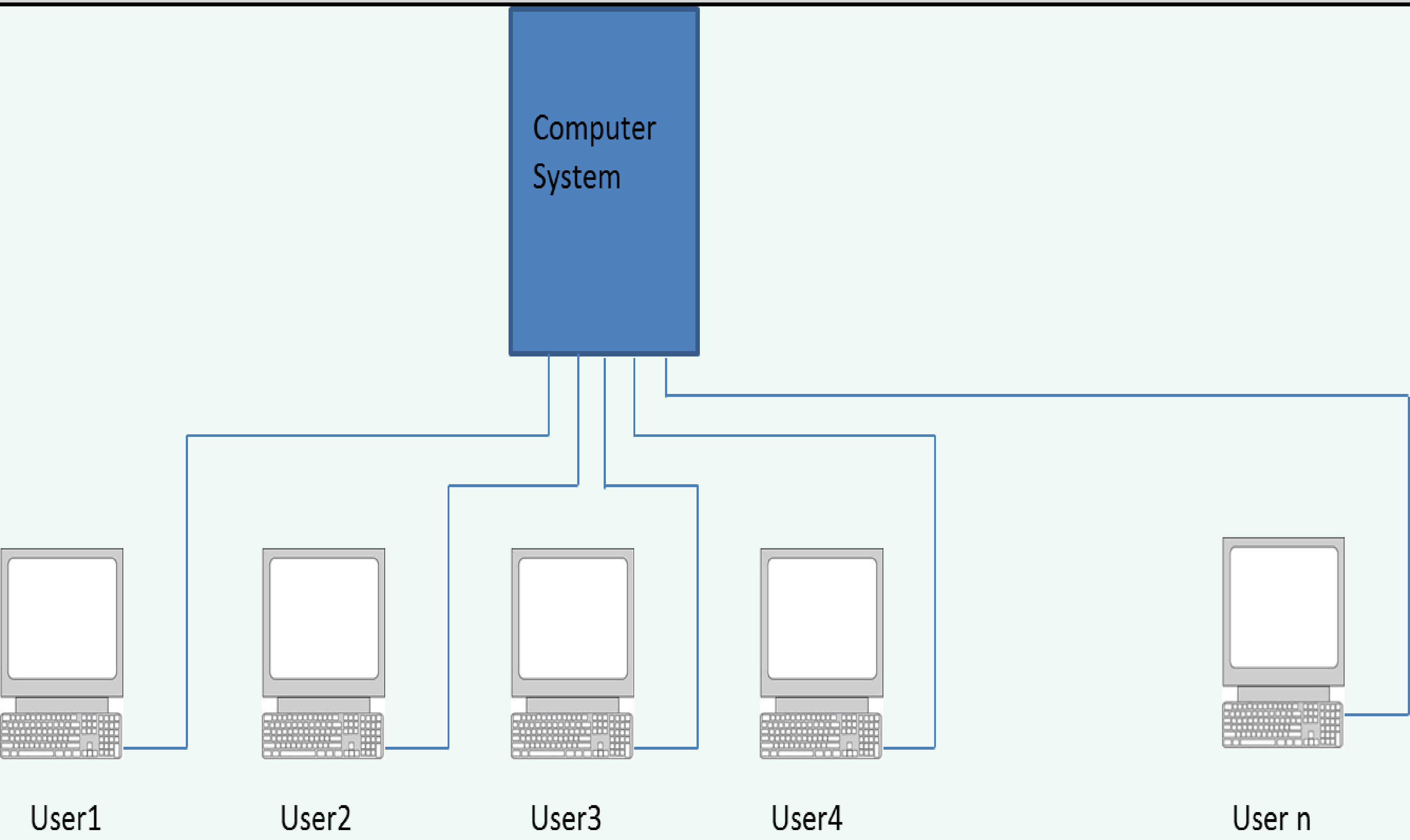
Disadvantages of Single User System

- After completion of a job, the loading of a new program requires an I/O operation from secondary storage devices.
- If the currently executing program asks for an I/O operation, the CPU has to wait till the I/O device provides the required service to the program.



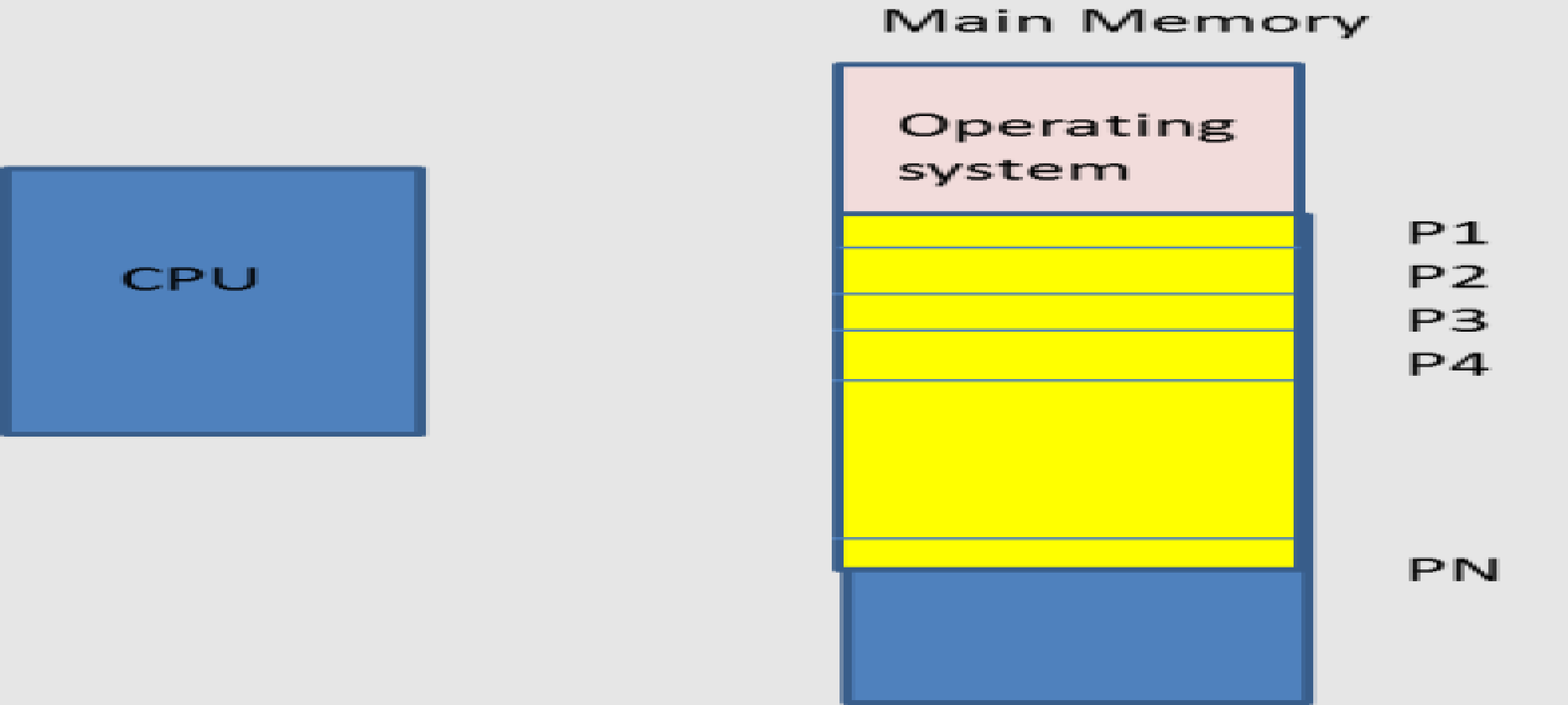
Multi-user System

In order to overcome the disadvantages of single user operating systems, computers were designed that they could support multiple users.



Multi-user to Multi-programmed Systems

If the currently executing program asks for an I/O operation, the OS suspends execution of the program and dispatches another program to the CPU for execution. Thus, the CPU never remains idle. ***The simultaneous residency of multiple programs in the main memory is called multiprogramming.***



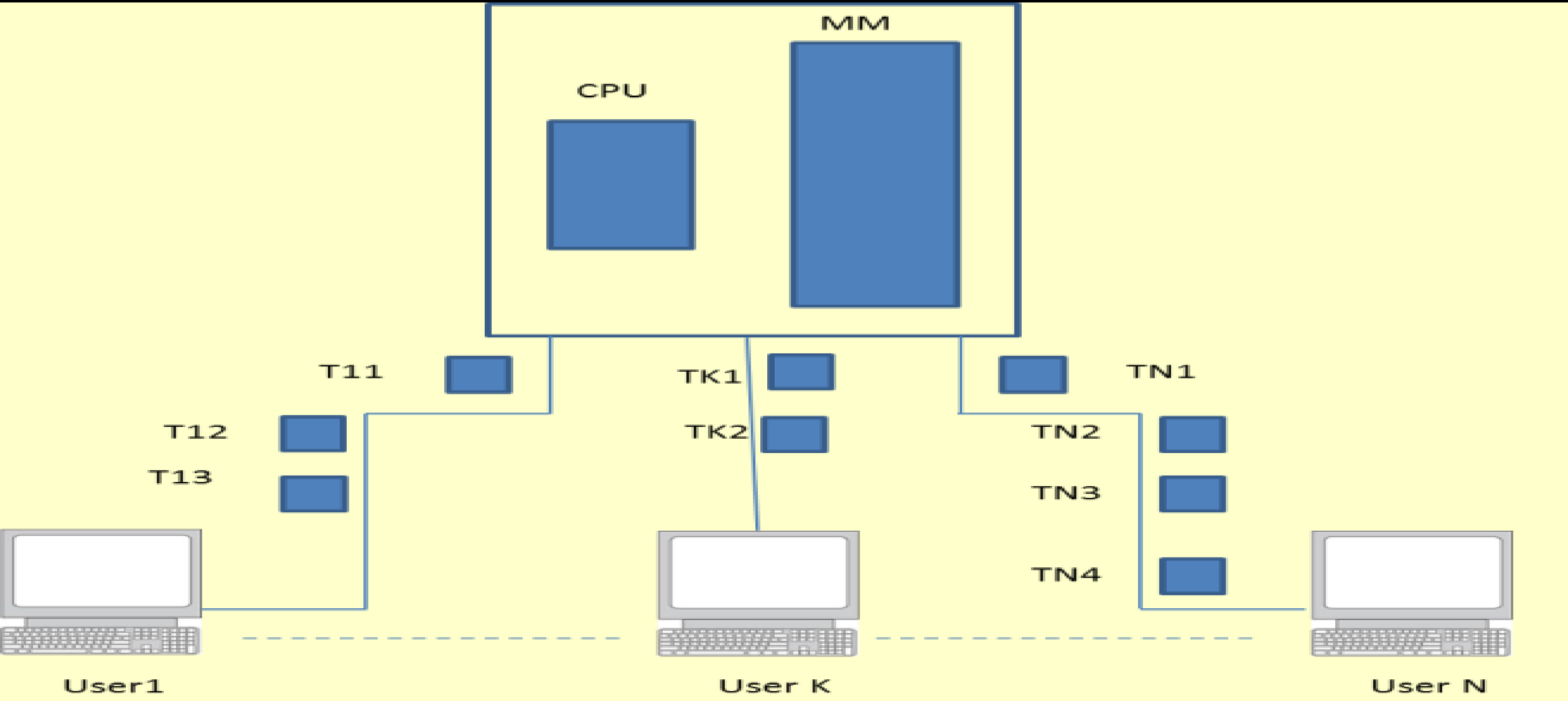
Objectives: Multiprogramming system is inherently a multi-user system

- Goal of multi programming: Maximum utilisation of CPU time
- Efficient management of main memory
- To achieve the above objectives, we need to answer the following questions:
  - Which program should be brought into main memory?
  - Of the resident programs, which one should be executed next?

- ***The ability of an operating system to accept more than one job from a single user is called multitasking.***

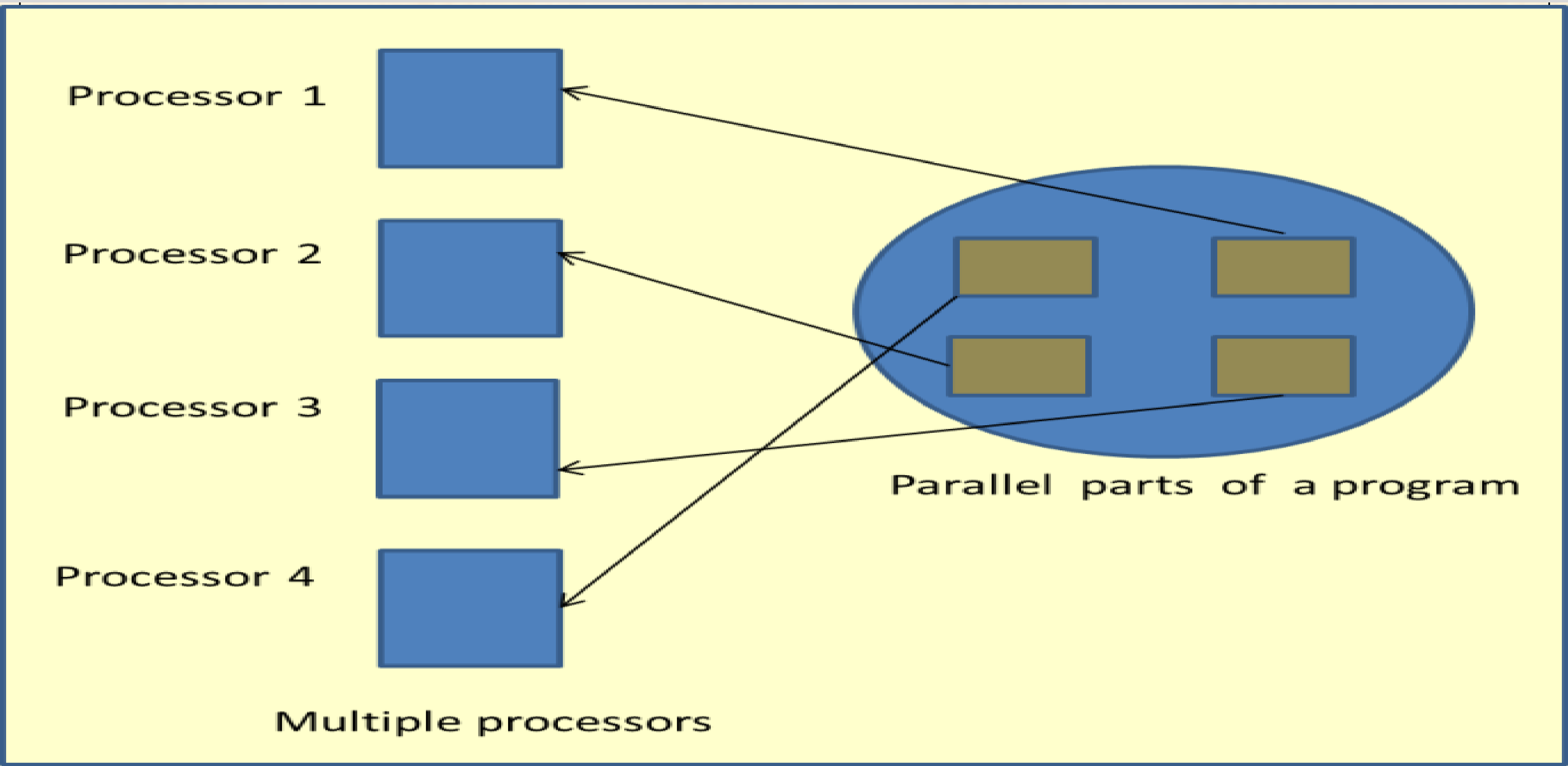
Multitasking System

- **Task**: The user or system job that needs to be performed.
- *The ability of an operating system to accept more than one job from a single user is called **multitasking.***
- Example: UNIX is a multi-user, multiprogramming and multitasking operating system.



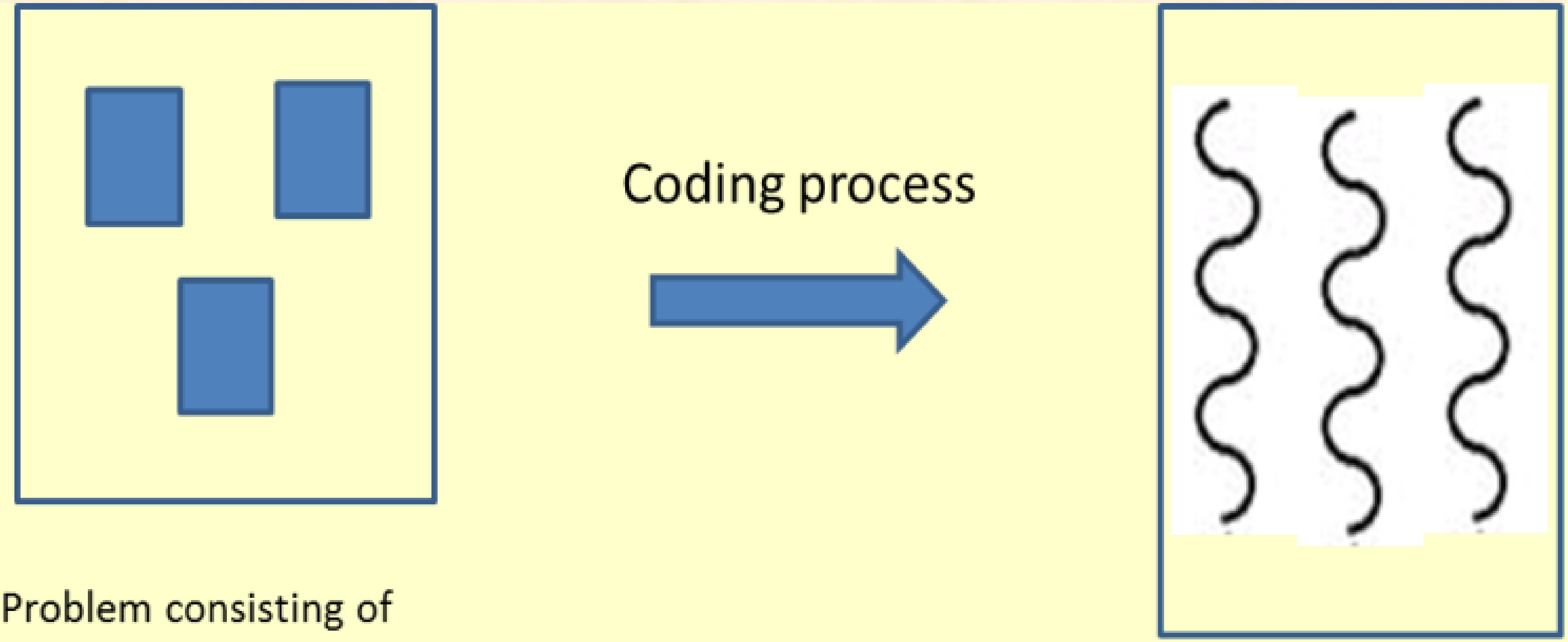
Parallel Processing

There are some algorithms which are inherently parallel in nature. Example, we have quick sort and merge sort dividing the data into partitions that can be processed independently of each other, provided a sufficiently number of processors are available in the computer system.



Multithreading operating system

A thread can be precisely defined as a separate path of execution through the code of a program. The operating system that can handle multiple threads of a program is called a multithreading operating system.

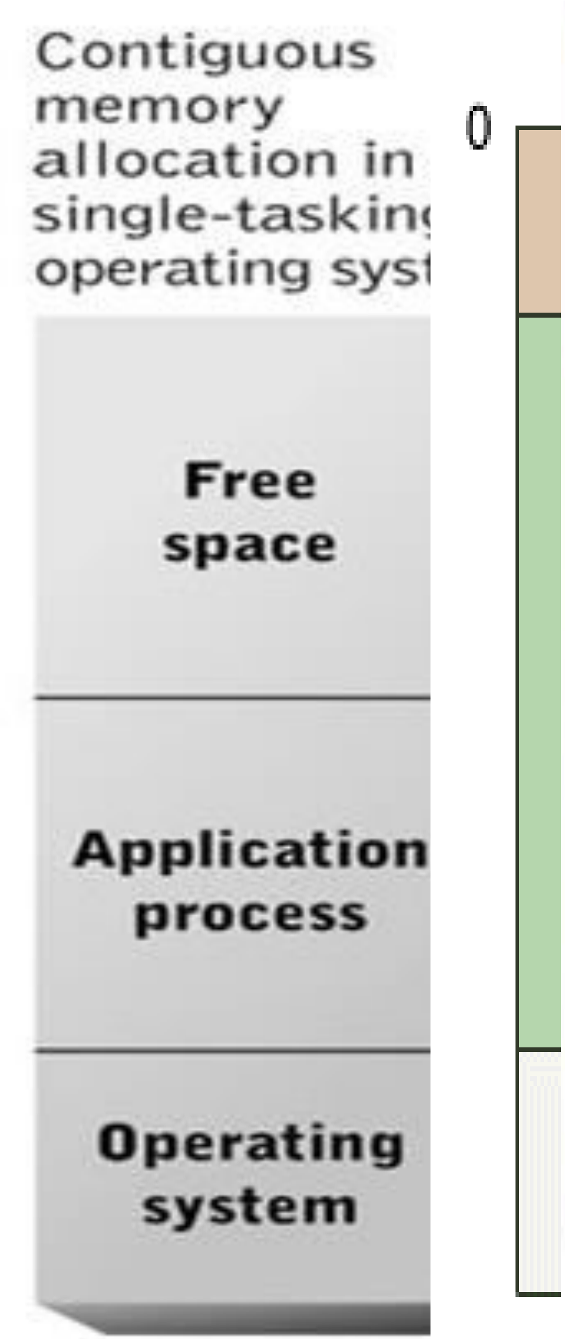


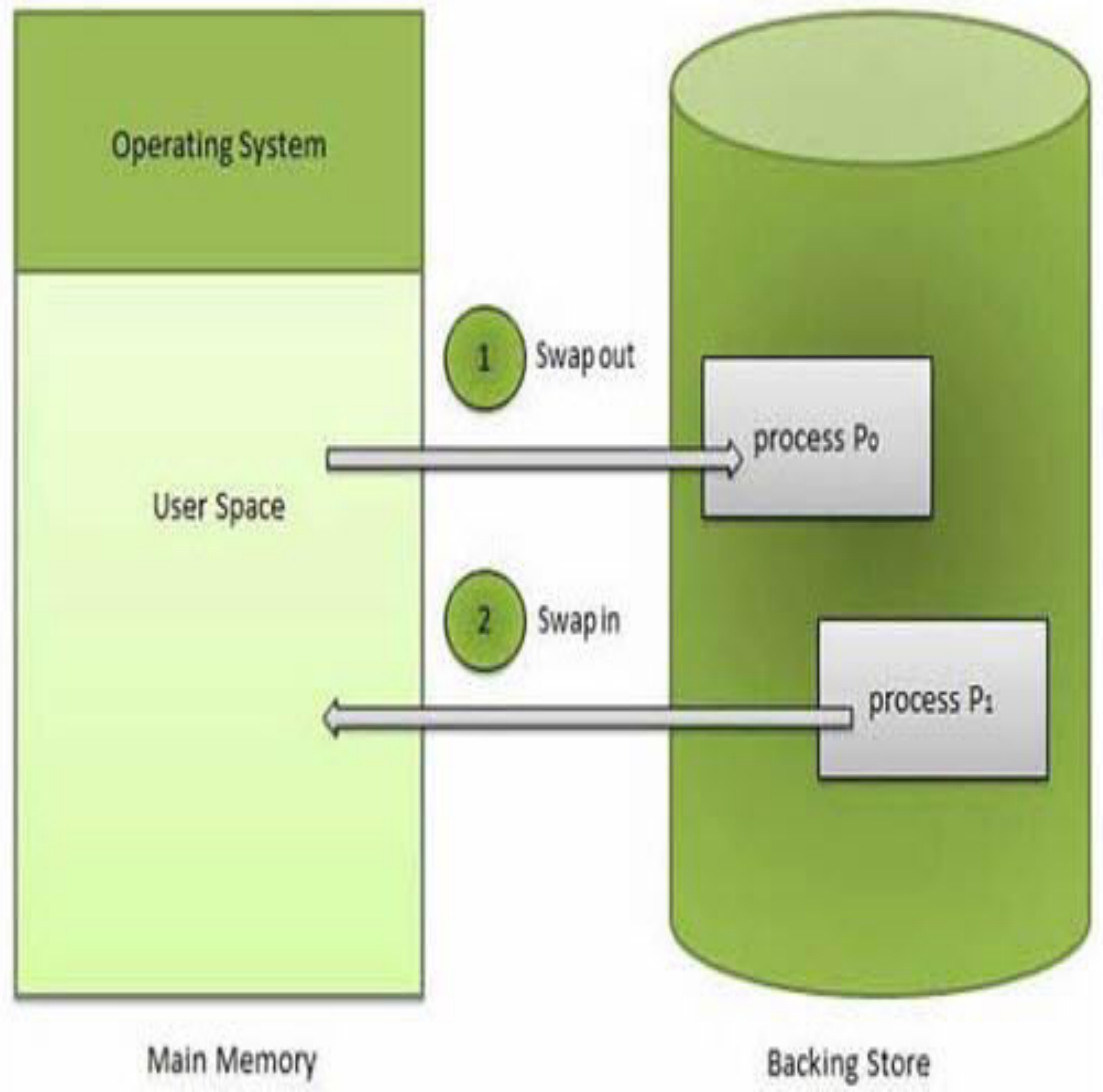
Coding process

Problem consisting of three independent activities

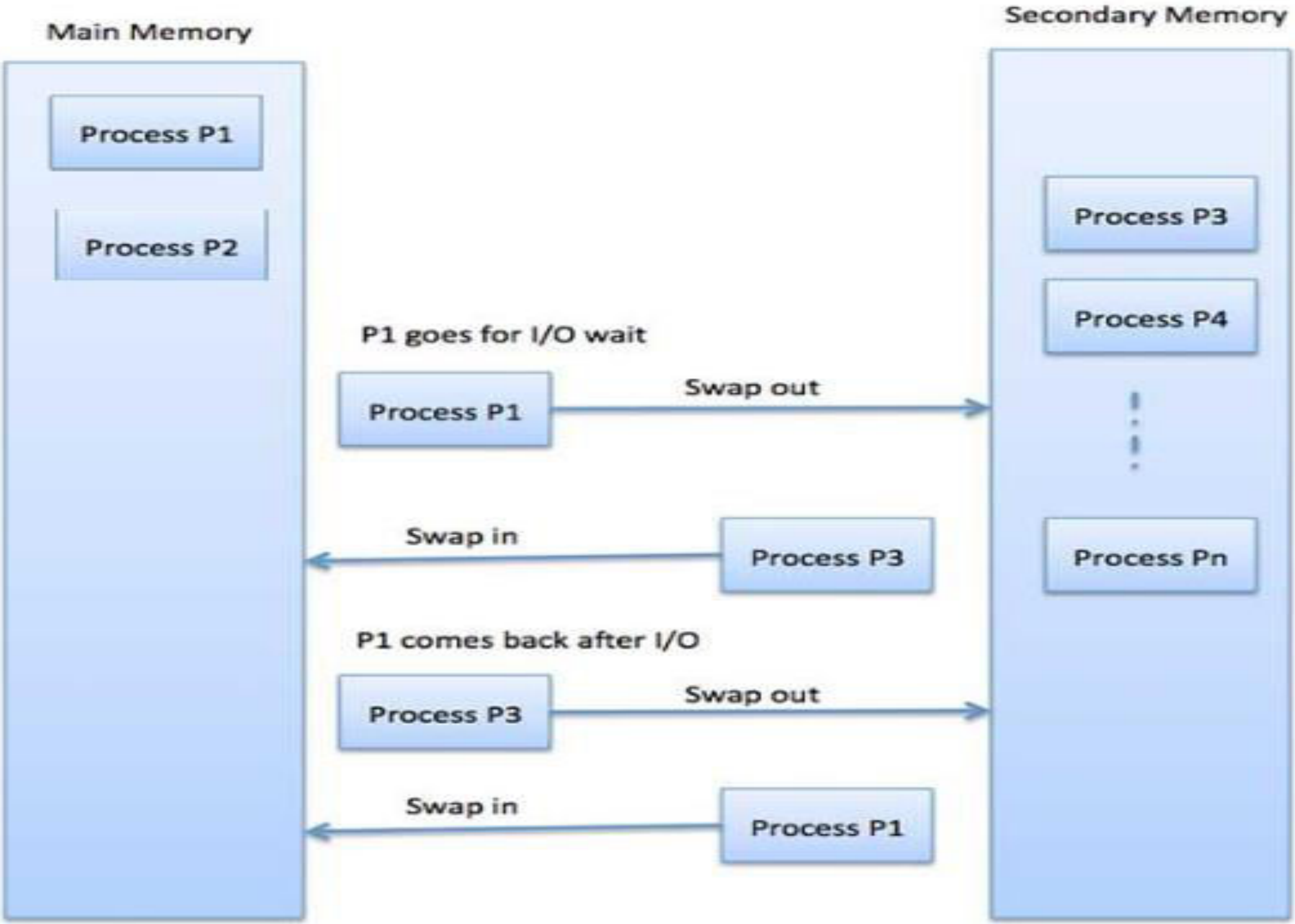
Program consisting of three independent threads

OS - MEMORY MANAGEMENT


Contiguous memory allocation in single-taskin operating syst
0
Free space
Application process
Operating system



OS - MEMORY MANAGEMENT…



Real-Time Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- *Hard real-time system*.
  - Secondary storage limited or absent, data stored in short-term memory, or read-only memory (ROM)
  - Conflicts with time-sharing systems, not supported by general-purpose operating systems.
- *Soft real-time system*
  - Limited utility in industrial control or robotics
  - Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.

Distributed Systems

- Distribute the computation among several physical processors.
- *Loosely coupled system* – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.
- Advantages of distributed systems.
  - Resources Sharing
  - Computation speed up – load sharing
  - Reliability
  - Communications

Distributed Systems (Cont.)

- Network Operating System
  - provides file sharing
  - provides communication scheme
  - runs independently from other computers on the network
- Distributed Operating System
  - less autonomy between computers
  - gives the impression there is a single operating system controlling the network.

Some info. that may be handy…..

- Windows, OS X of Apple and Unix are multithreading operating systems.
- A network operating system (NOS) can be define as an operating system that manages resources distributed over a computer network and enables communication across the network. Ex: Novell NetWare 2.5, Windows NT Server and Mac OS AppleShare are network operating system.
- Real time operating systems (RTOS) have the following attributes: 1) ***Fault tolerance; 2) Deterministic scheduling***; There may be jobs such as *flight control, telephone switching, missile control* etc., require a response to events within a certain amount of time. These time sensitive jobs are called real-time jobs or processes. RTOS provides services to these processes. *Job scheduling is the most important activity of RTOS, i.e., deciding which one of the available jobs will run next so that the impending deadlines are met.*

Functions of Operating Systems ➔ what's coming up……

- Process management (processes, threads, cpu scheduling, process synchronization, mutual exclusion, deadlock, semaphores, monitors…..)
- Memory management (MM + VM)
- Processor management (keywords given… concurrency control….)
- Input/Output and file/disk management
- Security