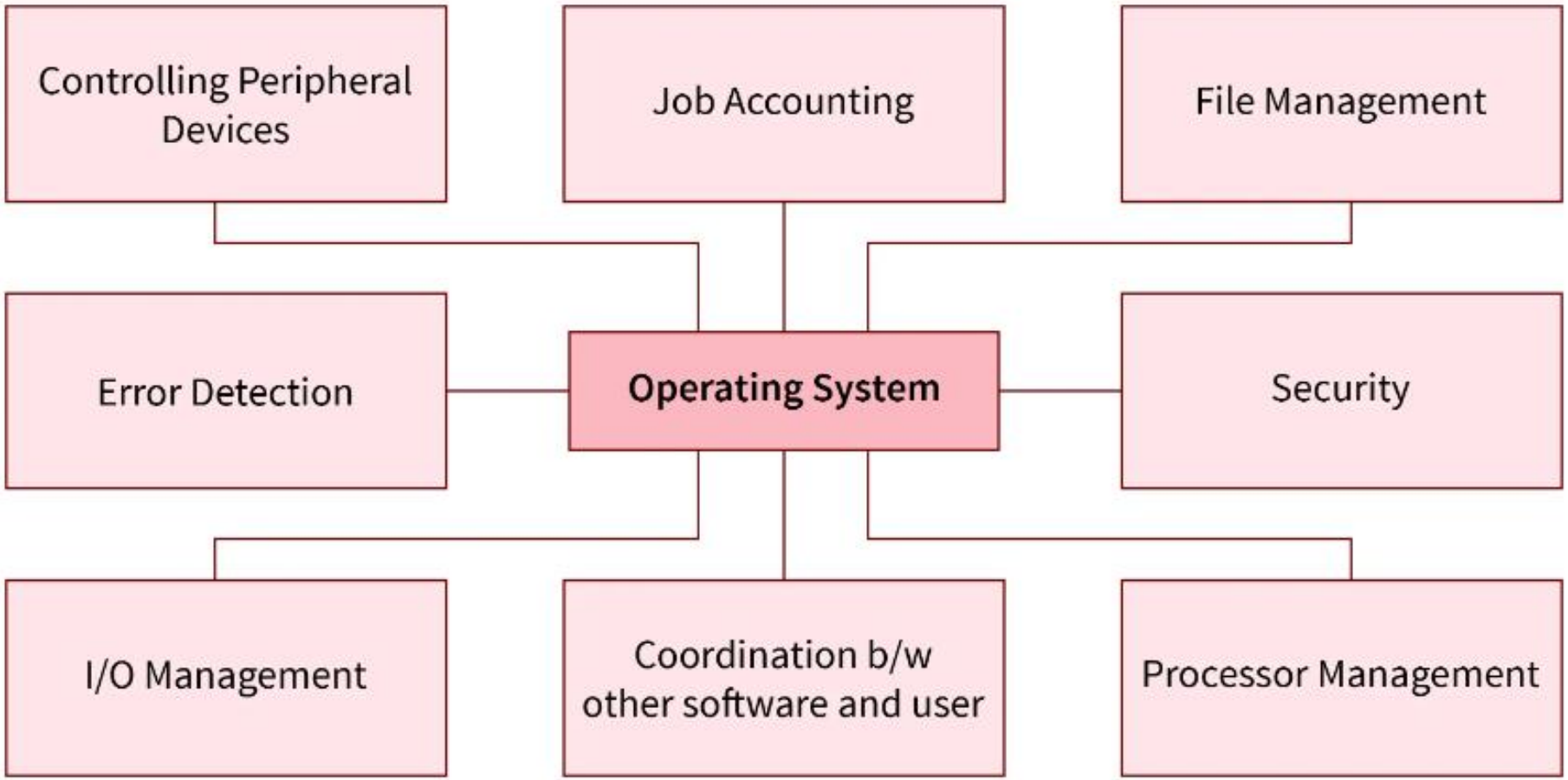
Operating System Concepts

Introduction to OS

- An Operating System is a System Software.
- Acts as an interface between computer hardware and the user.
- Every computer system must have at least one operating system to run other application programs.
- Applications programs include Browsers, MS Office, Notepad, Games, etc.
- Applications need some environment to run and perform their tasks.
- Popular Operating Systems are-

  Windows, Unix, Linux, MAC OS, Android
  and many more…..

Architecture of OS



Process Management

- An Operating System performs each and every task as a **Process**.

- Every OS has a Process management system which includes variety of responsibilities like process development, scheduling, process termination, etc.

What is a Process?

Process is actually a program in execution or a running program.

Example-

When we write a C or C++ code, the compiler generates binary code. When this binary code is executed, it turns into a process.

- There may be more than one process in the system which require the same resource at the same time.

- It is the Operating System that manages all the processes and the resources in a convenient and efficient way.

Context of a Process

- Every process has some attributes, also known as **"Context of the Process"**.
  - **Process ID-** A unique identifier assigned by the OS.
  - **Process State-** Can be ready, running, wait, etc.
  - **CPU Registers-** It is the program counter that is saved and restored when a process swapped in and out of the CPU.
  - **Accounts Information-** CPU used for process execution, execution id, etc.
  - **I/O Status-** Devices allocated to the process, open files, taking input, etc.
  - **CPU Scheduling Information-** Assigns priority to every process.
- All the above mentioned attributes of a Process are kept inside a **"Process Control Block"** (PCB).
- Each Process has its own PCB.

States of a Process

When a Process is created, it may be in one of the following states-

1. **New-** Newly Created Process or a process being-created.
2. **Ready-** After the **New** state, process moves to the **Ready** state, means the process is ready for execution.
3. **Run-** Currently running process in CPU. A single processor can execute only one process at a time can.
4. **Wait-** Process requests for Input/Output access.
5. **Complete(or Terminated)-** Process completed its execution.
6. **Suspended Ready-** When the ready queue becomes full, some processes are moved to a **suspended ready** state.
7. **Suspended Block-** When the waiting queue becomes full.

Context Switching

- **Context Switching** is a procedure that a CPU follows to change the process from one state to another and ensuring that the processes do not conflict with each other.
- Context Switching helps to share a single CPU among all processes to complete its execution.
- If a high priority process falls into the ready queue, the currently running process will be terminated or stopped by a high priority process to complete its tasks in the system.
- If any running process requires I/O resources in the system, it will be switched by another process to use the CPU and when the I/O requirement is met, the old process goes into a ready state to wait for its execution in the CPU.
- If a running process interrupts, the process status is saved as registers using context switching. After resolving the interrupts, the process switches from a wait state to a ready state to resume its execution.

Process Queues

- The OS manages various types of queues for each of the process states.
- If the Process is moved from one state to another state then its PCB is also moved to the corresponding queue.
- There are three following queues maintained by the OS-
  - **Job queue-** At the beginning, all the processes get stored in this. The job scheduler pick some jobs/processes from job queue and put them in the primary memory to execute.
  - **Ready queue-** It is maintained in the primary memory. The process is picked from here and dispatched to CPU for execution.
  - **Waiting queue-** When a process requires some I/O operation, OS changes its state from running to waiting.

Memory Tables

- Used to keep track of both main and secondary memory.
- Processes can be in main or secondary memory.
- Memory tables include information such as-
    - Allocation of main memory to processes
    - Allocation of secondary memory to processes
    - Protection attributes of blocks of main and virtual memory
    - Any information needed to manage virtual memory

I/O and File Tables

- I/O tables are used by OS to manage the I/O devices and channels.
- In case of I/O operation in progress, OS needs to know the status of I/O operation and location in main memory.
- File tables provide information about the existence of files, their location on secondary memory, current status and other attributes.