Day 2

> Date : 28 Aug 2024

OS Content Discussed Day-2

- Linux commands revision
- implementation of chown, adduser, Head, Tail, Cat commands
- Redirection variable (>)
- Process and Process Management

1. What is process;
2. preemptive and non-preemptive processes
3. Process management;
4. Process life cycle
5. What are schedulers – Short term, Medium term and Long term.
6. Process scheduling algorithms – FCFS, Shortest Job First, Priority, RR, Queue.
7. Examples associated with scheduling algorithms to find turnaround time to find the better performing scheduler.

- Memory Management

1. Continuous and Dynamic allocation
2. First Fit, Best Fit, worst Fit
3. Internal and external fragmentation
4. Compaction

Process Management

- CPU sched information, memory management information, information about resources allocated for that process, execution context etc...
- Running program is called as a process.
- When a program gets loaded into the main memory .
- It is referred as a process.
- Running instance of a program is referred as a process.

Execution of process is always done by processor means cpu means ram , but with requirements change , mode also changes .

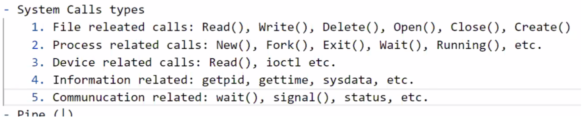
1. User Mode : When process are running in normal mode , they don't need any other devices , they run in user mode .
2. Kernel Mode : When process are running in kernel mode , they need any other devices , then processor will run the process in kernel mode (Under the supervision of kernel).

> **Software Interrupt / Trap ⇒ A software interrupt is a signal from software to the processor that indicates an event that needs immediate attention**

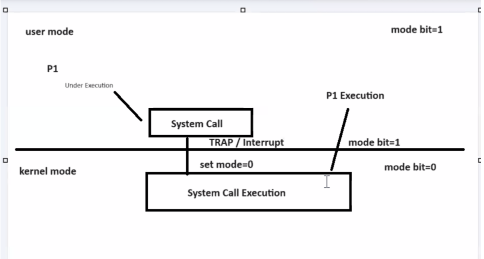
System Calls

1 . When we want to access any IO devices(reading files , printing , etc) we may take the permission from kernel (kernel has all the access of hardware) , want to make a system call .

2. Then CPU will immediately switch from User Mode to Kernel Mode .
3. Meanwhile the current process will be waiting in the queue(IO queue) to access the of IO device .
4. Whenever process is running in user mode ⇒ there is mode bit which is set to 1 when in user mode , when system call(Kernel mode) is generated the mode bit is set to be zero .
5. After the execution of IO device , kernel will generate an interrupt , and the mode will again switch to user mode .
6. They are only method how communication happens.
7. System call change the mode bit only .



```
- System Calls types
    1. File releated calls: Read(), Write(), Delete(), Open(), Close(), Create()
    2. Process related calls: New(), Fork(), Exit(), Wait(), Running(), etc.
    3. Device related calls: Read(), ioctl etc.
    4. Information related: getpid, gettime, sysdata, etc.
    5. Communucation related: wait(), signal(), status, etc.
- Pipe (|)
```



Working of Process :

> Per Process a PCB (Process Control Block) is created , where all the imp info about the process is stored and PCB is itself a part of process . PCB is the metadata of process mainly contains:
> • PID : Process ID
> • PPID : Parent Processes ID
> • PC : Program Counter
> • Execution context
> • Kernel stack
> • Exit status

> CPU sched information, memory management information, information about resources allocated for that process, execution context etc...

Process Life Cycle :

Throughout execution, process goes through different states out of which at a time it can be
only in a one state.

- States of the process:

1. **New state:** upon submission or when a PCB for a process gets created into the main
memory process is in a new state.
2. **Ready state:** after submission, if process is in the main memory and waiting for the CPU
time, it is in a ready state.
3. **Running state:** if currently the CPU is executing any process then state of that process is
considered as a running state.
4. **Waiting state:** if a process is requesting for any i/o device then state of that process is
considered as a waiting state.
5. **Terminated state:** upon exit, process goes into terminated state and its PCB gets destroyed from the main memory.

CPU schedulers : They perform process life cycle .

1. Long Term scheduler : It is a system program which selects/schedules jobs/processes from job queue to load them onto the ready queue.
2. Short Term scheduler : it is a system program which selects/schedules job/process from ready queue to load it onto the CPU.

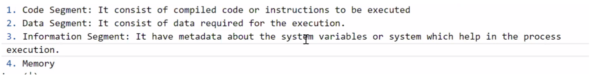
   # Dispatcher: it is a system program which loads a process onto the CPU which is scheduled by the CPU scheduler, and the time required for the dispatcher to stops an   execution of one process and to starts an. execution of another process is referred as dispatcher latency.

3. Mid Term scheduler : it is a system program which selects/schedules job/process from ready queue to load it onto the CPU.

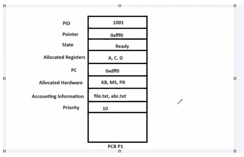
   # Starvation: in this algorithm, as shorter processes has got higher priority, process which is having larger CPU burst time may gets blocked i.e. control of the CPU will never gets
   allocated for it, such situation is called as starvation/indefinite blocking.

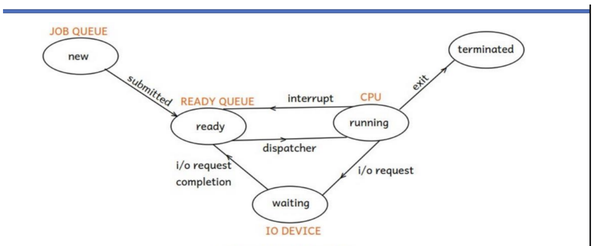
   a . Preemptive scheduling

   b . Non preemptive Scheduling



1. Code Segment: It consist of compiled code or instructions to be executed
2. Data Segment: It consist of data required for the execution.
3. Information Segment: It have metadata about the system variables or system which help in the process execution.
4. Memory





PROCESS STATE DIAGRAM

CPU Scheduling Algo :

CPU Scheduler gets called in the following four cases:

Case-1: Running ⇒Terminated
Case-2: Running ⇒ Waiting
Case 3: Running ⇒ Ready
Case-4: Waiting ⇒ Ready

⇒ There are two types of CPU scheduling:

1. Non-preemptive: under non-preemptive cpu scheduling, process releases the control of
the CPU by its own i.e. voluntarily.

- e.g. in above case 1 & case 2

1. Preemptive: under preemptive cpu scheduling, control of the CPU taken away forcefully
from the process.

- e.g. in above case 3 & 4

1. **FCFS (First Come First Served) CPU Scheduling**
2. **SJF (Shortest Job First) CPU Scheduling**
3. **Round Robin CPU Scheduling**
4. **Priority CPU Scheduling**

Do every kind of numerical on this schedules

Memory Management

1. Memory means RAM management. Which is divided into number of blocks which can be of fixed size or variable size.
2. This process is known as fixed size partition or Variable size partitioning .
    a. **Fixed** ⇛ Memory are divided into fixed size blocks . The process which will get the memory may be small or equal to the then the partitioning size . As the process size can be smaller than the block , it give rise to **internal fragmentation.**
    b. **Variable** ⇛ Memory are given with accordance to the available space and variable size blocks . This leaded to external fragmentation when memory cant be allocated contigous .
    c. Compaction is the solution of external fragmentation .
    d. Compaction is very time consuming process , and cpu doesn't work at that time , waste of cpu , lower the overall efficiency.
    e. **First fit** ⇛ The process when come in memory , and have blocks variable , and the coming process needed 2 bytes , so it will get allocated to the very first available without seeing the space according process.
    f. **Best fit** ⇛ The Os will allot the space with according to the size process , and assign minimum or equal space avaiable with accordance to size of process.
    g. **Worst fit** ⇛ Os is assigning the largest available space but the process size is too low .

Topic: Recording
Module: COS
Date: 28/08/24
Session: Morning

https://us02web.zoom.us/rec/share/y8PdzRgNRuDLIQPWsgAU7j1pfipTj2ND-vVrAeGgpUvSwj4TcQ35tes4GfMNActQ.7lm_KHvxcKlWl-xC

Session: Afternoon

https://us02web.zoom.us/rec/share/k0RAXTVtsN11UttAmyoy49O7r8MNgUGmiEWkEDvARkCR74JCUQemHo8paCEuU24k.axCHglIlU6BSyX-0