Date No

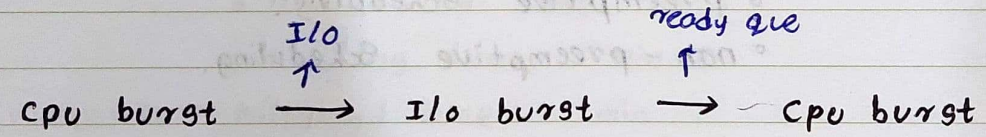
Cpu Scheduling - Short term Scheduling (Sos) lecture 05

- Cpu Scheduling used in multiprogramming systems to utilize the cpu usage in positive manner.

- have 2 types of burst in Scheduling.
  - Cpu burst
  - I/o burst

- cpu burst - The time took the process to being execute in the processor.

- I/o burst - When the cpu is whaiting for I/o for further execution.



- Start with cpu burst and end with also cpu burst.

- have 2 types of bound mesureses.
  - I/o bound
  - cpu bound.

- cpu bound - time took to Complete a task determind with Speed of cpu. have few but long cpu burst
- I/o bound - time took to Complete I/o operation. have Short cpu burst (dealing with cpu with short time)

Date No

Cpu Skheduler. (short term Scheduler)

- this will select an process from ready que and allocate to cpu.
- Cpu Schedular will select a new process only in,

① o process Switches from running State to Whaiting State (I/o whait) (Parent whait for child end)
② o Switches from running to ready (Interrupt)
③ o Whaiting to ready (after I/o completion).
④ o Terminates (end of a process).

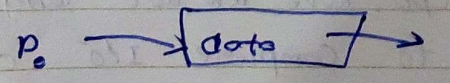
① and ④ → must select a new process.
② and ③ → can select any.

- Cpu Skheduling has 2 types
  - preemptive schedulings.
  - non-preemptive Scheduling.

- Preemptive - has a timer. each prosses given a time to execute in cpu. (newdays) - 2, 3

- non-preemptive - Cpu only released by the time of termination or to move to Whaiting que. no timer, (1, 4)

- Preemptive Scheduling can cause race conditions. many proses access the same data at given time



Date No

Dispatcher.

- this component give the control of the cpu to the Schedular that select the current process. (Short term Schedular or cpu schedular.

  - The time took to stop a process start another called dispatch latency.

  - Dispatcher involves,
    - When switching context
    - running user application
    - When restarting a program.

Using below points we can mesure the algorithem.

- CPU Utilization - max
- Throughput - max
- Turnaround time - min
- Whaiting time - min
- Repond time - min

refer 5.10 for more.

There are 3 types of cpu scheduling algorithems

- first come, first served (FCFS)
- Shortes job first (SJF)
- Round Robin (RR)

Date No

(01) first come first served (FCFS)

- Which process coming first will be served
- if the burst time was short average

Whaiting time will be reduced.

Convoy effect - Short processes behind long proces

to this solution SJF algorithem come.

(02) Shortest Job first, (SJF)

- SJF gives minimum whaiting time
- In here first have to find the cpu burst.
- In preemptive called Shortest-remaining-time first.

Cpu burst can be predected using exopential averagh formula.

$$\tau_{n+1} = \alpha t_n + (1 - \alpha)\tau_n$$

- if there is no recent histor $\alpha = 1$;
- if there it will be the Value.
- if we dont care $\alpha = 0$.

Date No

(03) Round Robin algorithem (RR)

- each process gets a small unit of cpu time. (time quantem) (10 - 100 milliseconds).

- in round robin performance depend on a time quantem.

  - if que is large → FCFS
  - if q is small → have to do context switch

- We want quantem to be larg.
- good for interactive systems.

(04) Priority Scheduling.

- based on the priority cpu will allocated and each process given a ~~time~~ number

  - Smallest priority → highest integer and vise versa.

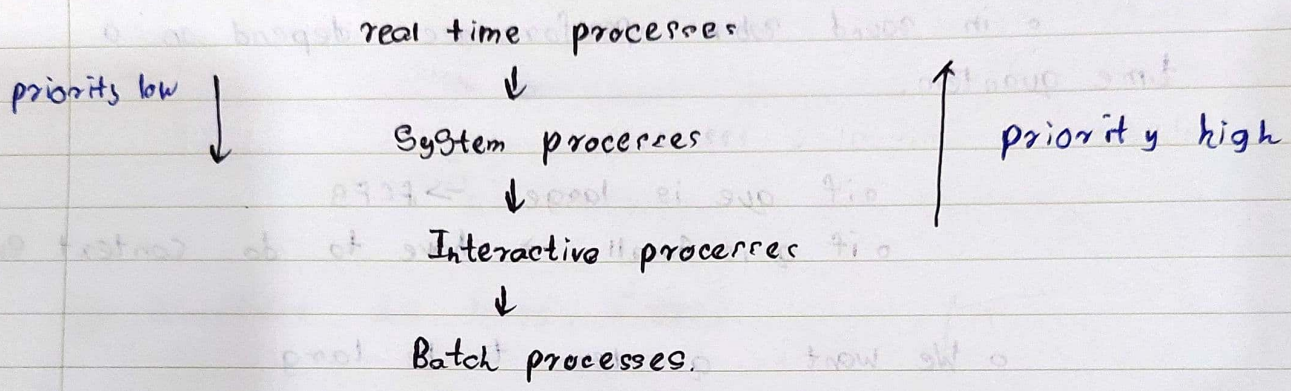
- SJF is priority Scheduling method.

Starvation - low priority processes may not execut
Solutions (aging) - increase the priority of the process.

Date No

(05) Multilevel que.

- We can have Separate que for each priority. prioritization can do based on process type.



(06) Multivelev feedback ques.

- here processes can move between ques.

refere 5.30 to more.

- by this method bw priority ques can be moved to higher priority que.

Thread Scheduling.

there a 2 types of threads.

- user-level threads
- kernel-level threads.

- if the threads support then threads will be Scheduled not the processes.

Date No

- Thread library will schedule user-level threads to light weigh processes. (LWP). also known as process contention scope (PCS)

- System contention scope(SCS) will decide which the kernel thread schedul into cpu. this

~~Pthread Scheduling~~.

Multiprocessor Scheduling.

- this is used when a system has more than a single processor.
- Multiprocessors can be,
  - Multicore cpu's
  - Multi threaded core's
  - NUMA system
  - Heterogeneous multiprocesing

Symmetric multiprocessing (SMP)

processer schedule it self. has a common ready que. also has a private que of threads

refer 5.38 for more., 5.39

Multithreaded Multicore System.

- each core has a thread. if one thread has not enough space it will switch to another thread.
- Chip multi threading (CMT) - assings each Core multiple threads.

~~Multicor~~
- this System has 2 levels of Scheduling.
  - OS decides which software thread to run on cpu.
  - each core decided which hardware to run on physical core.

Load balancing,
- how the work load is distributed among processors. 2 ways to do it.
  - Push migration
  - Pull migration.

s. 42

Processor afinity.
- When thread running on one processor the cache of that processer information of processor stored by that threads.
- Load balancing effect afinity. 2 types of affinity.
  - Soft afinity
  - hard afinity

s. 43

Date No

Real time cpu Scheduling.

- have 2 types of real time systems. 5.45
  - Soft real time
  - hard real time.

Event latency.

- the time took to event occurse to being serviced. 2 types of latencys. 5.46
  - Interrupt lateney
  - Dispatch latency.

Algorithm Evaluation.

- Deterministic modeling.
  - We have predfined work load and test it on each algorithm. (performance).
- Queueing model.
  - Compute the throughput, utilization, waiting estimate the cpu burst and arrival time disribution.
- little's formula.
  - Valid for any algorithm and for any arrival distribution.
- Simulation.
  - to get more accurate result we can use this.
- implimentation