Process / CPU Scheduling Policies

Scheduling Goals

Scheduling Goals

Performance

Throughput

Response Time

Throughput - Rate at which useful work is accomplished, e.g, search transaction on Google.

Many context switch – not good for high throughput, but may be necessary for quick response time.

Remember: It is not only CPU, efficient use of I/O, memory efficiently, a scheduler can maximize throughput.

Scheduling Mechanism

Non-preemptive vs. Preemptive

Process Parameters for Scheduling

Arrival Time (AT)

Service Time (ST) / Execution Time (ET) / Burst Time (BT)

Priority – Static vs. Dynamic

Input Parameters

Output Parameters

Finish Time (FT)

Waiting Time (WT) and Mean Waiting Time

Turnaround time (TAT) and Mean Turnaround Time

Process Schedule

Visualization

Gantt Charts (or Time scale diagram)

Uniprocessor Scheduling Policies

- Non-Preemptive
  - First Come First Serve (FCFS)
    - What if all processes arrive at the same time?
    - What if AT is not specified? (default: AT=(t=0))
  - Shortest Job First (SJF)
  - Priority (static)
    - How to use the assigned process priority? (default: low → high)
    - What if two processes have same priority? (use AT to resolve)
  - Highest Response Ratio Next (HRRN)
    - How to compute the Response Ratio (RR) [(WT+ST)/ST]
    - When to compute the RR?
    - What if two processes have same RR? (use AT to resolve)

Example 1

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Total No. Of processes = 5

Using FCFS CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters suitably labelled in it and summarize your computation using the given table.

Policy: FCFS						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule	A→B→C→D→E					

Example 2

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Using SJF CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters suitably labelled in it and summarize your computation using the given table.

Total No. Of processes = 5

Policy: SJF						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Example 3.1

The following table gives details of processes.

Process	Arrival Time	Service Time	Priority
A	0	3	2
B	0	6	3
C	0	4	5
D	0	5	4
E	0	2	1

Total No. Of processes = 5

Using static Priority CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters suitably labelled in it and summarize your computation using the given table.

Policy: PRIORITY						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Example 3.2

The following table gives details of processes.

Process	Arrival Time	Service Time	Priority
A	0	3	2
B	2	6	3
C	4	4	5
D	6	5	4
E	8	2	1

Using static Priority CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters suitably labelled in it and summarize your computation using the given table.

Total No. Of processes = 5

Policy: PRIORITY						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Example 4

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Using HRRN CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters suitably labelled in it and summarize your computation using the given table.

Total No. Of processes = 5

Policy: HRRN						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Example 5

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	1	5
C	3	2
D	9	5
E	12	5
F	14	4

Using FCFS, SJF and HRRN CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters suitably labelled in it and summarize your computation using the given table.

Policy: FCFS / SJF / HRRN							
Process	**A**	**B**	**C**	**D**	**E**	**F**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	FT(f)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	WT(f)	(WT(a)+WT(b)+WT(c) +WT(d)+WT(e)+WT(f)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	TAT(f)	(TAT(a)+TAT(b)+TAT(c) +TAT(d) + TAT(e) +TAT(f)) / Total number of processes
Schedule							

Example 6

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	1	5
C	3	2
D	9	5
E	12	5
F	14	4

Using FCFS, SJF and HRRN CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters suitably labelled in it and summarize your computation using the given table.

Consider context switching time of 1 unit.

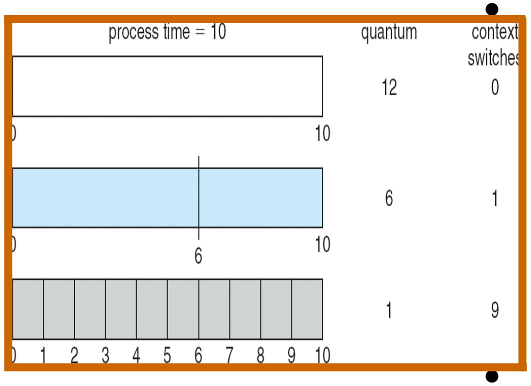
Policy: FCFS / SJF / HRRN							
Process	**A**	**B**	**C**	**D**	**E**	**F**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	FT(f)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	WT(f)	(WT(a)+WT(b)+WT(c) +WT(d)+WT(e)+WT(f)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	TAT(f)	(TAT(a)+TAT(b)+TAT(c) +TAT(d) + TAT(e) +TAT(f)) / Total number of processes
Schedule							

Uniprocessor Scheduling Policies

- Preemptive
  - Shortest Remaining Time Next (SRTN)
    - Uses ST like SJF, but more dynamically deciding between ST of new processes and remaining ST of existing processes.
    - When to check the above condition? (On every process arrival)
  - Round Robin (RR) scheduling
    - Every process gets a fixed and equal CPU time slot.
      - Deciding the value of time slot is a OS design issue.
      - Too high → long waits, too low → lot of switching
      - No process can get more than one time slot in one go.
        - » Exception: No other processes ready and waiting.
    - Every new process gets appended to existing ready processes who have been already waiting for their time slot.
    - Any process that uses a time slot lesser than what it was assigned does not affect other process(s). When it happens?
      - The next ready process gets a full new CPU time slot.

Uniprocessor Scheduling Policies

- Preemptive
  - Shortest Remaining Time Next (SRTN)
    - Uses ST like SJF, but more dynamically deciding between ST of new processes and remaining ST of existing processes.
    - When to check the above condition? (On every process arrival)
  - Round Robin (RR) scheduling



- Every process gets a fixed and equal CPU time slot.
  - Deciding the value of time slot is a OS design issue.
  - Too high → long waits, too low → lot of switching
  - No process can get more than one time slot in one go.
    - » Exception: No other processes ready and waiting.
- Every new process gets appended to existing ready processes who have been already waiting for their time slot.
- Any process that uses a time slot lesser than what it was assigned does not affect other process(s). When it happens?
  - The next ready process gets a full new CPU time slot.

Example 7

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Using SRTN CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters suitably labelled in it and summarize your computation using the given table.

Total No. Of processes = 5

Policy: SRTN						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Example 8

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Using RR (t=2) CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters suitably labelled in it and summarize your computation using the given table.

Total No. Of processes = 5

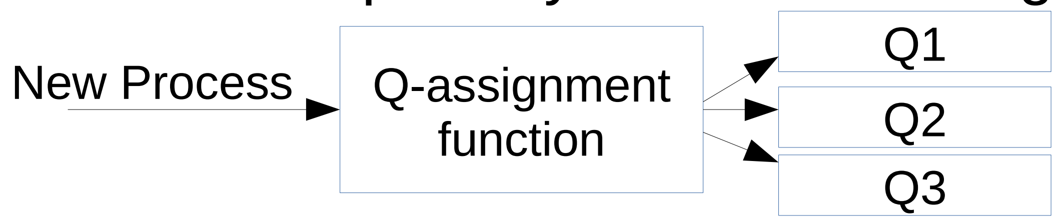
Policy: RR (t=2)						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Multi-level Queue (MLQ) Scheduling

- Multi-level queue scheduling is used when processes can be classified into groups.
- Multiple queues (hierarchy of queues).
- Each queue has its scheduling function. In that sense, MLQ is a hybrid scheduling policy.
- A process placed in one of the queues continues to remain in that queue till it terminates.
  - In other words, it follows static queue assignment for every process.
- Scheduler scans all queues based on queue ordering.
- Examples scenarios – queue ordering left to right.
  - 3 queues : RR (q=2), RR(q=3), RR(q=5)
  - 4 queues : SPN, SRTN, HRRN, FCFS.
- Two important questions: process queue assignment and queue processing.

Multi-level Queue (MLQ) Scheduling

- Q-assignment function
  - It is possible to place a process in one or more queues.
  - Decided based on a criteria (for e.g., execution time, memory usage, ratio of CPU and I/O usage) to assign a queue.
    - For e.g., FCFS(ST>10), SPN(ST<5), SRTN(5<ST<10)
  - Conflict resolution on Q-assignment in case of multiple options – a process is placed in higher in the hierarchy as it will be processed first.
  - Adaptive behavior – Advanced scheduling policy
    - A different queue ordering can be considered, based on history of scheduling in MLQ.
    - This will give a different schedule than obtained previously for a given set of processes - suggesting multiple ways to schedule a given set of processes.



Multi-level Queue (MLQ) Scheduling

- How these multiple queues are processed?
- Queue at level N is processed, provided all (N-1) queues have already been processed (i.e., they are empty).
    - Unless the first queue is empty, the second queue will not be considered in scheduling other processes.
    - In case of 3 queues, queue 2 not processed until queue 1 gets empty, and queue 3 will not be processed until queues 1 and 2 both are already processed.
    - Further, if a process from queue 3 is utilizing the CPU, and a new process arrives in queue 1 or 2 or both, the current process completes its CPU usage and than process from queue 1 or 2 is assigned the CPU.
        - **Q-check**: What it means that on every completion of CPU usage at level N, (N-1) queues are checked before the CPU could again be assigned to a process from queue N.

Example 9

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Using MLQ (3 Qs, RR(t=1,2,3)) CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Q-assignment (Q1:ST<=2, Q2:2<ST<=5, Q3>=6)

Construct a Gantt chart to show all of the above parameters along with the state of all 3 queues at every stage.

Total No. Of processes = 5

Policy: MLQ (3Qs, RR(t=1,2,3))						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Example 10

Consider the following processes. Construct the process execution schedule using SRTN and RR (time slot=3 units) and compute the necessary output parameters.

Process	Arrival Time	Service Time
A	0	5
B	2	2
C	4	10
D	6	4

Policy: SRTN / RR (t=3)						
Process	**A**	**B**	**C**	**D**		**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)		
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)		(WT(a)+WT(b)+WT(c) +WT(d)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)		(TAT(a)+TAT(b)+TAT(c) +TAT(d)) / Total number of processes
Schedule						

Example 11

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	3	4
D	4	2
E	5	5

Using RR (t=4) CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show the all of the above parameters suitably labelled in it and summarize your computation using the given table.

Total No. Of processes = 5

Policy: RR (t=4)						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Multi-level Feedback Queue (MLFQ) Scheduling

- The overhead of MLQ scheduling about classifying the processes using the Q-assignment function is resolved in MLFQ.
- Multiple queues (hierarchy of queues), but every process passes through the first and subsequent queues depending on its ST.
- Like MLQ, each queue in MLFQ also has its scheduling function.
- A process is always placed in the first queue, and moves down the other queues as it progresses (indicating decreased priority)
  - In other words, it follows dynamically changing the queue (and its priority) for every process, if it requires more CPU time.
- Like MLQ, scheduler scans all queues based on queue ordering.
- Examples scenarios – queue ordering left to right.
  - 3 queues : RR (q=2) → RR(q=3) → RR(q=5) (P1 (ST=2), P2(ST=6))
  - 4 queues : SPN → SRTN → FCFS (Time slot with Non-preemptive policy)
- How are queues processed? (similar to MLQ, but with Feedback)

Example 12

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Using MLFQ (3 Qs, RR(t=1,2,3)) CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Q-assignment (Q1:ST<=2, Q2:2<ST<=5, Q3>=6) – This is not needed in MLFQ.

Construct a Gantt chart to show all of the above parameters along with the state of all 3 queues at every stage.

Total No. Of processes = 5

Policy: MLFQ (3Qs, RR(t=1,2,3))						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Example 13

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Using MLFQ (3 Qs, RR(t=2^n, n=0,1,2)) CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters along with the state of all 3 queues at every stage.

Total No. Of processes = 5

Policy: MLFQ (3Qs, RR(t=1,2,4))						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						

Example 14

The following table gives details of processes.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Using MLFQ (3 Qs, RR(t=1),SRT(t=2), FCFS(t=3)) CPU scheduling policy, compute the waiting time, finish time, turnaround time, and schedule for the processes listed in the above table.

Construct a Gantt chart to show all of the above parameters along with the state of all 3 queues at every stage.

Total No. Of processes = 5

Policy: MLFQ (3Qs, RR(t=1),SRT(t=2),FCFS(t=3))						
Process	**A**	**B**	**C**	**D**	**E**	**Average**
Finish Time (FT)	FT(a)	FT(b)	FT(c)	FT(d)	FT(e)	
Wait Time (WT)	WT(a)	WT(b)	WT(c)	WT(d)	WT(e)	(WT(a)+WT(b)+WT(c)+WT(d)+WT(e)) / Total number of processes
Turnaround Time (TAT)	TAT(a)	TAT(b)	TAT(c)	TAT(d)	TAT(e)	(TAT(a)+TAT(b)+TAT(c)+TAT(d) + TAT(e)) / Total number of processes
Schedule						