PES UNIVERSITY ONLINE

OPERATING SYSTEMS

Unit1_Unit2_Unit3: Revision Class #1

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

Revision Class Outline - Student Queries

- How valid is the diagram which shows the stack and heap expanding towards each other given that they can be on different segments ?
- Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?
- Sir, can you explain how the CPU burst prediction is done in detail ?
- Comparative study on various CPU scheduling strategies - like advantages and disadvantages

Revision Class Outline - Student Queries

- Revision on Real time scheduling sir
- Revision for Round Robin Scheduling
- Recursive Mutexes, Semaphores, Spinlocks
- Revision of Banker's safety algorithm for detecting deadlocks

Revision Class Outline - Student Queries

- **Page replacement algorithm in general**
- **Demand paging performance calculation**
- **Inverted page table**

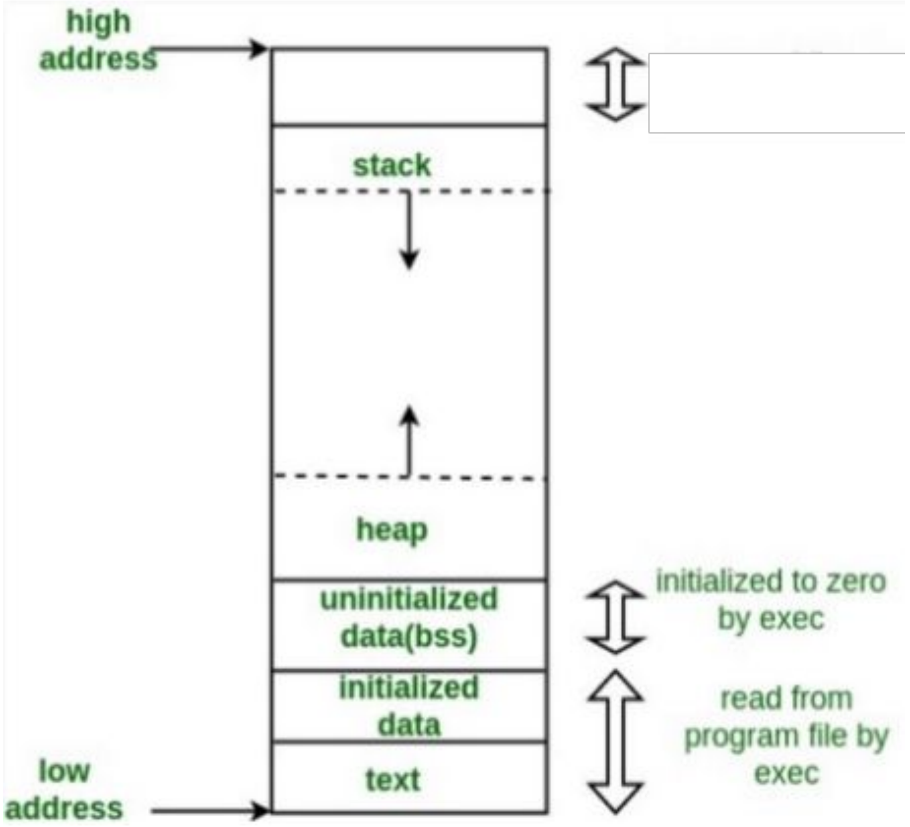
Revision Class Outline - Student Queries

- sir can you explain the difference between long term and short term scheduler
- what is the difference between device driver and device controller, how are they placed inside the system?
- sir explanation for mutex and semaphores
- Sir can you revise unit-3 case study and intel IA-32 architecture?
- sir explain the concept of virtual memory

Revision Class Outline - Student Queries

- sir can you do an example question for calculating waiting time in round robin scheduler?
- Sir can you give few example sums for scheduling algorithms with different arrival times?
-

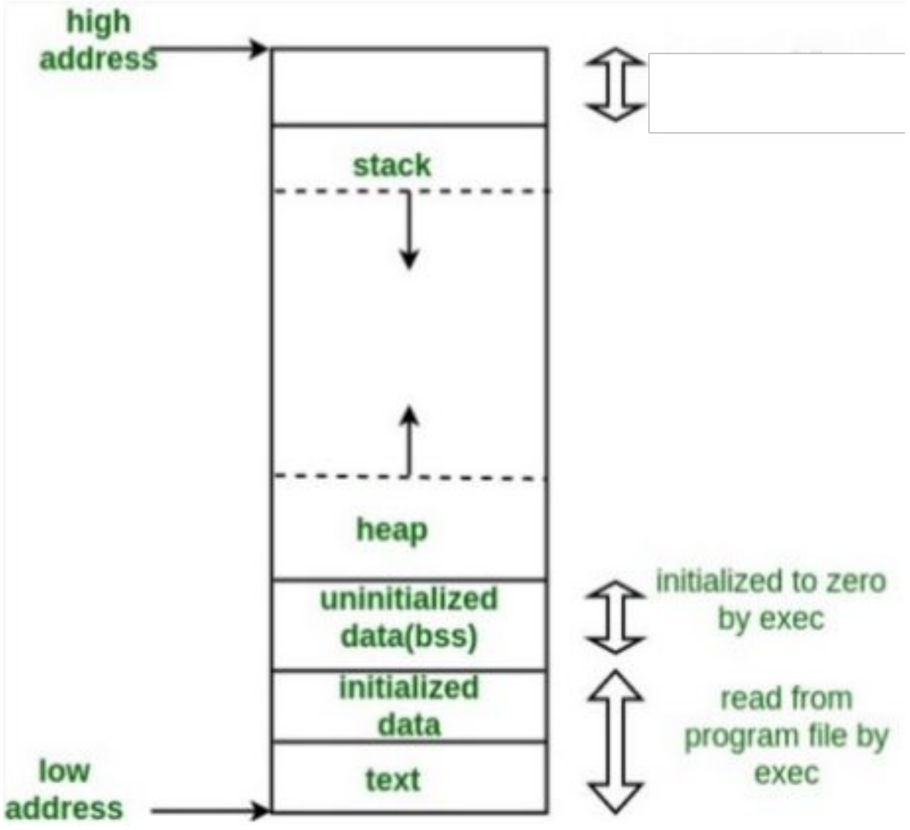
How valid is the diagram which shows the stack and heap expanding towards each other given that they can be on different segments ?



Stack

- The stack is a segment of memory where data like your local variables and function calls get added and/or removed in a last-in-first-out (LIFO) manner.
- When you compile for example a C program, the compiler enters through the main function and a stack frame is created on the stack.
- A frame, also known as an activation record is the collection of all data on the stack associated with one subprogram call.
- The main function and all the local variables are stored in an initial frame.

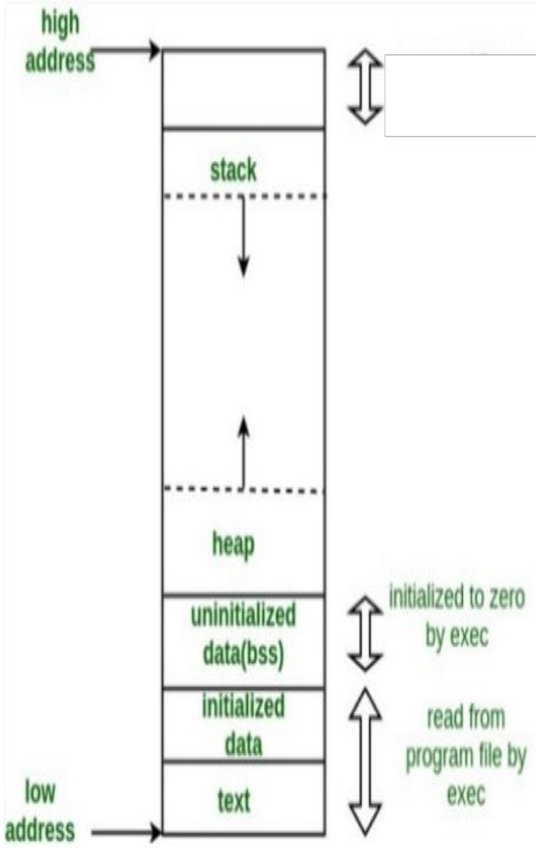
How valid is the diagram which shows the stack and heap expanding towards each other given that they can be on different segments ?



Heap

- The heap is the segment of memory that is not set to a constant size before compilation and can be controlled dynamically by the programmer.
- Think of the heap as a “free pool” of memory you can use when running your application.
- The size of the heap for an application is determined by the physical constraints of the RAM (Random access memory) and is generally much larger in size than the stack.

How valid is the diagram which shows the stack and heap expanding towards each other given that they can be on different segments ?



Stack v/s Heap

Stack

- Very Fast access
- Don't have to explicitly deallocate variables
- Space is managed efficiently by usually by Hardware Architecture and OS combined.
- Memory will not become fragmented
- Local variables only
- Limit on stack size (OS-dependent)
- Variables cannot be resized

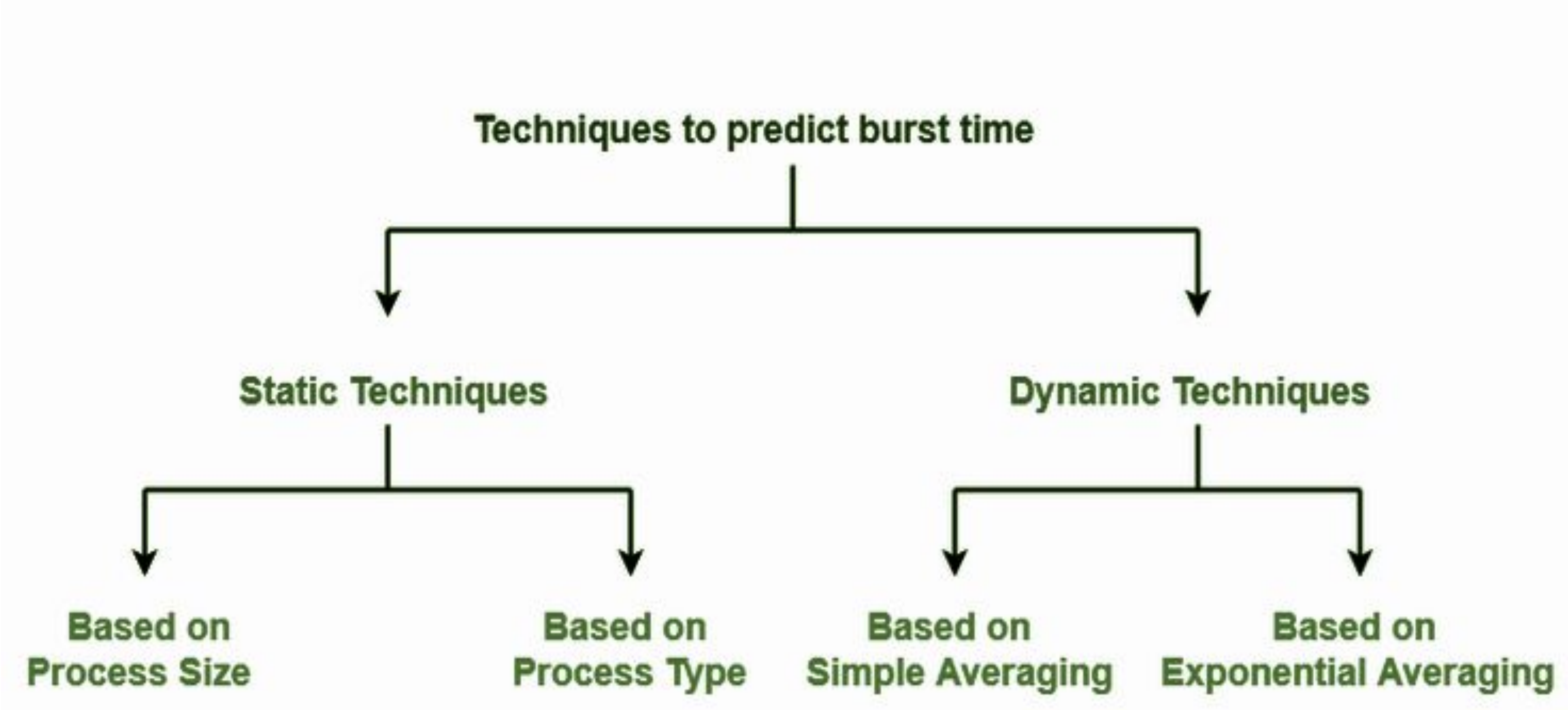
Heap

- Variables can be accessed globally
- No limit on memory size
- Relatively slower access
- No guaranteed efficient use of space, memory may become fragmented over time as blocks of memory are allocated, then freed
- The programmer must manage memory of allocating and freeing
- Variables can be resized using realloc()

Revision Class Outline - Student Queries

- Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?

- Sir, can you explain how the CPU burst prediction is done in detail ?



Revision Class Outline - Student Queries

- **Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?**

- **Sir, can you explain how the CPU burst prediction is done in detail ?**

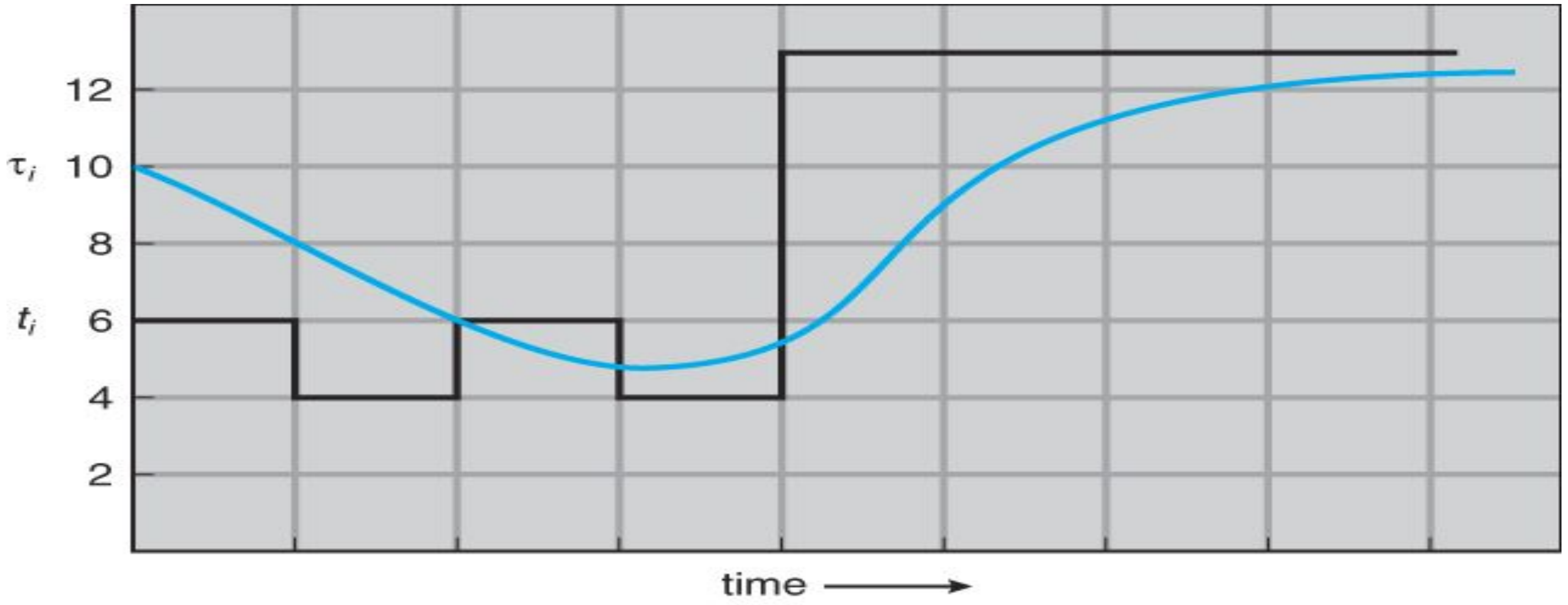
- Given n processes P_1, P_2, … , P_n and burst time of each process P_i as t_i. Then, predicted burst time for process P_{n+1} is given as
    - where
        - α is called smoothening factor (0<= α <=1)
        - tn = actual burst time of process Pn
        - Tn = Predicted burst time for process Pn

$$T_{n+1} = \alpha t_n + (1 - \alpha) T_n$$

Revision Class Outline - Student Queries

- **Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?**

- **Sir, can you explain how the CPU burst prediction is done in detail ?**



CPU burst (t_i)		6	4	6	4	13	13	13	. . .
"guess" (τ_i)	10	8	6	6	5	9	11	12	. . .

- Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?
- Sir, can you explain how the CPU burst prediction is done in detail ?
- Calculate the predicted burst time using exponential averaging for the sixth process if the predicted burst time for the first process is 9 units and actual burst time of the first five processes is 2, 10, 8, 9, 9 units respectively. Given α = 0.75.

$$T_{n+1} = \alpha t_n + (1 - \alpha) T_n$$

P	9
A	2

n	0					
n+1	1					

- Predicted burst time for 1st process = 9 units
- Actual burst time of the first four processes = 2,10, 8, 9,9
- α = 0.75

P - Predicted A - Actual	
Initial Prediction P	9

Revision Class Outline - Student Queries

- **Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?**

- **Sir, can you explain how the CPU burst prediction is done in detail ?**

- Calculate the predicted burst time using exponential averaging for the sixth process if the predicted burst time for the first process is 9 units and actual burst time of the first five processes is 2, 10, 8, 9, 9 units respectively. Given α = 0.75.

$$T_{n+1} = \alpha\, t_n + (1 - \alpha)\, T_n$$

P	**9**	**3.75**
A	**2**	

n	**0**	**1**				
n+1	**1**	**2**				

Predicted burst time for 2^{nd} process

= α x Actual burst time of 1^{st} process + (1-α) x Predicted burst time for 1^{st} process

= 0.75 x 2 + (1-0.75) x 9

= 1.5 + 0.25*9

= 3.75

Revision Class Outline - Student Queries

- **Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?**

- **Sir, can you explain how the CPU burst prediction is done in detail ?**

- Calculate the predicted burst time using exponential averaging for the sixth process if the predicted burst time for the first process is 9 units and actual burst time of the first five processes is 2, 10, 8, 9, 9 units respectively. Given α = 0.75.

$$T_{n+1} = \alpha t_n + (1 - \alpha) T_n$$

P	**9**	**3.75**	**8.437**
A	**2**	**10**	

n	**0**	**1**	**2**			
n+1	**1**	**2**	**3**			

Predicted burst time for 3rd process

= α x Actual burst time of 2nd process + (1-α) x Predicted burst time for 2nd process

= 0.75 *10 + 0.25 *3.75

= 8.437

Revision Class Outline - Student Queries

- **Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?**

- **Sir, can you explain how the CPU burst prediction is done in detail ?**

- Calculate the predicted burst time using exponential averaging for the sixth process if the predicted burst time for the first process is 9 units and actual burst time of the first five processes is 2, 10, 8, 9, 9 units respectively. Given α = 0.75.

$$T_{n+1} = \alpha t_n + (1 - \alpha) T_n$$

P	**9**	**3.75**	**8.437**	**8.109**
A	**2**	**10**	**8**	

n	**0**	**1**	**2**	**3**		
n+1	**1**	**2**	**3**	**4**		

Predicted burst time for 4th process

= α x Actual burst time of 3rd process + (1-α) x Predicted burst time for 3rd process

= 0.75 * 8 + .25 * 8.437

= 8.109

Revision Class Outline - Student Queries

- **Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?**

- **Sir, can you explain how the CPU burst prediction is done in detail ?**

- Calculate the predicted burst time using exponential averaging for the sixth process if the predicted burst time for the first process is 9 units and actual burst time of the first five processes is 2, 10, 8, 9, 9 units respectively. Given α = 0.75.

$$T_{n+1} = \alpha t_n + (1 - \alpha) T_n$$

P	9	3.75	8.437	8.109	8.777
A	2	10	8	9	

n	0	1	2	3	4	
n+1	1	2	3	4	5	

Predicted burst time for 5th process

= α x Actual burst time of 4th process + (1-α) x Predicted burst time for 4th process

= 0.75 * 9 + 0.25 * 8.109

= 8.777

- **Sir, in the textbook for shortest job first/next, they have spoken about a formula to find the CPU burst time of the next process (exponential average), could you explain that ?**

- **Sir, can you explain how the CPU burst prediction is done in detail ?**

- Calculate the predicted burst time using exponential averaging for the sixth process if the predicted burst time for the first process is 9 units and actual burst time of the first five processes is 2, 10, 8, 9, 9 units respectively. Given α = 0.75.

$$T_{n+1} = \alpha t_n + (1 - \alpha) T_n$$

P	9	3.75	8.437	8.109	8.777	8.944
A	2	10	8	9	9	

n	**0**	**1**	**2**	**3**	**5**	
n+1	**1**	**2**	**3**	**4**	**6**	

Predicted burst time for 6^{th} process

= α x Actual burst time of 5^{th} process + (1-α) x Predicted burst time for 5^{th} process

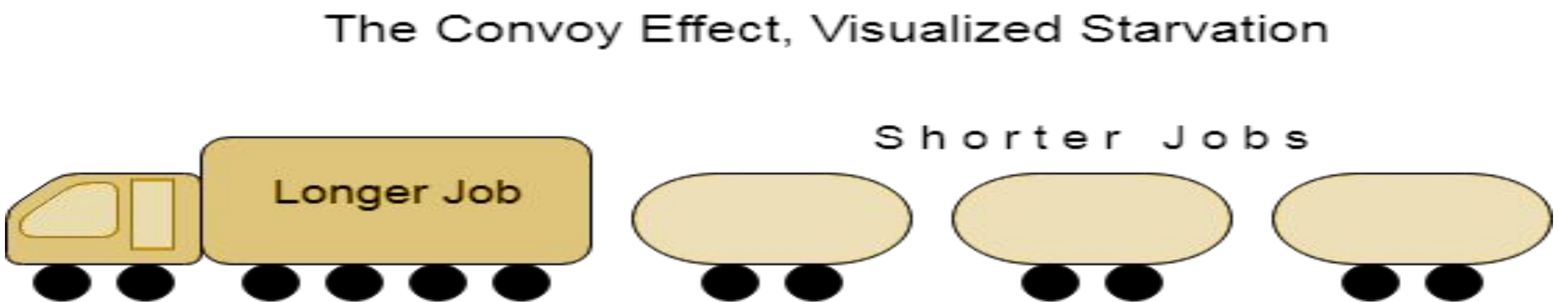
= 0.75 * 9 + 0.25 * 8.777

= 8.944

Revision Class Outline - Student Queries

- **Comparative study on various CPU scheduling strategies - like advantages and disadvantages**

Scheduling Algorithm	Advantages	Disadvantages
FCFS	• It is easy to implement and Understand	• The priority of the processes for execution by the CPU does not matter as it is a non-preemptive scheduling process • Average waiting time or AWT is high i.e. average waiting time is not optimal • There is no possibility of parallel utilization of resources, which give result to **Convoy Effect**
Convoy Effect:		



Revision Class Outline - Student Queries

PES UNIVERSITY ONLINE

- **Comparative study on various CPU scheduling strategies - like advantages and disadvantages**

Scheduling Algorithm	Advantages	Disadvantages
SJF	• It is optimal for minimizing the queuing time. • It is easy to implement in batch systems as the required CPU time is known in advance. • It's average waiting time, or AWT is minimum amongst all the scheduling algorithms.	• It is challenging to implement in the interactive system as required CPU time is not known in advance practically • The process which has shortest burst time will have to wait so that the current process can finish its performance based on respective arrival time • The reason behind this is non-preemptive mode where current process is not halted in between on arrival of shortest burst process. • This brings the starvation problem which is solved by the following the process of implementing **ageing**.

- **Comparative study on various CPU scheduling strategies - like advantages and disadvantages**

Scheduling Algorithm	Advantages	Disadvantages
SRTF / SJN	• Generally, it is used in batch system preference to the shortest jobs is given • The processes which have the shortest burst time will run fast as compare to the other one. • Increased throughput because a lot of shorter processes gets executed first	• It is challenging to implement in the interactive system as required CPU time is not known in advance practically • The process which has long burst time will have to wait for long time for their execution because the processes with short burst time are being executed by the CPU. • This brings the starvation problem which is solved by the following the process of implementing **ageing**.

- **Comparative study on various CPU scheduling strategies - like advantages and disadvantages**

Scheduling Algorithm	Advantages	Disadvantages
RR	• It is simple. • It is easy to implement. • It deals with all process without any priority as no priority is given in this type of scheduling. • In this, all jobs get easily allocated to the CPU for their execution. • No Convoy Effect • Round robin scheduling algorithm does not depend upon burst time. So we can easily implement the round robin scheduling algorithm on the system.	• As we know that the time quantum is the main requirement of the round robin scheduling algorithm to execute processes. So, deciding a perfect time quantum for scheduling the processes is a very difficult and not an optimal task. • Higher the time quantum, higher the response time of the system. • Lower the time quantum, higher the context switching overhead.

- **Comparative study on various CPU scheduling strategies - like advantages and disadvantages**

Scheduling Algorithm	Advantages	Disadvantages
Priority Scheduling	• The priority of process is selected on the basis of memory requirement, user preference or the requirement of time. • Processes are executed on the basis of priority. So high priority does not need to wait for long which saves time. • It is easy to use. • It is a user friendly algorithm. • Simple to understand. • It has reasonable support for priority.	• The major disadvantage of priority scheduling is the process of indefinite blocking or starvation. This problem appears when a process is ready to be executed but it has to wait for the long time for execution by CPU because other high priority processes are executed by the CPU. • The problem of starvation can be solved by aging. Aging is a technique in which the system gradually increases the priority of those processes which are waiting in the system from a long time for their execution. • In case if we have the processes which have the same priority, then we have to make use of preferable size, finally FCFS in case of contention • If the system session eventually crash, then all the processes having low priority which are not finished yet, also get lost.

- **Comparative study on various CPU scheduling strategies - like advantages and disadvantages**

Scheduling Algorithm	Advantages	Disadvantages
Multilevel Queue scheduling algorithm	• We can apply different scheduling algorithms to different processes. • It has low scheduling overhead. • There are many processes which we are not able to put them in the one single queue which is solved by this scheduling as we can now put them in different queues.	• Its main drawback is the starvation problem which exists for the lowest level of processes which means the most inferior priority process has to wait for a long time as high priority process takes much time to execute. • The process doesn’t move from one queue to another queue. • It is inflexible.

- **Comparative study on various CPU scheduling strategies - like advantages and disadvantages**

Scheduling Algorithm	Advantages	Disadvantages
Multilevel Feedback Queue scheduling algorithm	• It prevents starvation. • Low scheduling overhead • It speeds up the flow of task execution. • It improves the overall performance of the system.	• It requires some means of selecting values for all the parameters to define the best scheduler, thus it is also the most complex.

OPERATING SYSTEMS

Revision Class Outline - Student Queries

- **Revision of Round Robin Scheduling**

Process	Arrival Time	Burst Time
P0	0	17
P1	0	8
P2	4	4
P3	1	6
P4	2	9

Ready Queue
P1=>8

Time Quantum: 6 Units

At Arrival time 0, there is a conflict between process P0 and P1. Selecting shorter job is prefered at the time of decision making. So Choose P1 in this case with 8 CPU Burst

Round Robin Scheduling				
P1				

0 to 6

• Revision of Round Robin Scheduling

Process	Arrival Time	Burst Time
P0	0	17
P1	0	8-6=> 2
P2	4	4
P3	1	6
P4	2	9

Ready Queue
P3=>6
P4=>9
P0=>17
P1=>2

Time Quantum: 6 Units

At Arrival time 6, there is a conflict between process P0, P2, P3. Selecting shorter job is prefered at the time of decision making. So Choose P2 in this case with 4 CPU Burst, P3, P4, P0, P1 is put the ready queue in that order

Round Robin Scheduling

P1	P2			
0 to 6	6 to 10			

- **Revision of Round Robin Scheduling**

Process	Arrival Time	Burst Time
P0	0	17
P1	0	8-6=> 2
P2	4	4-4=>0
P3	1	6
P4	2	9

Ready Queue
P3=>6
P4=>9
P0=>17
P1=>2

Time Quantum: 6 Units

P3 is selected from the Queue and is scheduled. Since it requirement was <= Time Quantum; It exited Normally

Round Robin Scheduling				
P1	P2	P3		
0 to 6	6 to 10	10 to 16		

- ### Revision of Round Robin Scheduling

Process	Arrival Time	Burst Time
P0	0	17
P1	0	8-6=> 2
P2	4	4-4=>0
P3	1	6-6=>0
P4	2	9-6=>3

Ready Queue
P4=>9
P0=>17
P1=>2

Time Quantum: 6 Units

P4 is selected from the Queue and is scheduled. Since it requirement was > Time Quantum; After allotting the time quantum, it is put back in the queue behind P1

Round Robin Scheduling

P1	P2	P3	P4	
0 to 6	6 to 10	10 to 16	16 to 22	

Revision Class Outline - Student Queries

- **Revision of Round Robin Scheduling**

Process	Arrival Time	Burst Time
P0	0	17-6=>11
P1	0	8-6=> 2
P2	4	4-4=>0
P3	1	6-6=>0
P4	2	9-6=>3

Ready Queue
P0=>17
P1=>2
P4=>3

Time Quantum: 6 Units

P0 is selected from the Queue and is scheduled. Since it requirement was > Time Quantum; After allotting the time quantum, it is put back in the queue behind P4

Round Robin Scheduling

P1	P2	P3	P4	P0
0 to 6	6 to 10	10 to 16	16 to 22	16 to 22

PES UNIVERSITY ONLINE

• Revision of Round Robin Scheduling

Process	Arrival Time	Burst Time
P0	0	17-6=>11
P1	0	8-6=> 2-2=>0
P2	4	4-4=>0
P3	1	6-6=>0
P4	2	9-6=>3

Ready Queue
P1=>2
P4=>3
P0=>11

Time Quantum: 6 Units

P3 is selected from the Queue and is scheduled. Since it requirement now is <= Time Quantum; It exited Normally

Round Robin Scheduling

P1	P2	P3	P4	P0
0 to 6	6 to 10	10 to 16	16 to 22	16 to 22

- **Revision of Round Robin Scheduling**

Process	Arrival Time	Burst Time
P0	0	17-6=>11
P1	0	8-6=> 2-2=>0
P2	4	4-4=>0
P3	1	6-6=>0
P4	2	9-6=>3

Ready Queue
P0=>5

Time Quantum: 6 Units

P3 is selected from the Queue and is scheduled. Since it requirement now is <= Time Quantum; It exited Normally

Round Robin Scheduling

P1	P2	P3	P4	P0
0 to 6	6 to 10	10 to 16	16 to 22	16 to 22

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com