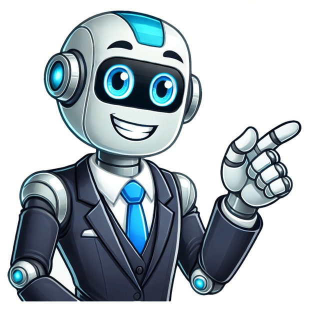

CPU scheduling is a process used by the operating system to decide which task or process gets to use the CPU at a particular time. This is important because a CPU can only handle one task at a time, but there are usually many tasks that need to be processed. The following are different purposes of a CPU scheduling time.Maximize the CPU utilizationMinimize the response and waiting time of the process.What is the Need for a CPU Scheduling Algorithm?CPU scheduling is the process of deciding which process will own the CPU to use while another process is suspended. The main function of CPU scheduling is to ensure that whenever the CPU remains idle, the OS has at least selected one of the processes available in the ready-to-use line.In Multiprogramming, if the long-term scheduler selects multiple I/O binding processes then most of the time, the CPU remains idle. The function of an effective program is to improve resource utilization.Terminologies Used in CPU SchedulingArrival Time: The time at which the process arrives in the ready queue.Completion Time: The time at which the process completes its execution.Burst Time: Time required by a process for CPU execution.Turn Around Time: Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival TimeWaiting Time(W.T): Time Difference between turn around time and burst time.Waiting Time = Turn Around Time – Burst TimeThings to Take Care While Designing a CPU Scheduling AlgorithmDifferent CPU Scheduling algorithms have different structures and the choice of a particular algorithm depends on a variety of factors. CPU Utilization: The main purpose of any CPU algorithm is to keep the CPU as busy as possible. Theoretically, CPU usage can range from 0 to 100 but in a real-time system, it varies from 40 to 90 percent depending on the system load.Throughput: The average CPU performance is the number of processes performed and completed during each unit. This is called throughput. The output may vary depending on the length or duration of the processes.Turn Round Time: For a particular process, the important conditions are how long it takes to perform that process. The time elapsed from the time of process delivery to the time of completion is known as the conversion time. Conversion time is the amount of time spent waiting for memory access, waiting in line, using CPU and waiting for I/O.Waiting Time: The Scheduling algorithm does not affect the time required to complete the process once it has started performing. It only affects the waiting time of the process i.e. the time spent in the waiting process in the ready queue.Response Time: In a collaborative system, turn around time is not the best option. The process may produce something early and continue to computing the new results while the previous results are released to the user. Therefore another method is the time taken in the submission of the application process until the first response is issued. This measure is called response time.Different Types of CPU Scheduling AlgorithmsThere are mainly two types of scheduling methods:Preemptive Scheduling: Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to the ready state.Non-Preemptive Scheduling: Non-Preemptive scheduling is used when a process terminates , or when a process switches from running state to waiting state.CPU SchedulingPlease refer Preemptive vs Non-Preemptive Scheduling for details.CPU Scheduling AlgorithmsLet us now learn about these CPU scheduling algorithms in operating systems one by one:Comparison of CPU Scheduling AlgorithmsHere is a brief comparison between different CPU scheduling algorithms:Algorithm Allocation Complexity Average waiting time (AWT) Preemption Starvation PerformanceFCFSAccording to the arrival time of the processes, the CPU is allocated. Simple and easy to implement Large. No No Slow performanceSJF Based on the lowest CPU burst time (BT). More complex than FCFSSmaller than FCFS No Yes Minimum Average Waiting TimeSRTFSame as SJF the allocation of the CPU is based on the lowest CPU burst time (BT). But it is preemptive.More complex than FCFS Depending on some measures e.g., arrival time, process size, etcYesYes The preference is given to the short jobsRRAccording to the order of the process arrives with fixed time quantum (TQ) The complexity depends on Time Quantum sizeLarge as compared to SJF and Priority scheduling. Yes No Each process has given a fairly fixed timePriority Pre-emptive According to the priority. The bigger priority task executes first This type is less complex Smaller than FCFSYes YesWell performance but contain a starvation problemPriority non-preemptive According to the priority with monitoring the new incoming higher priority jobsThis type is less complex than Priority preemptivePreemptive Smaller than FCFS No Yes Most beneficial with batch systemsMLQ According to the process that resides in the bigger queue priority More complex than the priority scheduling algorithmsSmaller than FCFS No Yes Good performance but contain a starvation problemMFLQ According to the process of a bigger priority queue. It is the most Complex but its complexity rate depends on the TQ size Smaller than all scheduling types in many cases No No Good performanceQuestions for PracticeQuestion: Which of the following is false about SJF? S1: It causes minimum average waiting time S2: It can cause starvation (A) Only S1(B) Only S2 (C) Both S1 and S2 (D) Neither S1 nor S2 Answer: (D) S1 is true SJF will always give minimum average waiting time. S2 is true SJF can cause starvation. Question: Consider the following table of arrival time and burst time for three processes P0, P1 and P2.ProcessArrival timeBurst TimeP00 ms 9 msP11 ms4 msP22 ms9 msThe pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of processes. What is the average waiting time for the three processes? (A) 5.0 ms (B) 4.33 ms (C) 6.33 (D) 7.33 Solution: (A) Process P0 is allocated processor at 0 ms as there is no other process in the ready queue. P0 is preempted after 1 ms as P1 arrives at 1 ms and burst time for P1 is less than remaining time of P0. P1 runs for 4ms. P2 arrived at 2 ms but P1 continued as burst time of P2 is longer than P1. After P1 completes, P0 is scheduled again as the remaining time for P0 is less than the burst time of P2. P0 waits for 4 ms, P1 waits for 0 ms and P2 waits for 11 ms. So average waiting time is (0+4+11)/3 = 5. Question: Consider the following set of processes, with the arrival times and the CPU-burst times given in milliseconds.ProcessArrival timeBurst TimeP10 ms 5 msP21 ms3 msP32 ms3 msP44 ms1 msWhat is the average turnaround time for these processes with the preemptive Shortest Remaining Processing Time First algorithm ? (A) 5.50 (B) 5.75 (C) 6.00 (D) 6.25 Answer: (A) Solution: The following is Gantt Chart of executionTurn Around Time = Completion Time – Arrival Time Avg Turn Around Time = (12 + 3 + 6+ 1)/4 = 5.50 Question: An operating system uses the Shortest Remaining Time First (SRTF) process scheduling algorithm. Consider the arrival times and execution times for the following processes:ProcessBurst TimeArrival TimeP120 ms 0 msP225 ms15 msP310 ms30 msP415 ms45 msWhat is the total waiting time for process P2? (A) 5 (B) 15 (C) 40 (D) 55 Answer (B) Solution: At time 0, P1 is the only process, P1 runs for 15 time units. At time 15, P2 arrives, but P1 has the shortest remaining time. So P1 continues for 5 more time units. At time 20, P2 is the only process. So it runs for 10 time units At time 30, P3 is the shortest remaining time process. So it runs for 10 time units At time 40, P2 runs as it is the only process. P2 runs for 5 time units. At time 45, P3 arrives, but P2 has the shortest remaining time. So P2 continues for 10 more time units. P2 completes its execution at time 55.Total waiting time for P2 = Completion time - (Arrival time + Execution time)= 55 - (15 + 25)= 15 In the realm of Modern Operating Systems (OS), CPU scheduling stands as a cornerstone. It's the mechanism that determines which processes get access to the CPU, how long they can use it, and in what order they execute. Efficient CPU scheduling is paramount to maximizing system performance, responsiveness, and resource utilization. In this comprehensive guide, we'll explore various CPU scheduling algorithms, their characteristics, and their impact on OS performance. Understanding CPU Scheduling At its core, CPU scheduling is about managing the execution of multiple processes in a multitasking environment. The OS must decide which process to run next, given the limited resources of the CPU. Effective CPU scheduling aims to minimize waiting times, maximize throughput, and ensure fair allocation of the CPU among competing processes. We have a simple explanation to understand the Main functions of the Operating System. Common CPU Scheduling Algorithms First-Come, First-Served (FCFS) The First-Come, First-Served (FCFS) algorithm is the simplest CPU scheduling algorithm. It schedules processes in the order they arrive in the ready queue. While easy to implement, FCFS can lead to a phenomenon known as "convoy effect," where shorter processes wait for longer processes to complete, resulting in increased average waiting times. Shortest Job Next (SJN) or Shortest Job First (SJF) The Shortest Job Next (SJN) or Shortest Job First (SJF) algorithm selects the process with the shortest execution time next. This algorithm minimizes average waiting time and improves system throughput. However, it requires knowledge of the execution time of each process, which may not be available in a real-world scenario. Remember the processes in Operating System? Round Robin (RR) The Round Robin (RR) algorithm is a preemptive scheduling algorithm that allocates a fixed time slice, or quantum, to each process in a circular queue. If a process doesn't complete within its quantum, it's preempted, and the next process is scheduled. RR provides fair allocation of CPU time but may result in high context switch overheads and may not be optimal for all types of workloads. Priority Scheduling Priority Scheduling assigns a priority to each process based on factors such as process type, importance, or user-defined criteria. The highest priority process is executed first. While Priority Scheduling can improve system responsiveness and meet specific service level agreements (SLAs), it can also lead to starvation of lower priority processes if not implemented carefully. Feel free to understand System Calls in Operating Systems with our simple explanation. Evaluating Scheduling Algorithms When evaluating CPU scheduling algorithms, it's crucial to consider various performance metrics, including: CPU Utilization: The percentage of time the CPU is busy executing processes. Throughput: The number of processes completed per unit time. Waiting Time: The total time a process spends waiting in the ready queue. Turnaround Time: The total time taken from the submission of a process to its completion. Response Time: The time taken from the submission of a request until the first response is produced. Real-World Considerations In real-world scenarios, OS designers often employ hybrid or multi-level feedback queue scheduling algorithms that combine the strengths of multiple scheduling algorithms to optimize CPU utilization, minimize waiting times, and provide fair allocation of resources. Additionally, modern CPUs and OSes utilize advanced scheduling techniques, such as multithreading, multiprocessing, and load balancing, to further enhance performance and responsiveness. Conclusion CPU scheduling algorithms play a pivotal role in determining the performance and efficiency of Modern Operating Systems. By understanding the characteristics, advantages, and limitations of different scheduling algorithms, OS designers and administrators can make informed decisions to optimize system performance, enhance user experience, and ensure efficient resource utilization. As we continue to explore the intricacies of Modern Operating Systems in this series, we'll delve deeper into other critical aspects, such as I/O Device Management, Concurrency and Synchronization, Virtual Memory Management, and Network Operating Systems, to provide a comprehensive understanding of the principles, design considerations, and practical applications of operating systems in today's computing landscape. Stay tuned for our next post, where we will delve into the complexities of I/O Device Management, exploring the role of I/O devices in the OS, I/O operations, buffering, and interrupt handling, to further enhance your understanding of Modern Operating Systems. CPU scheduling is a crucial feature of operating systems that govern sharing processor time among the numerous tasks running on a computer. Hence, it is essential to ensure efficiency and fairness in executing processes. It also ensures that the system can fulfill its users' performance and responsiveness requirements. CPU scheduling comprises many essential concepts. In this tutorial, we'll discuss concepts central to CPU scheduling, including arrival, burst, completion, turnaround, waiting, and response time. By understanding these concepts and how they are used in different scheduling algorithms, we can gain a deeper understanding of how operating systems manage the execution of processes on a computer. 2. Key Concepts in CPU Scheduling In CPU Scheduling, the arrival time refers to the moment in time when a process enters the ready queue and is awaiting execution by the CPU. In other words, it is the point at which a process becomes eligible for scheduling. Many CPU scheduling algorithms consider arrival time when selecting the next process for execution. A scheduler, for example, may favor processes with earlier arrival timings over those with later arrival times to reduce the waiting time for a process in the ready queue. Hence, it can assist in ensuring the execution of processes efficiently. 2.2. Burst Time Burst time, also referred to as "execution time". It is the amount of CPU time the process requires to complete its execution. It is the amount of processing time required by a process to execute a specific task or unit of a job. Factors such as the task's complexity, code efficiency, and the system's resources determine the process's burst time. The burst time is also an essential factor in CPU scheduling. A scheduler, for example, may favor processes with shorter burst durations over those with longer burst times. This will reduce the time a process spends operating on the CPU. Hence, it can assist in ensuring that the system can make optimum use of the processor's resources. 2.3. Completion Time Completion time is when a process finishes execution and is no longer being processed by the CPU. It is the summation of the arrival, waiting, and burst times. Completion time is an essential metric in CPU scheduling, as it can help determine the efficiency of the scheduling algorithm. It is also helpful in determining the waiting time of a process. For example, a scheduling algorithm that consistently results in shorter completion times for processes is considered more efficient than one that consistently results in longer completion times. 2.4. Turnaround Time The time elapsed between the arrival of a process and its completion is known as turnaround time. That is, the duration it takes for a process to complete its execution and leave the system. Turnaround Time = Completion Time – Arrival Time A scheduling algorithm that regularly produces shorter turnaround times for processes is considered more efficient than one with longer turnaround times. 2.5. Waiting Time This is a process's duration in the ready queue before it begins executing. It helps assess how efficient the scheduling algorithm is. A scheduling method that consistently results in reduced wait times for processes, for example, is considered more efficient than one that regularly results in longer wait times. Waiting Time = Turnaround Time – Burst Time Furthermore, it helps to measure the efficiency of a scheduling algorithm. Also, it aids in determining a system's perceived responsiveness to user queries. A long wait time can contribute to a negative user experience. This is because consumers may view the system as slow to react to their requests. 2.6. Response Time Response time is the amount of time it takes for the CPU to respond to a request made by a process. It is the duration between the arrival of a process and the first time it runs. It is an essential parameter in CPU scheduling since it may assist in determining a system's perceived responsiveness to user requests. Response Time = Time it Started Executing – Arrival Time The number of processes waiting in the ready queue, the priority of the processes, and the features of the scheduling algorithm are all variables that might impact response time. For example, a scheduling algorithm that prioritizes processes with shorter burst times may result in quicker response times for those processes. 3. Example for Illustration To further illustrate this concept and how they are calculated, let's consider an example with four processes as shown in the table, including arrival time and burst time. Using the non-preemptive shortest-job-first algorithm, we can see how the processes are completed: Process Arrival Time Burst Time P1 3 3 P2 6 3 P3 0 4 P4 2 5 At time=0: P3 arrives and starts execution without waiting. Let's note that P3 is first attended to at time=0: At time=2: P4 arrives, and P3 continues executing. Hence, P4 waits in the queue: At time=3: P1 arrives, and P3 continues executing: At time=4: P3 completes execution. The burst time for P4 and P1 are compared. Hence P1 starts executing: At this point, we can calculate the Turnaround, Wait, and Response Time for P3: Completion Time (P3) = 4 Turnaround Time (P3) = 4 - 0 = 4 Wait Time (P3) = 4 - 4 = 0 Response Time (P3) = 0 - 0 = 0 At time=6: P2 arrives, and P1 is still executing: At time=7: P1 completes execution. The burst time for P4 and P2 are compared. Hence, P2 starts executing: Now, we can make calculations for P1: Completion Time (P1) = 7 Turnaround Time (P1) = 7 - 3 = 4 Wait Time (P1) = 4 - 3 = 1 Response Time (P1) = 4 - 3 = 1 At time=10: P2 completes execution, and only P4 is in the wait queue. Hence, P4 starts executing: At this point, we can make calculations for P2: Completion Time (P2) = 10 Turnaround Time (P2) = 10 - 6 = 4 Wait Time (P2) = 4 - 3 = 1 Response Time (P2) = 7 - 6 = 1 At time=15: P4 completes execution. Hence, we have the Gantt Chart: We can now make calculations for P4: Completion Time (P4) = 15 Turnaround Time (P4) = 15 - 2 = 13 Wait Time (P4) = 13 - 5 = 8 Response Time (P4) = 10 - 2 = 8 4. Conclusion This article discussed CPU scheduling concepts: arrival, burst, completion, turnaround, waiting, and response times. We also discussed how to calculate them, providing an example for illustration. By taking these elements into account, operating systems may efficiently balance the demands of various activities to enhance the system's overall performance and efficiency. CPU scheduling is a process used by the operating system to decide which task or process gets to use the CPU at a particular time. This is important because a CPU can only handle one task at a time, but there are usually many tasks that need to be processed. The following are different purposes of a CPU scheduling time.Maximize the CPU utilizationMinimize the response and waiting time of the process.What is the Need for a CPU Scheduling Algorithm?CPU scheduling is the process of deciding which process will own the CPU to use while another process is suspended. The main function of CPU scheduling is to ensure that whenever the CPU remains idle, the OS has at least selected one of the processes available in the ready-to-use line.In Multiprogramming, if the long-term scheduler selects multiple I/O binding processes then most of the time, the CPU remains idle. The function of an effective program is to improve resource utilization.Terminologies Used in CPU SchedulingArrival Time: The time at which the process arrives in the ready queue.Completion Time: The time at which the process completes its execution.Burst Time: Time required by a process for CPU execution.Turn Around Time: Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival TimeWaiting Time(W.T): Time Difference between turn around time and burst time.Waiting Time = Turn Around Time – Burst TimeThings to Take Care While Designing a CPU Scheduling AlgorithmDifferent CPU Scheduling algorithms have different structures and the choice of a particular algorithm depends on a variety of factors. CPU Utilization: The main purpose of any CPU algorithm is to keep the CPU as busy as possible. Theoretically, CPU usage can range from 0 to 100 but in a real-time system, it varies from 40 to 90 percent depending on the system load.Throughput: The average CPU performance is the number of processes performed and completed during each unit. This is called throughput. The output may vary depending on the length or duration of the processes.Turn Round Time: For a particular process, the important conditions are how long it takes to perform that process. The time elapsed from the time of process delivery to the time of completion is known as the conversion time. Conversion time is the amount of time spent waiting for memory access, waiting in line, using CPU and waiting for I/O.Waiting Time: The Scheduling algorithm does not affect the time required to complete the process once it has started performing. It only affects the waiting time of the process i.e. the time spent in the waiting process in the ready queue.Response Time: In a collaborative system, turn around time is not the best option. The process may produce something early and continue to computing the new results while the previous results are released to the user. Therefore another method is the time taken in the submission of the application process until the first response is issued. This measure is called response time.Different Types of CPU Scheduling AlgorithmsThere are mainly two types of scheduling methods:Preemptive Scheduling: Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to the ready state.Non-Preemptive Scheduling: Non-Preemptive scheduling is used when a process terminates , or when a process switches from running state to waiting state.CPU SchedulingPlease refer Preemptive vs Non-Preemptive Scheduling for details.CPU Scheduling AlgorithmsLet us now learn about these CPU scheduling algorithms in operating systems one by one:Comparison of CPU Scheduling AlgorithmsHere is a brief comparison between different CPU scheduling algorithms:Algorithm Allocation Complexity Average waiting time (AWT) Preemption Starvation PerformanceFCFSAccording to the arrival time of the processes, the CPU is allocated. Simple and easy to implement Large. No No Slow performanceSJF Based on the lowest CPU burst time (BT). More complex than FCFSSmaller than FCFS No Yes Minimum Average Waiting TimeSRTFSame as SJF the allocation of the CPU is based on the lowest CPU burst time (BT). But it is preemptive.More complex than FCFS Depending on some measures e.g., arrival time, process size, etcYesYes The preference is given to the short jobsRRAccording to the order of the process arrives with fixed time quantum (TQ) The complexity depends on Time Quantum sizeLarge as compared to SJF and Priority scheduling. Yes No Each process has given a fairly fixed timePriority Pre-emptive According to the priority. The bigger priority task executes first This type is less complex Smaller than FCFSYes YesWell performance but contain a starvation problemPriority non-preemptive According to the priority with monitoring the new incoming higher priority jobsThis type is less complex than Priority preemptivePreemptive Smaller than FCFS No Yes Most beneficial with batch systemsMLQ According to the process that resides in the bigger queue priority More complex than the priority scheduling algorithmsSmaller than FCFS No Yes Good performance but contain a starvation problemMFLQ According to the process of a bigger priority queue. It is the most Complex but its complexity rate depends on the TQ size Smaller than all scheduling types in many cases No No Good performanceQuestions for PracticeQuestion: Which of the following is false about SJF? S1: It causes minimum average waiting time S2: It can cause starvation (A) Only S1(B) Only S2 (C) Both S1 and S2 (D) Neither S1 nor S2 Answer: (D) S1 is true SJF will always give minimum average waiting time. S2 is true SJF can cause starvation. Question: Consider the following table of arrival time and burst time for three processes P0, P1 and P2.ProcessArrival timeBurst TimeP00 ms 9 msP11 ms4 msP22 ms9 msThe pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of processes. What is the average waiting time for the three processes? (A) 5.0 ms (B) 4.33 ms (C) 6.33 (D) 7.33 Solution: (A) Process P0 is allocated processor at 0 ms as there is no other process in the ready queue. P0 is preempted after 1 ms as P1 arrives at 1 ms and burst time for P1 is less than remaining time of P0. P1 runs for 4ms. P2 arrived at 2 ms but P1 continued as burst time of P2 is longer than P1. After P1 completes, P0 is scheduled again as the remaining time for P0 is less than the burst time of P2. P0 waits for 4 ms, P1 waits for 0 ms and P2 waits for 11 ms. So average waiting time is (0+4+11)/3 = 5. Question: Consider the following set of processes, with the arrival times and the CPU-burst times given in milliseconds.ProcessArrival timeBurst TimeP10 ms 5 msP21 ms3 msP32 ms3 msP44 ms1 msWhat is the average turnaround time for these processes with the preemptive Shortest Remaining Processing Time First algorithm ? (A) 5.50 (B) 5.75 (C) 6.00 (D) 6.25 Answer: (A) Solution: The following is Gantt Chart of executionTurn Around Time = Completion Time – Arrival Time Avg Turn Around Time = (12 + 3 + 6+ 1)/4 = 5.50 Question: An operating system uses the Shortest Remaining Time First (SRTF) process scheduling algorithm. Consider the arrival times and execution times for the following processes:ProcessBurst TimeArrival TimeP120 ms 0 msP225 ms15 msP310 ms30 msP415 ms45 msWhat is the total waiting time for process P2? (A) 5 (B) 15 (C) 40 (D) 55 Answer (B) Solution: At time 0, P1 is the only process, P1 runs for 15 time units. At time 15, P2 arrives, but P1 has the shortest remaining time. So P1 continues for 5 more time units. At time 20, P2 is the only process. So it runs for 10 time units At time 30, P3 is the shortest remaining time process. So it runs for 10 time units At time 40, P2 runs as it is the only process. P2 runs for 5 time units. At time 45, P3 arrives, but P2 has the shortest remaining time. So P2 continues for 10 more time units. P2 completes its execution at time 55.Total waiting time for P2 = Completion time - (Arrival time + Execution time)= 55 - (15 + 25)= 15 Feel free to contact our support at any time and with any issue. A challenging essay in just 6 hours? Our professionals can do it! More time, less money — be an early bird to find your best price. You can trust us! The security of customer information is our priority. You get fully edited plagiarism-free work tailored for you. With us, you will find an expert on your topic. If not, we will fix that. Give us as many details as possible so we can meet your expectations. Choose the payment method that works best for you. Contact us if you need help. Check the status of your order at any time, and follow up with any questions. Review your completed work, confirm it, and download the file! Want to make your essay worthy of a high grade? LET OUR EDITORS REVIEW YOUR WORK AT A 25% DISCOUNT! Had some minor typos to fix but overall the content was great. I am a very busy person who values time. ProEssays knows how to do its job really fast!!! It was my first time using such service and I was really nervous. Support Jamie was very friendly and helped me to fill in my order form. Thank you, guys!!! I will definitely use your service again. My professor is a real jerk! He likes to change the assignment in the very last moment. I have never thought that a 5-page research can be done in less than 6 hours. Thanks to ProEssays I managed to kick my professor's butt and party all the night. :) I am a foreigner, and I study at an English-speaking university. It is difficult for me to write serious papers in English. Here I solved my problem by ordering an essay from professionals. I am used to keeping everything under control, so it was important to me that the work I ordered included all my guidelines. The ProEssays staff was attentive to all my requirements. I will come back here again. I used the portal once. But I decided that it is better to write by yourself because you need to know the thought process when describing your topic. Otherwise, you risk finding yourself uncomfortable when you are asked questions about your work. But I definitely liked the service. It's good. When you're a mom of two and decide to get extra education, you don't always have time to do all the homework. I think people will understand me. So, ProEssays has become a lifesaver for me. I am glad that I found this service! I had a pretty specific topic, but I contacted customer service, and we found a writer who was able to help me. It wasn't easy, and we had to make several attempts. It took some time, but I understood that there might be difficulties with my topic. We got the job done in time in the end! I had a very urgent request and was willing to pay for speed, but they didn't agree to do the job in 3 hours. Every college student has had difficult times working on academic assignments. Some learners lack the motivation and inspiration necessary for the successful completion of the project, while others have no competence, knowledge, or writing skills it takes to thrive. Fortunately, there is always an opportunity to get professional help offered by our company. We are a team of expert writers aimed at aiding learners with their complicated college assignments. There are two main directions we are developing and advancing. First of all, browsing our website, you will notice an extensive database of free essays that provide learners with the desired motivation and inspiration for work. The most different subjects, topics, and issues are covered, so there is no chance you will fail to find the paper you need. Additionally, we also care for learners who have no chance to cope with their academic writing assignments due to a lack of time, skills, or other reasons. Students can place their orders and receive a well-structured and coherent paper from scratch. Editing and proofreading are extra services available on the website. 100% Free Essays: A Database Worth Your Attention Browsing the company website, the first thing you will notice is a list of subjects you can be interested in. Additionally, there is a variety of recently added essays that may seem relevant and helpful. In fact, these are free essay samples every college student can read and comprehend in order to get interesting ideas, powerful writing tools, and beneficial tips. The database of free essays is continuously updated with new essays offered by experienced college students. At this point, it is indispensable to mention that the papers you find here are not edited or proofread, so that you may find some typos, inaccuracies, and mistakes in the text. Besides, our company is not responsible for the quality of the essay samples available on the platform and does not recommend using them as custom-created essays. Instead, the works can inspire learners and fill them with new ideas to use in their personal papers. Thus, if you want to find a free essay to use as a template, you should follow numerous tips: Remember that you cannot copy the sample papers or use their paragraphs in your work, as it will be considered as plagiarism. Find the most relevant paper and take your time to read and analyze it. Pay attention to the structure of the text, style, and formatting. Write down interesting examples and statements you can change and expand in your essay. Read a few other essay examples to get more ideas and writing tools. Once students get their assignments, they may get overwhelmed and disappointed, especially if they have nothing to say on the topic. However, the struggle is still real, especially if you check out an extensive database of essay samples for college. Browsing the website, you will come across an unlimited number of appealing and relevant topics, generalized in broader categories. Each section is divided into subcategories that contain numerous samples each. As it has already been mentioned before, our collection is increasing every single day, so you should not give up looking for relevant topics and prominent ideas. Essential Literature-Related Topics How is it possible to create a coherent, well-structured, and deep analysis of the book if you have never read it? Well, it may be complicated to cope with the assignment, but it is still real. Browsing the Literature section, you may find a free essay to copy and use as a template for your project. Check out the variety of topics available within the category and make sure you will definitely find a paper worth your attention. STEM Essays and Their Themes Science, Technology, Engineering, and Mathematics are surely the most complicated subjects to analyze and write about. However, with the help of college writing essay examples, the undertaking may be a bit easier. Take your time to look through the topics available within the STEM category and find the most advantageous and helpful papers. Relevant statistics, up-to-date data, inspiring facts, and detailed guidelines on the paper creation are the most prominent points you will detect. Additionally, it is a great way to use the schemes, pictures, and tables to contribute to the quality of your work. Custom-Written Papers You Have Been Looking for Could not you find a sample of a college essay that seems helpful for your next project? Or do you simply lack time to browse the service and analyze the existing papers? There is no reason to give up, as due to the individual approach and maximum flexibility of our services, we always have a solution to offer. No matter if you have no desire to cope with the assignment or you failed with your attempts to detect a necessary paper sample, our professional writers will help you out. Contact the representatives of the customer support service to get detailed information about our custom essay writing service. Irrespective of the topic, academic level, number of pages, and deadline of your project, the team of qualified writers will provide you with the best solution. Additionally, you can ask the scholars to rewrite one of the sample papers so it is unique and free of plagiarism. The anonymity of the experience, confidentiality, safety, dependability, timeliness, and affordability of the services are guaranteed. Browse the ProEssays website to find a collection of the most coherent and appealing sample papers. Choose the subject you are interested in and get a list of essays available within the category. It took us a while to gather the most prominent and up-to-date papers in our database. However, it has been a rewarding project that helps hundreds of students succeed with their assignments every single day. Besides, it is indispensable to mention that the database is regularly updated, so you can often find new papers. Is it complicated for you to choose an interesting and appealing topic for your paper? Do you lack information about the proper paper style, formatting, or structure? These are the exact instances where our sample papers may be helpful. Besides, you should not forget about the inspiration, new writing tools, and effective instruments you will discover reading some of the works. Browsing the website, customers can find a range of other helpful sections that will draw them to the desired outcomes. Thus, there is a comprehensive blog with guidelines for flawless essay writing and an extensive list of relevant topics. Moreover, if you have no time to read all the instructions and paper samples necessary for the completion of the assignment, you can get professional assistance with your project. Place an order, pay for it and get it delivered within the specified while. Creating a database of free college essays, we wanted to help students stay inspired and motivated. Therefore, you can take maximum advantage of the information you find on the website. However, make sure you do not copy the parts of the sample papers, which will be marked as plagiarism. Tired of digging through piles of information? ORDER A UNIQUE ESSAY HERE AND ENJOY YOUR FREE TIME! Scheduling in operating system is the process of selecting a process from a ready queue. And allotting CPU to this process for execution. The operating system schedules the processes in such a way that the CPU doesn't sit idle. And keeps processing some or the other process. Scheduling is an important part of an operating system. As perfect scheduling increases the efficiency of the operating system. In this content, we will be discussing scheduling in more detail. Content: Scheduling in Operating System What is Scheduling in Operating System? Scheduling is the process of allotting the CPU to the processes present in the ready queue. We also refer to this procedure as process scheduling. The operating system schedules the process so that the CPU always has one process to execute. This reduces the CPU's idle time and increases its utilization. The part of OS that allots the computer resources to the processes is termed as a scheduler. It uses scheduling algorithms to decide which process it must allot to the CPU. Why Scheduling in Operating System is Important? In your day to day life, you schedule your work sequences to get: better efficiency. reduce any kind of delay. even waiting time. Like in the morning you wake up, have breakfast, work for some hours take lunch, again resume your work, at night have dinner and go to bed. Doing this way, you make your day more logical and efficient. Similarly, the operating system schedules the processes to achieve maximum CPU utilization. The processor can keep several processes in the main memory in the multiprogramming environment. And some or the other processes must be running at all times. As we know a typical process has both I/O times as well as CPU time. When a process completes its CPU time and now needs to perform I/O operation. The operating system takes the CPU from this process. And provides it to the other process waiting for the CPU. In this way, the operating system manages the CPU resources in an efficient way. The operating system schedules every computer resource before providing it to the process. Scheduling Queue The operating system maintains some queues. These queues sort out the scenario that which the process is waiting for what kind of operation. The queues maintained by the system are as follows: Job Queue As soon as any process enters the system, the system assembles it in the job queue. The system stores the job queue on the mass storage device i.e. secondary memory. Ready Queue If the process is ready for execution, Os brings it to the ready queue i.e. in the main memory. The process in the ready queue is the one those are waiting to get the CPU cycle. Device Queue The operating system maintains a separate device queue for each I/O device. This device queue holds the processes that are waiting to perform I/O on the device. Well, both the device queue and ready queue maintain the PCB for the processes they are carrying. I have briefly explained the process control block (PCB) in my earlier content. PCB of a process stores the relative information of the process such as: Scheduling Queue Sequences The process that is ready for execution is brought into the ready queue from the job queue. In the ready queue, it waits until the scheduler allots the CPU to the process. Once the process gets CPU and starts execution following sequence may take place: While execution, the process may create a new process. Then the system suspends the parent process until the child process terminates. Once the child process terminates the parent process is put back into the ready queue. If the time allotted to process for execution gets expired. And if the process execution isn't completed the scheduler put it back in the ready queue. If an interrupt occurs during the process execution, the processor suspends its execution. It handles the interrupt first and then puts the process back in the ready queue. During execution, if the process wants to perform an I/O operation, the system stores it in the device queue. After it completes its I/O operations, the system puts it back into the ready queue. If the process completes its execution, the processor terminates it. And it deallocates the PCB related to the process. It also releases the resources allotted to the process. Types of Scheduler Above we have discussed three scheduling queues. A process migrates among these queues before it terminates. The scheduler picks an appropriate process from these queues to maximize CPU utilization. This schedular can be further distinguished as: Long Term Scheduler Short Term Scheduler Medium Term Scheduler Long Term Scheduler/ Process Scheduling/ Job Scheduling The long-term schedular picks the process from the mass storage device such as a hard disk. And places it in the ready queue that is in the main memory. Because this scheduler selects a new process from the mass storage less frequently. The creation of new processes one after the other is at least separated by the minutes of duration. The long-term scheduler decides the degree of multiprogramming. If the degree of multiprogramming is stable. Then the average rate of process creation is equal to the average rate of process departure. In this condition, the long-term scheduler has to select a process from the mass storage. Only when a process leaves or departs the system. Short Term Scheduler/ CPU Scheduling The short-term schedular picks one process from the ready queue. And allots the CPU to this process. Why do we term it Short-Term Scheduler? Because this scheduler frequently selects the process from the ready queue. And allots it to the CPU. As the processes include both I/O time and CPU time. They keep on switching between the CPU and I/O devices. Thus, the CPU scheduler schedules the processes more frequently. The CPU scheduling is further classified into two types: Preemptive Scheduling In this scheduling, the schedular allots CPU to a process for a limited amount of time. The processor preempts the execution of a process if the process with higher priority occurs. Non-Preemptive Scheduling In this scheduling, once the schedular allots the CPU to any process, the process doesn't release the CPU. Until it gets terminated or it switches to the waiting state. Medium Term Scheduler The medium-term scheduler removes the partially executed process from the ready queue. This reduces the degree of multiprogramming. This process can be later reintroduced in the ready queue. And they can resume the execution from where it left off. This procedure is also referred to as swapping. Scheduling Criteria Scheduling criteria help to compare scheduling algorithms. CPU Utilization CPU utilization must be high, this is possible if the CPU is busy all-time executing one or the other process. Throughout Throughput is the number of processes executed by the CPU per unit of time. Turnaround Time Turnaround time is the time required by the process to execute completely. We count this time from the moment the process comes in the ready queue. Till the process completes its execution. Waiting Time Waiting time is time spent by the process in the ready queue. Response Time Response time is the time counted from the moment the process comes to the ready queue. Till the process produces the first result. Scheduling Algorithm The scheduling algorithm helps the scheduler in deciding which process it must pick from the ready queue. There are several scheduling algorithms as discussed below: First-Come, First-Served Scheduling The processor services the processes in the sequence they arrive in the ready queue. Shortest-Job-First Scheduling The processor services the shortest process in the ready queue first. Priority Scheduling The processor services the process with high priority in the ready queue first. Round-Robin Scheduling This is similar to the first-in-first-out scheduling. As the process arriving first in the ready queue will be serviced first. But the processor preempts the process after a time quantum. And placed back in the ready queue, the next process in the ready queue is given a chance to execute. There are a few more algorithms for scheduling the process. We will discuss them in some of our later content. So, scheduling is the important part of the operating system that controls the degree of multiprogramming. Thus perfect scheduling increases the CPU utilization making the system more efficient. A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms. There are six popular process scheduling algorithms which we are going to discuss in this chapter − First-Come, First-Served (FCFS) Scheduling Shortest-Job-Next (SJN) Scheduling Priority Scheduling Shortest Remaining Time Round Robin(RR) Scheduling Multiple-Level Queues Scheduling These algorithms are either non-preemptive or preemptive. Non-preemptive algorithms are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time, whereas the preemptive scheduling is based on priority where a scheduler may preempt a low priority running process anytime when a high priority process enters into a ready state. First Come First Serve (FCFS) Jobs are executed on first come, first serve basis. It is a non-preemptive, pre-emptive scheduling algorithm. Easy to understand and implement. Its implementation is based on FIFO queue. Poor in performance as average wait time is high. Wait time of each process is as follows − Process Wait Time : Service Time - Arrival Time P0 0 - 0 = 0 P1 5 - 1 = 4 P2 8 - 2 = 6 P3 16 - 3 = 13 Average Wait Time: (0+4+6+13) / 4 = 5.75 This is also known as shortest job first, or SJF This is a non-preemptive, pre-emptive scheduling algorithm. Best approach to minimize waiting time. Easy to implement in Batch systems where required CPU time is known in advance. Impossible to implement in interactive systems where required CPU time is not known. The processer should know in advance how much time process will take. Given: Table of processes, and their Arrival time, Execution time Process Arrival Time Execution Time Service Time P0 0 5 0 P1 1 3 5 P2 2 8 14 P3 3 6 8 Waiting time of each process is as follows − Process Waiting Time P0 0 - 0 = 0 P1 5 - 1 = 4 P2 14 - 2 = 12 P3 8 - 3 = 5 Average Wait Time: (0 + 4 + 12 + 5)/4 = 21 / 4 = 5.25 Priority Based Scheduling Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems. Each process is assigned a priority. Process with highest priority is to be executed first and so on. Processes with same priority are executed on first come first served basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement. Given: Table of processes, and their Arrival time, Execution time, and priority. Here we are considering 1 is the lowest priority. Process Arrival Time Execution Time Priority Service Time P0 0 5 1 0 P1 1 3 2 11 P2 2 8 1 14 P3 3 6 3 5 Waiting time of each process is as follows − Process Waiting Time P0 0 - 0 = 0 P1 11 - 1 = 10 P2 14 - 2 = 12 P3 5 - 3 = 2 Average Wait Time: (0 + 10 + 12 + 2)/4 = 24 / 4 = 6 Shortest Remaining Time Shortest remaining time (SRT) is the preemptive version of the SJN algorithm. The processor is allocated to the job closest to completion but it can be preempted by a newer ready job with shorter time to completion. Impossible to implement in interactive systems where required CPU time is not known. It is often used in batch environments where short jobs need to give preference. Round Robin Scheduling Round Robin is the preemptive process scheduling algorithm. Each process is provided a fix time to execute, it is called a quantum. Once a process is executed for a given time period, it is preempted and other process executes for a given time period. Context switching is used to save states of preempted processes. Wait time of each process is as follows − Process Wait Time : Service Time - Arrival Time P0 (0 - 0) + (12 - 3) = 9 P1 (3 - 1) = 2 P2 (6 - 2) + (14 - 9) + (20 - 17) = 12 P3 (9 - 3) + (17 - 12) = 11 Average Wait Time: (9+2+12+11) / 4 = 8.5 Multiple-Level Queues Scheduling Multiple-level queues are not an independent scheduling algorithm. They make use of other existing algorithms to group and schedule jobs with common characteristics. Multiple queues are maintained for processes with common characteristics. Each queue can have its own scheduling algorithms. Priorities are assigned to each queue. For example, CPU-bound jobs can be scheduled in one queue and all I/O-bound jobs in another queue. The Process Scheduler then alternately selects jobs from each queue and assigns them to the CPU based on the algorithm assigned to the queue. CPU scheduling is a process used by the operating system to decide which task or process gets to use the CPU at a particular time. This is important because a CPU can only handle one task at a time, but there are usually many tasks that need to be processed. The following are different purposes of a CPU scheduling time.Maximize the CPU utilizationMinimize the response and waiting time of the process.What is the Need for a CPU Scheduling Algorithm?CPU scheduling is the process of deciding which process will own the CPU to use while another process is suspended. The main function of CPU scheduling is to ensure that whenever the CPU remains idle, the OS has at least selected one of the processes available in the ready-to-use line.In Multiprogramming, if the long-term scheduler selects multiple I/O binding processes then most of the time, the CPU remains idle. The function of an effective program is to improve resource utilization.Terminologies Used in CPU SchedulingArrival Time: The time at which the process arrives in the ready queue.Completion Time: The time at which the process completes its execution.Burst Time: Time required by a process for CPU execution.Turn Around Time: Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival TimeWaiting Time(W.T): Time Difference between turn around time and burst time.Waiting Time = Turn Around Time – Burst TimeThings to Take Care While Designing a CPU Scheduling AlgorithmDifferent CPU Scheduling algorithms have different structures and the choice of a particular algorithm depends on a variety of factors. CPU Utilization: The main purpose of any CPU algorithm is to keep the CPU as busy as possible. Theoretically, CPU usage can range from 0 to 100 but in a real-time system, it varies from 40 to 90 percent depending on the system load.Throughput: The average CPU performance is the number of processes performed and completed during each unit. This is called throughput. The output may vary depending on the length or duration of the processes.Turn Round Time: For a particular process, the important conditions are how long it takes to perform that process. The time elapsed from the time of process delivery to the time of completion is known as the conversion time. Conversion time is the amount of time spent waiting for memory access, waiting in line, using CPU and waiting for I/O.Waiting Time: The Scheduling algorithm does not affect the time required to complete the process once it has started performing. It only affects the waiting time of the process i.e. the time spent in the waiting process in the ready queue.Response Time: In a collaborative system, turn around time is not the best option. The process may produce something early and continue to computing the new results while the previous results are released to the user. Therefore another method is the time taken in the submission of the application process until the first response is issued. This measure is called response time.Different Types of CPU Scheduling AlgorithmsThere are mainly two types of scheduling methods:Preemptive Scheduling: Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to the ready state.Non-Preemptive Scheduling: Non-Preemptive scheduling is used when a process terminates , or when a process switches from running state to waiting state.CPU SchedulingPlease refer Preemptive vs Non-Preemptive Scheduling for details.CPU Scheduling AlgorithmsLet us now learn about these CPU scheduling algorithms in operating systems one by one:Comparison of CPU Scheduling AlgorithmsHere is a brief comparison between different CPU scheduling algorithms:Algorithm Allocation Complexity Average waiting time (AWT) Preemption Starvation PerformanceFCFSAccording to the arrival time of the processes, the CPU is allocated. Simple and easy to implement Large. No No Slow performanceSJF Based on the lowest CPU burst time (BT). More complex than FCFSSmaller than FCFS No Yes Minimum Average Waiting TimeSRTFSame as SJF the allocation of the CPU is based on the lowest CPU burst time (BT). But it is preemptive.More complex than FCFS Depending on some measures e.g., arrival time, process size, etcYesYes The preference is given to the short jobsRRAccording to the order of the process arrives with fixed time quantum (TQ) The complexity depends on Time Quantum sizeLarge as compared to SJF and Priority scheduling. Yes No Each process has given a fairly fixed timePriority Pre-emptive According to the priority. The bigger priority task executes first This type is less complex Smaller than FCFSYes YesWell performance but contain a starvation problemPriority non-preemptive According to the priority with monitoring the new incoming higher priority jobsThis type is less complex than Priority preemptivePreemptive Smaller than FCFS No Yes Most beneficial with batch systemsMLQ According to the process that resides in the bigger queue priority More complex than the priority scheduling algorithmsSmaller than FCFS No Yes Good performance but contain a starvation problemMFLQ According to the process of a bigger priority queue. It is the most Complex but its complexity rate depends on the TQ size Smaller than all scheduling types in many cases No No Good performanceQuestions for PracticeQuestion: Which of the following is false about SJF? S1: It causes minimum average waiting time S2: It can cause starvation (A) Only S1(B) Only S2 (C) Both S1 and S2 (D) Neither S1 nor S2 Answer: (D) S1 is true SJF will always give minimum average waiting time. S2 is true SJF can cause starvation. Question: Consider the following table of arrival time and burst time for three processes P0, P1 and P2.ProcessArrival timeBurst TimeP00 ms 9 msP11 ms4 msP22 ms9 msThe pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of processes. What is the average waiting time for the three processes? (A) 5.0 ms (B) 4.33 ms (C) 6.33 (D) 7.33 Solution: (A) Process P0 is allocated processor at 0 ms as there is no other process in the ready queue. P0 is preempted after 1 ms as P1 arrives at 1 ms and burst time for P1 is less than remaining time of P0. P1 runs for 4ms. P2 arrived at 2 ms but P1 continued as burst time of P2 is longer than P1. After P1 completes, P0 is scheduled again as the remaining time for P0 is less than the burst time of P2. P0 waits for 4 ms, P1 waits for 0 ms and P2 waits for 11 ms. So average waiting time is (0+4+11)/3 = 5. Question: Consider the following set of processes, with the arrival times and the CPU-burst times given in milliseconds.ProcessArrival timeBurst TimeP10 ms 5 msP21 ms3 msP32 ms3 msP44 ms1 msWhat is the average turnaround time for these processes with the preemptive Shortest Remaining Processing Time First algorithm ? (A) 5.50 (B) 5.75 (C) 6.00 (D) 6.25 Answer: (A) Solution: The following is Gantt Chart of executionTurn Around Time = Completion Time – Arrival Time Avg Turn Around Time = (12 + 3 + 6+ 1)/4 = 5.50 Question: An operating system uses the Shortest Remaining Time First (SRTF) process scheduling algorithm. Consider the arrival times and execution times for the following processes:ProcessBurst TimeArrival TimeP120 ms 0 msP225 ms15 msP310 ms30 msP415 ms45 msWhat is the total waiting time for process P2? (A) 5 (B) 15 (C) 40 (D) 55

Answer (B) Solution: At time 0, P1 is the only process, P1 runs for 15 time units. At time 15, P2 arrives, but P1 has the shortest remaining time. So P1 continues for 5 more time units. At time 20, P2 is the only process. So it runs for 10 time units At time 30, P3 is the shortest remaining time process. So it runs for 10 time units At time 40, P2 runs as it is the only process. P2 runs for 5 time units. At time 45, P3 arrives, but P2 has the shortest remaining time. So P2 continues for 10 more time units. P2 completes its execution at time 55.Total waiting time for P2 = Completion time - (Arrival time + Execution time)= 55 - (15 + 25)= 15