****

**Luxor university**

**Faculty of computers and information**

**OS COURSE PROJECT**

**Second Semester (2022-2023)**

**The Third level**

**CPU Scheduling**

* **Names of team members and The task of each team member:**

|  |  |
| --- | --- |
| Sarah Nabil Kamel (CS). | FCFS,SJFP,SJFNP,RR |
| Esraa Mohammed Abu Al-Wafa (CS). | **RR** |
| Ebtihal Abdel Moneim (CS). | **FCFS** |
| Hadeer Hadeer (IT). | **SJFNP** |

* **Scheduling policies:**

**are essential tools for managing resources and optimizing task execution in various systems. Each policy has its own advantages and limitations, and the choice of an appropriate policy depends on the specific requirements and characteristics of the system under consideration. A combination of multiple policies or hybrid approaches can also be employed to achieve better results in complex scenarios and make the CPU in maximum Utilization.**

**First-Come, First-Serve (FCFS) scheduler**

**FCFS is a simple scheduling policy where tasks or processes are executed in the order of their arrival. It is easy to implement and ensures fairness by treating all tasks equally. However, it suffers from the "convoy effect," where a long-running task can delay subsequent tasks, leading to inefficient resource utilization and increased response times.**

**A screenshot of a computer

Description automatically generatedA screen shot of a computer program

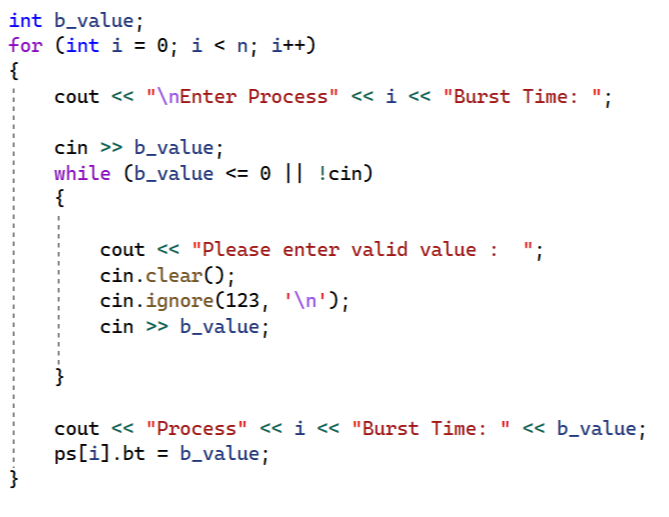
Description automatically generated with low confidence**

**A screenshot of a computer program

Description automatically generated with low confidence**

**A picture containing text, font, screenshot, line

Description automatically generated**

**A screenshot of a computer code

Description automatically generated with medium confidence** **A screenshot of a computer program

Description automatically generated with low confidence**

**A screen shot of a computer code

Description automatically generated with low confidenceA picture containing text, screenshot, font, number

Description automatically generated**

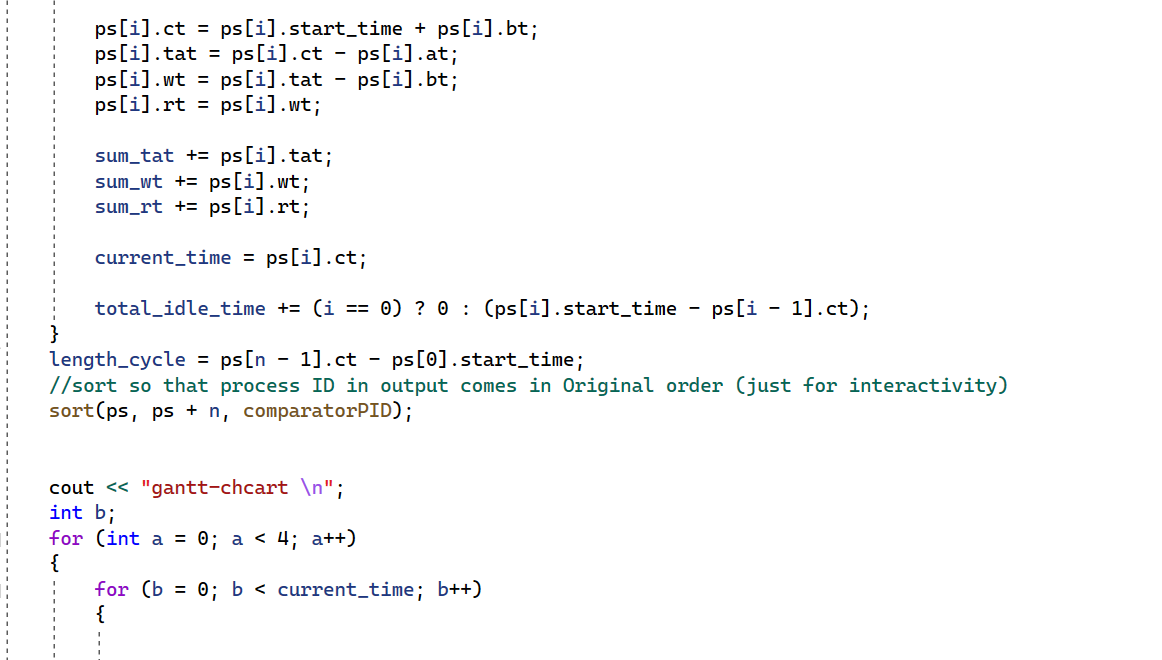
**A screenshot of a computer program

Description automatically generated with low confidence**

****

**A screen shot of a computer code

Description automatically generated with low confidence**

****

**A line of dots on a white background

Description automatically generated with low confidenceA screenshot of a computer code

Description automatically generated with low confidence**

**A line of dots on a white background

Description automatically generated with low confidence**

**A picture containing text, screenshot, font, line

Description automatically generated**

**A white background with red text

Description automatically generated with low confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

**Shortest Job First non-preemptive (SJFnp) scheduler**

**A screenshot of a computer program

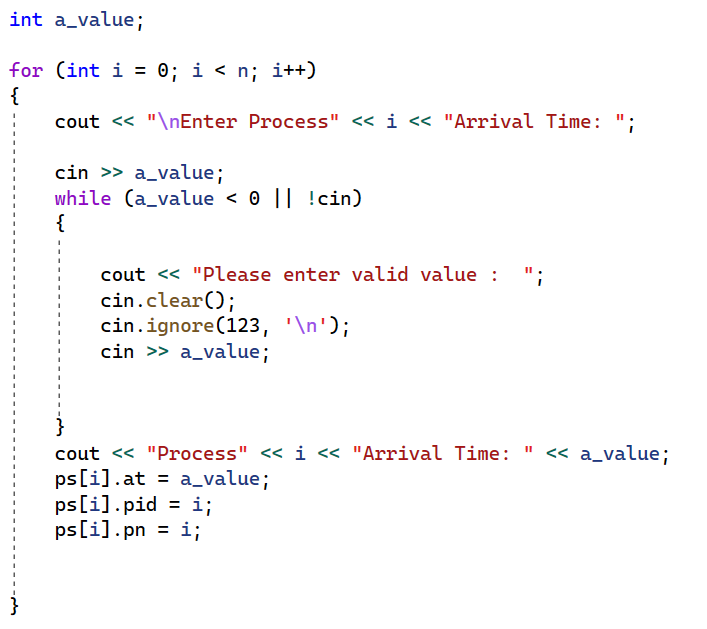
Description automatically generated with medium confidenceaims to minimize the response time and overall execution time of processes by prioritizing the shortest remaining bursts. However, it may cause more frequent context switches and overhead due to preemption, which can impact system performance.**

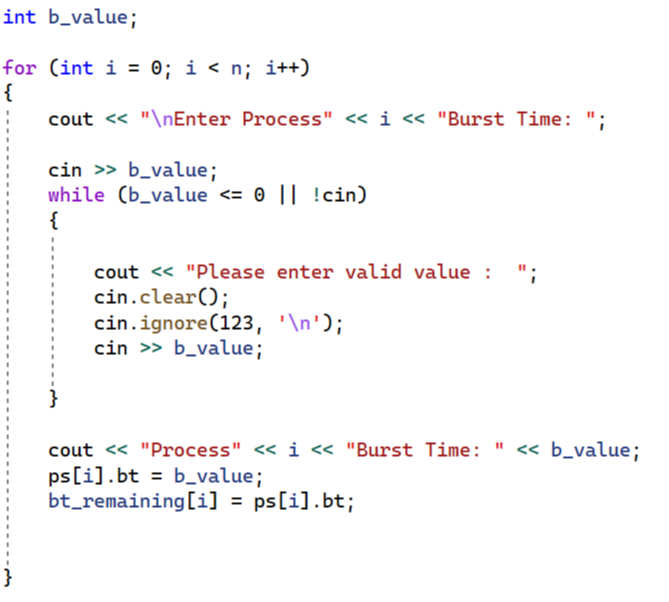
**A picture containing text, screenshot, software, number

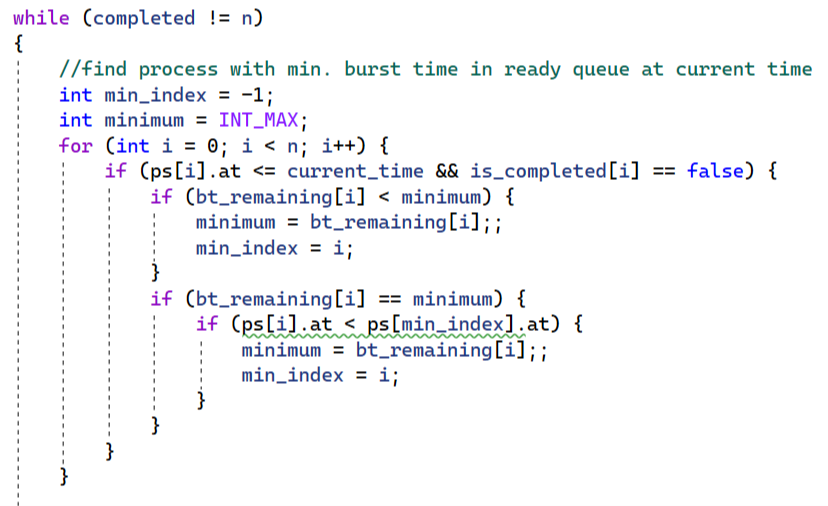
Description automatically generated**

**A picture containing text, font, number, line

Description automatically generated**

****

****

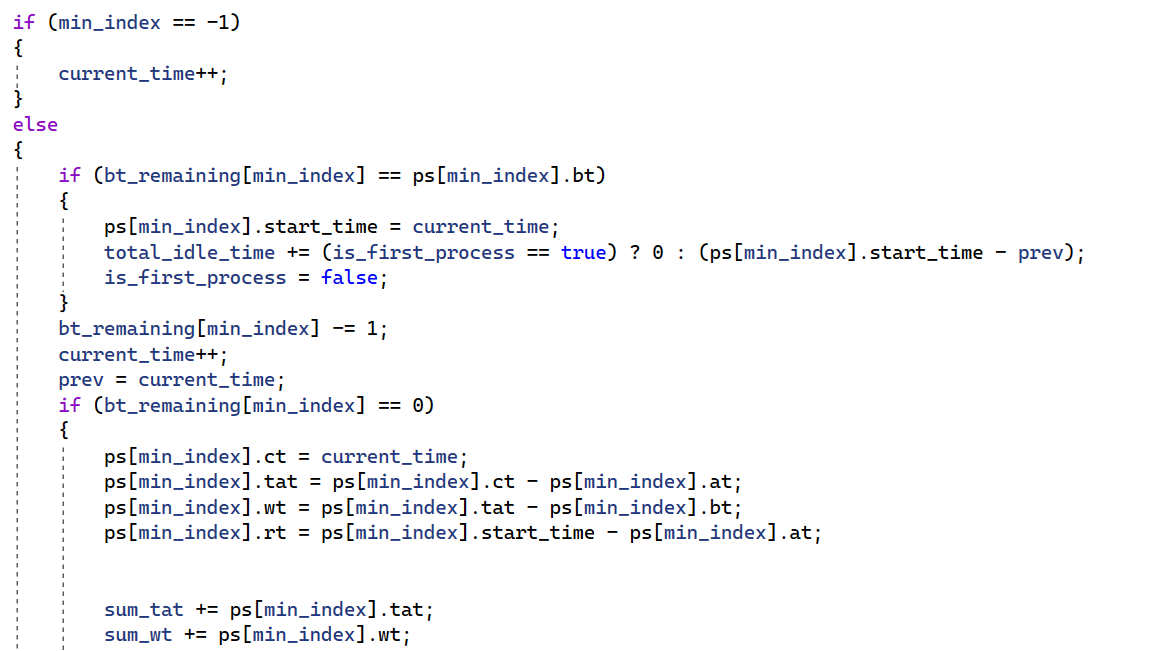
****

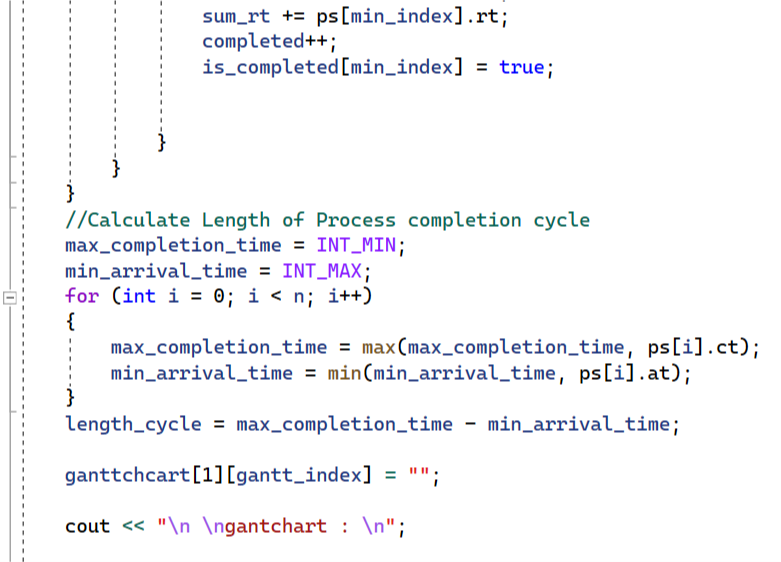
**A screenshot of a computer program

Description automatically generated with low confidence**

**A screenshot of a computer program

Description automatically generated with low confidence**

****

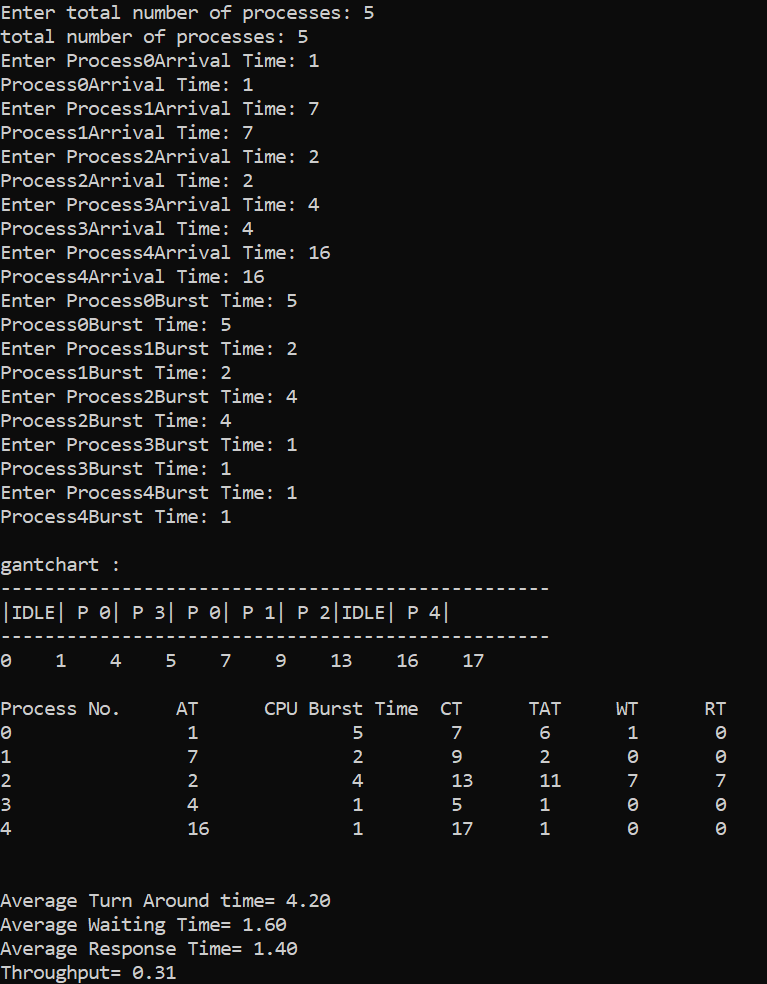
****

**A screenshot of a computer program

Description automatically generated with low confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

**Shortest Job First non-preemptive (SJFnp) scheduler**

**is a scheduling algorithm used in operating systems to determine the order in which processes are executed on a CPU . In SJF scheduling, the process with the shortest burst time is given the highest priority and is scheduled to run first.SJF non-preemptive scheduling aims to minimize the average waiting time of processes because it prioritizes shorter processes, allowing them to finish quickly. However, it can suffer from a potential drawback known as "starvation." If a long process arrives early and there are several shorter processes in the queue, the longer process may have to wait for a significant amount of time until all the shorter processes are executed. This can result in a delay for longer processes and unfairness in resource allocation.**

**A screenshot of a computer program

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated**

**A screen shot of a computer program

Description automatically generated with low confidence**

**A picture containing text, screenshot, display, software

Description automatically generated**

**A picture containing text, screenshot, font, number

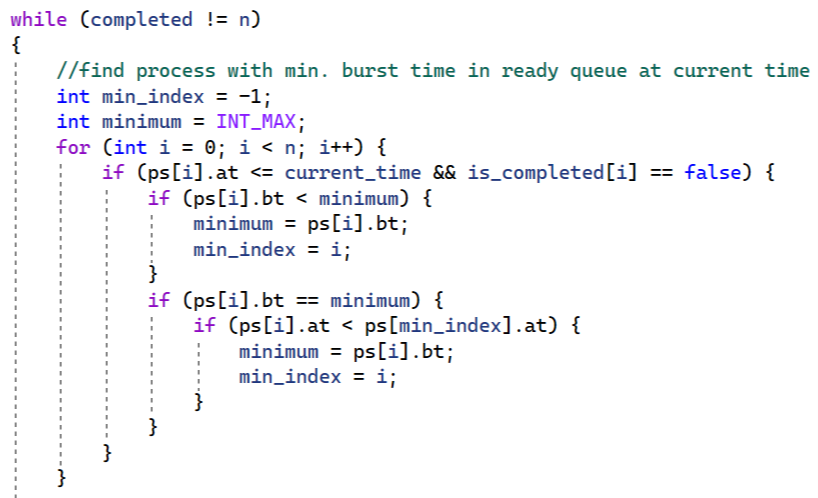
Description automatically generated**

**A screenshot of a computer code

Description automatically generated with medium confidence**

**A screenshot of a computer code

Description automatically generated with medium confidence**

****

**A screenshot of a computer program

Description automatically generated with low confidence**

**A screenshot of a computer program

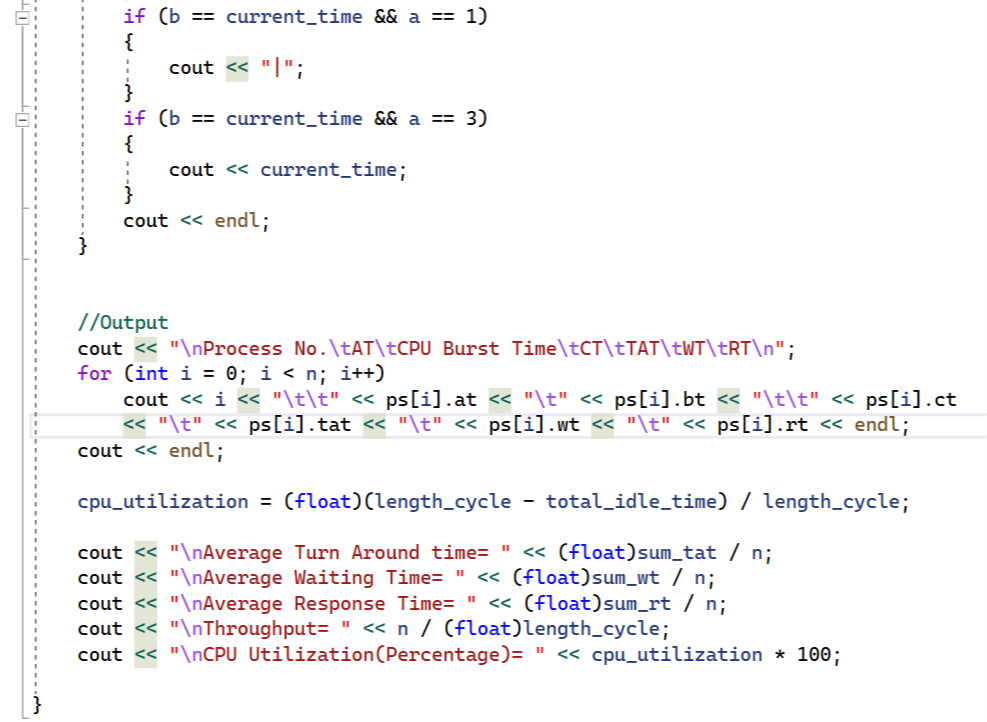
Description automatically generated with medium confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

****

**A screenshot of a computer program

Description automatically generated with medium confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

**Round Robin (RR):**

**RR is a preemptive scheduling policy where tasks are assigned a fixed time quantum or time slice. Tasks are executed in a cyclic manner, with each task running for a specified time and then preempted to allow other tasks to execute. RR ensures fairness and prevents starvation but may suffer from high overhead due to frequent context switching**

**Effects of different length quantum on a Round-Robin scheduler:**

**1-Impact on task responsiveness:**

**When the quantum length is short, tasks are scheduled more frequently, resulting in improved responsiveness. However, this also leads to increased overhead due to frequent context switches between tasks. On the other hand, when the quantum length is long, fewer context switches occur, which reduces overhead but can lead to longer response times for tasks. Therefore, the quantum length should be chosen carefully based on the specific requirements of the system.**

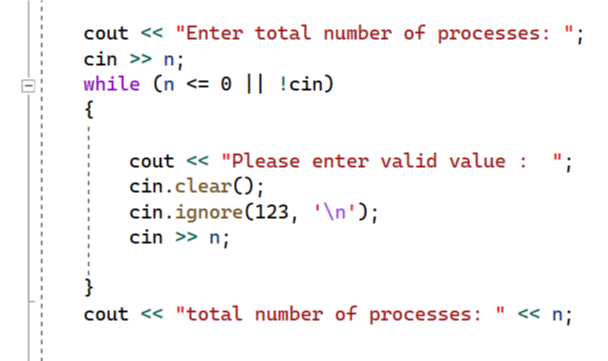
**2-Impact on throughput:**

**The quantum length can also affect the overall throughput of the system. A shorter quantum length results in more frequent context switches, which can increase the overhead and reduce the overall throughput. However, a longer quantum length can lead to lower utilization of the CPU, which can also reduce the overall throughput.**

**3-Impact on fairness:**

**A screenshot of a computer program

Description automatically generated with medium confidenceA shorter quantum length can improve fairness by ensuring that each task receives its fair share of the CPU time. This is because tasks are scheduled more frequently, which reduces the likelihood of a long-running task monopolizing the CPU. However, a longer quantum length can also be fair if it allows each task to complete a significant portion of its work before being preempted.**

**A screen shot of a computer code

Description automatically generated with low confidence**

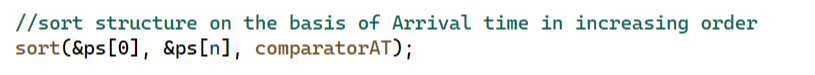
**A screenshot of a computer program

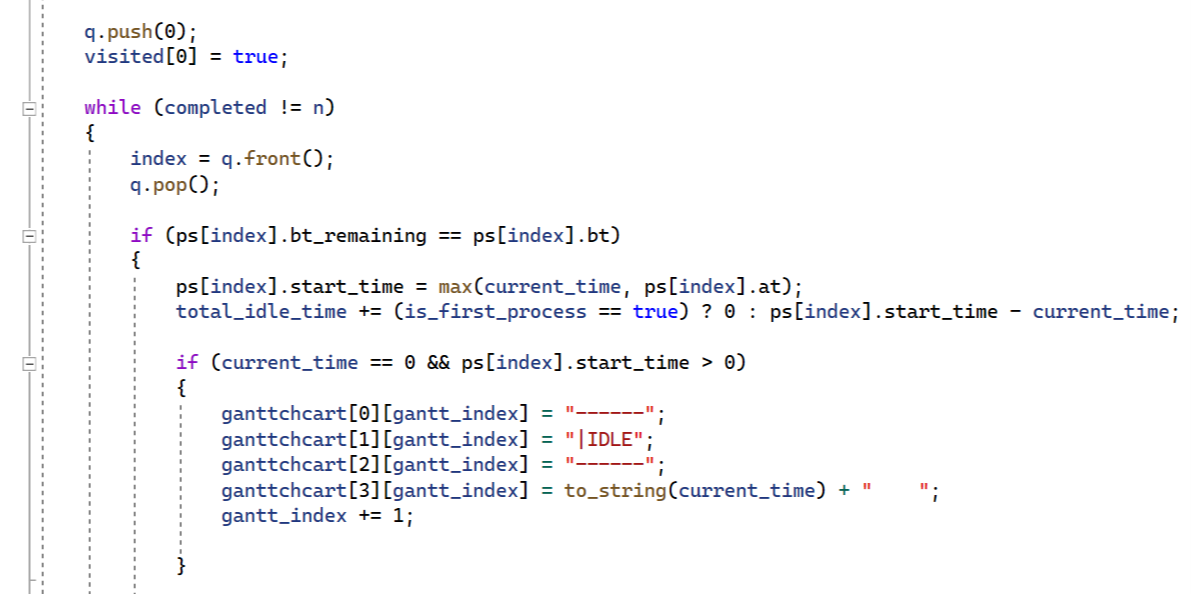
Description automatically generated with medium confidence**

**A screenshot of a computer code

Description automatically generated with medium confidenceA screenshot of a computer program

Description automatically generated with low confidence**

****

****

**A picture containing text, screenshot, font, software

Description automatically generated**

**A screenshot of a computer code

Description automatically generated with low confidence**

**A picture containing text, font, line, screenshot

Description automatically generated**

**A picture containing text, screenshot, font, number

Description automatically generated**

**A picture containing text, font, line, screenshot

Description automatically generated**

**A screenshot of a computer program

Description automatically generated with medium confidence**

**A picture containing text, font, line, screenshot

Description automatically generated**

**A screenshot of a computer program

Description automatically generated with medium confidenceA screenshot of a computer program

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated**

**Effects of length quantum A screenshot of a computer

Description automatically generated= 0 :**

**A black background with white text

Description automatically generated with low confidenceImpossible because there is no process Brust Time equal to zero In addition, if the value is zero, this means that the process has no time to execute**

**A picture containing text, screenshot, font, black

Description automatically generatedA screenshot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated with low confidence A screenshot of a computer program

Description automatically generated with medium confidence**

**The effects of preemptive versus non-preemptive scheduling:**

**1-Response time:**

**Preemptive scheduling allows a higher-priority task to interrupt and preempt a lower-priority task currently running. This reduces the response time for critical tasks since they can be executed immediately. In non-preemptive scheduling, a task continues to run until it completes or voluntarily releases the CPU. As a result, response times for high-priority tasks may be delayed if lower-priority tasks are occupying the CPU.**

**2-Fairness:**

**Preemptive scheduling ensures fairness by allowing higher-priority tasks to execute as soon as they become available. This prevents lower-priority tasks from monopolizing the CPU for extended periods. Non-preemptive scheduling may result in lower-priority tasks experiencing increased waiting times if higher-priority tasks continue to arrive or remain active.**

**3-Overhead:**

**Preemptive scheduling generally incurs higher overhead due to the need for context switching. When a higher-priority task preempts a lower-priority task, the system must save the current state of the preempted task and restore the state of the higher-priority task. This additional overhead can reduce overall system efficiency. Non-preemptive scheduling avoids this overhead since tasks run to completion without interruption.**

**4-Deadline and resource management:**

**Preemptive scheduling is often preferred in real-time systems where meeting deadlines is critical. It allows for the timely execution of high-priority tasks to ensure that deadlines are met. Additionally, preemptive scheduling facilitates efficient resource management**

**as tasks can be interrupted if higher-priority tasks require resources.**

**5-Complexity:**

**Preemptive scheduling algorithms are generally more complex than non-preemptive ones**

**due to the need for priority evaluation, context switching, and handling of task preemption.**

**Non-preemptive scheduling is simpler to implement since tasks run to completion without interruptions.**