Project1:

In this project, show different types of CPU scheduling (FCFS, Round robins, Shortest Job First, Priority Scheduling, Shortest Remaining Time First, Longest Remaining Time First, and Highest Response Ratio Next). Give a transient event and show how these different scheduling react to that.

The whole code part: <https://github.com/nanxuanhui/CS5531.git>

1. Number of process= last two digits of your student ID (i.e., if your student id is 10101025, then number of process will be “25”. For process less than 10 add ten to it, i.e., if your student id is 10101005 then number of process will be “05+10=15”. If you have any confusion, please email me with your student id.

Student ID: 16356630

Number of process = 30

1. Chose any of the five CPU scheduling from the following,

* FCFS (First Come First Serve)
* SJF (Shortest Job First)
* LJF (Longest Job First)
* Round Robin (RR)
* Priority Scheduling
* Shortest Remaining Time First
* Longest Remaining Time First
* Highest Response Ratio Next

So, I choose for FCFS (First Come First Serve), LJF (Longest Job First), Round Robin (RR), Longest Remaining Time First, Highest Response Ratio Next.

1. Do scheduling for the number of process on step 1.

文本

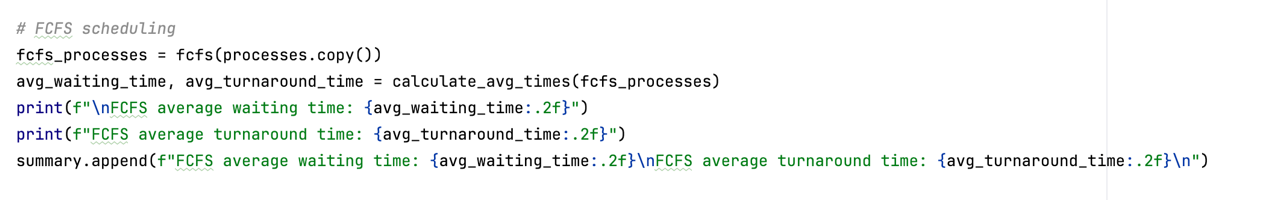
低可信度描述已自动生成

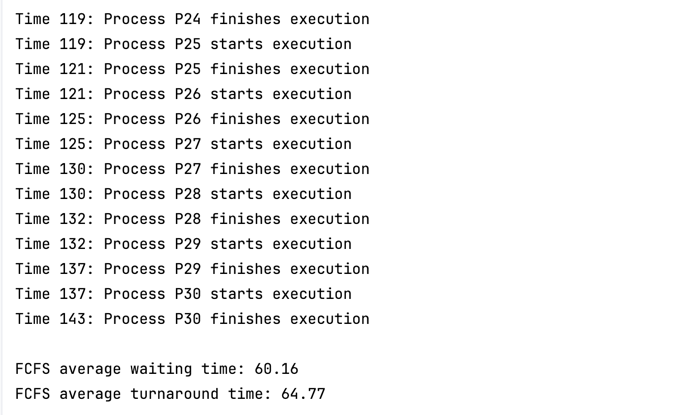
表格

描述已自动生成

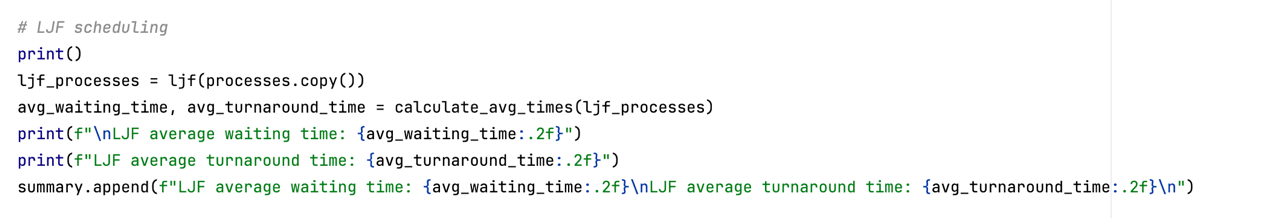
1. Give a transient event and show how these different scheduling react to that.

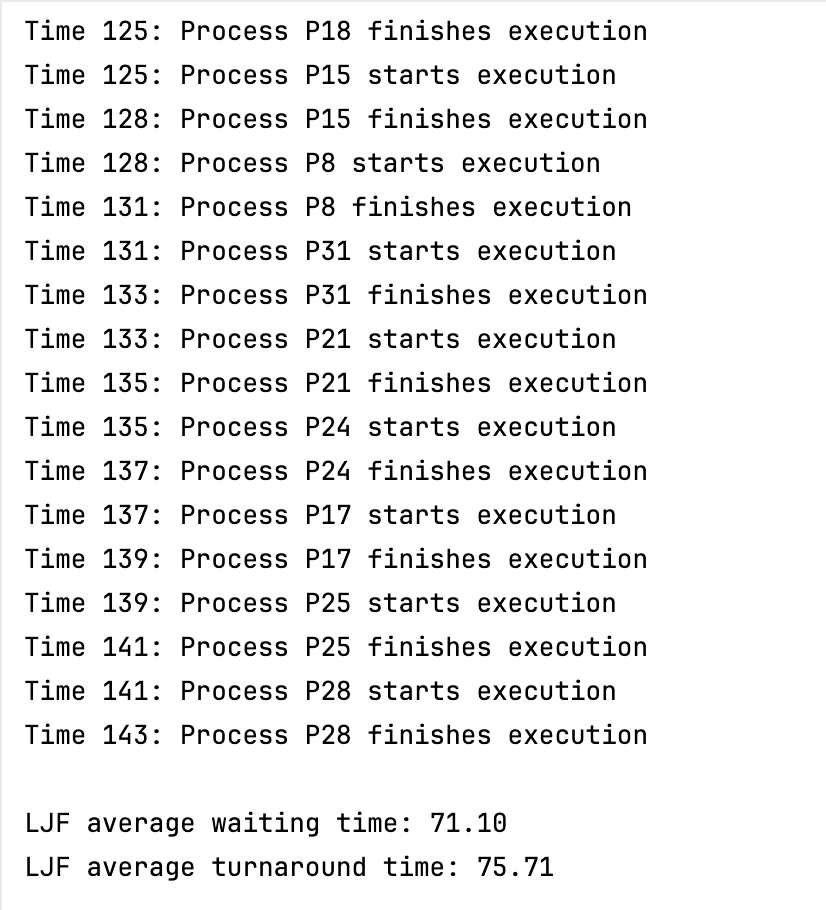
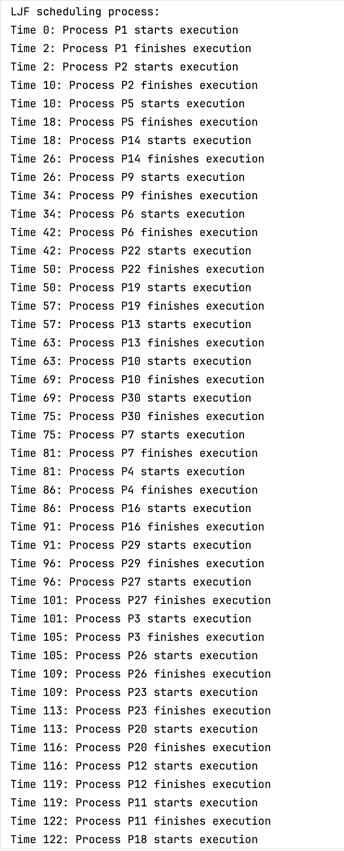
FCFS (First Come First Serve):



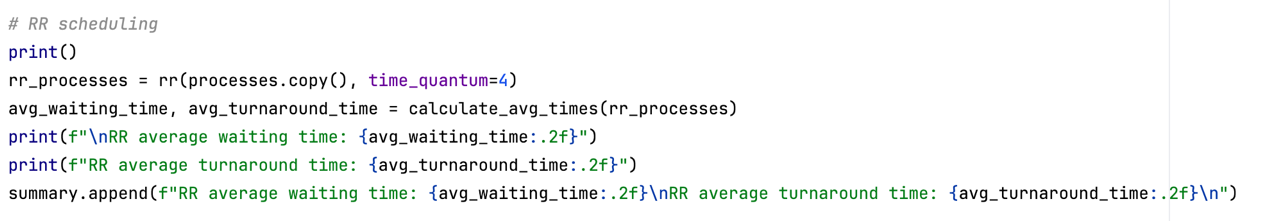
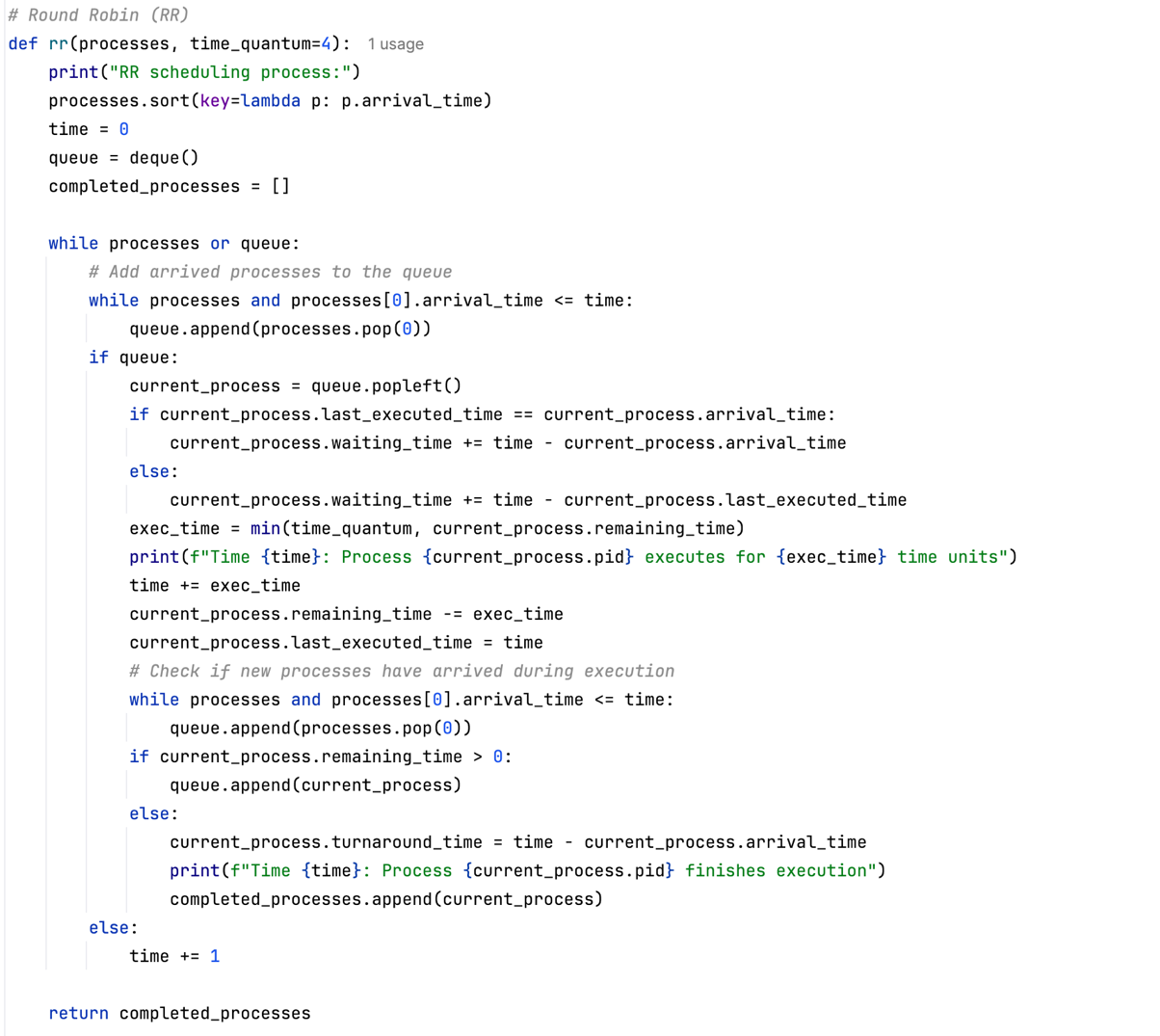


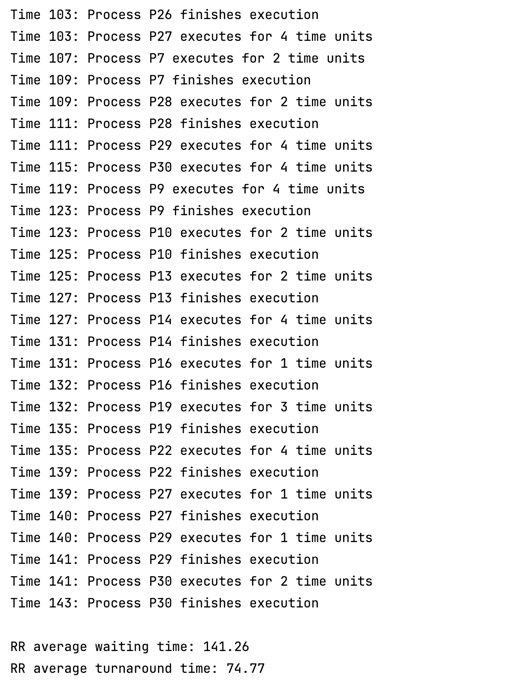
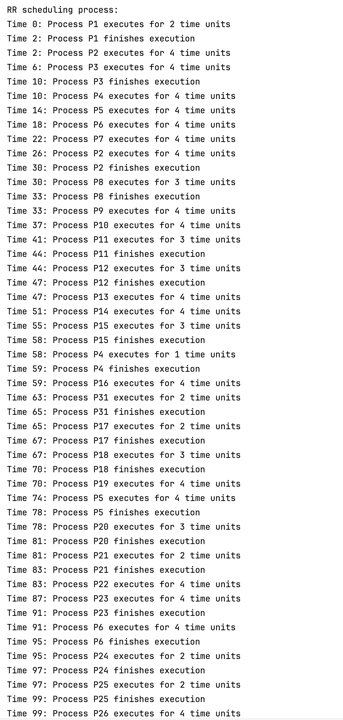
LJF (Longest Job First):





Round Robin (RR):



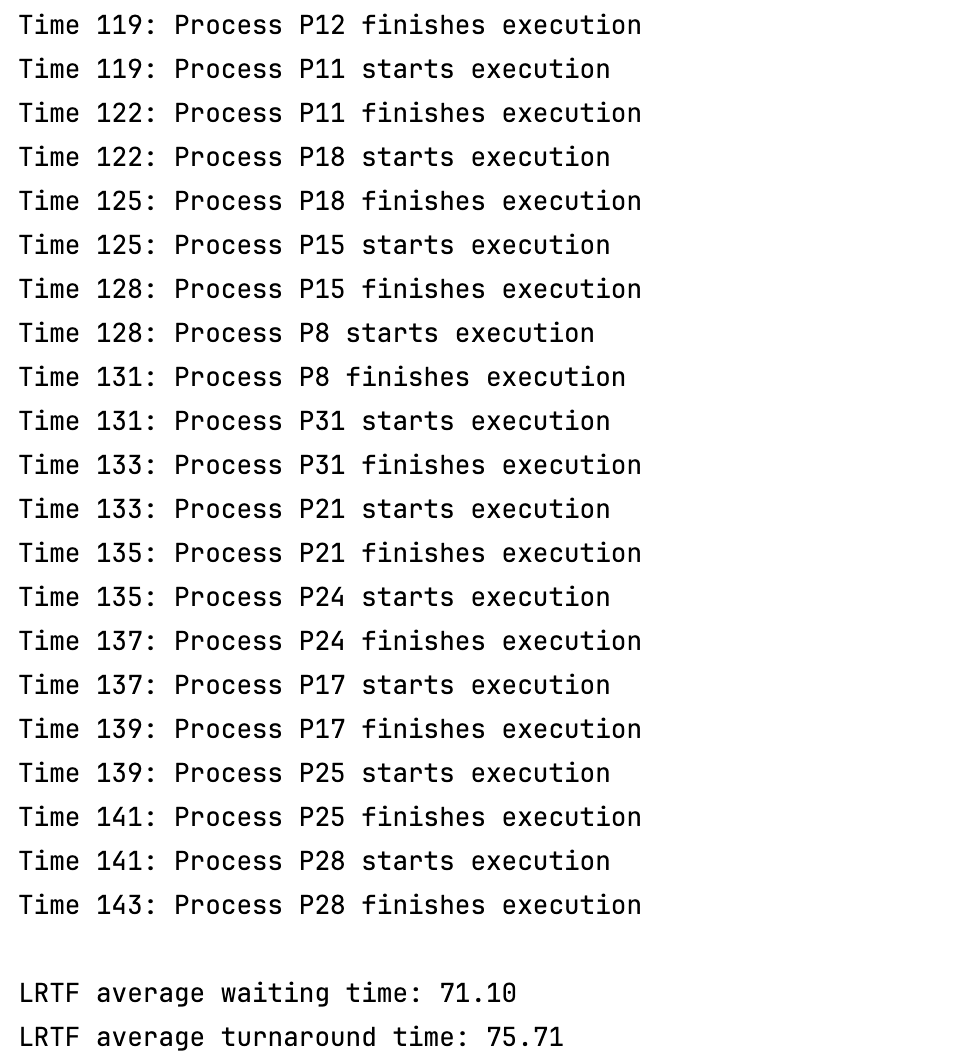
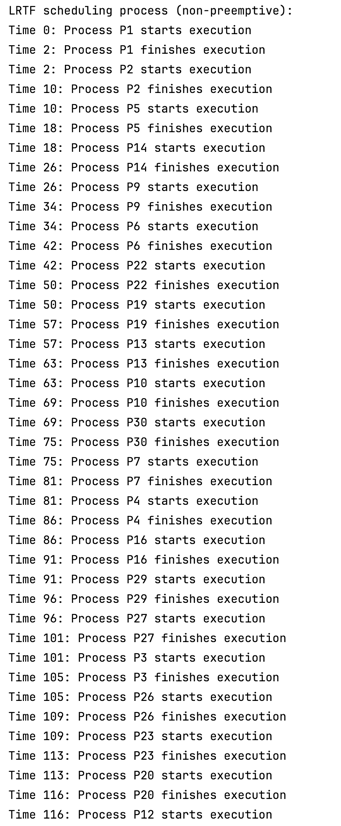


Longest Remaining Time First:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成文本

描述已自动生成

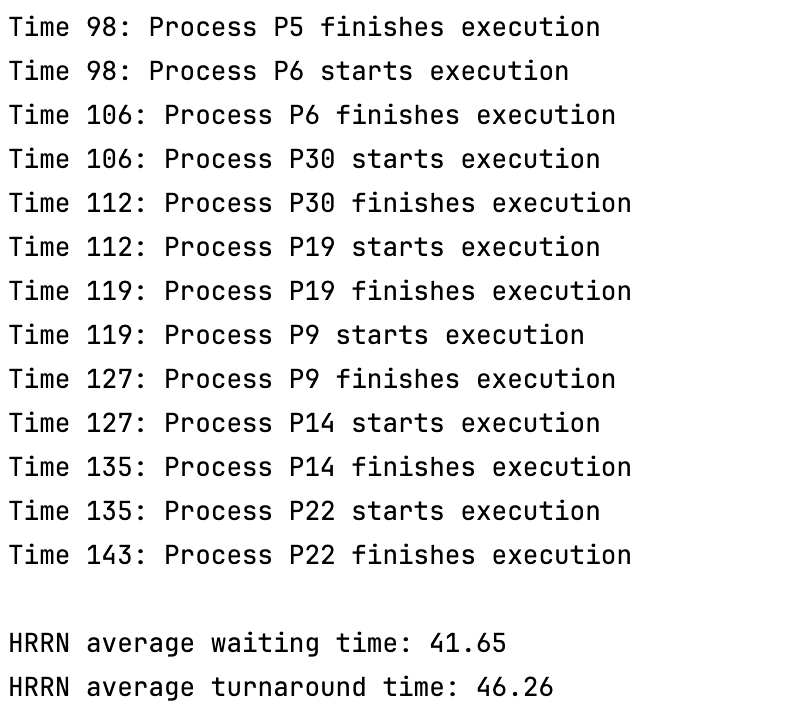
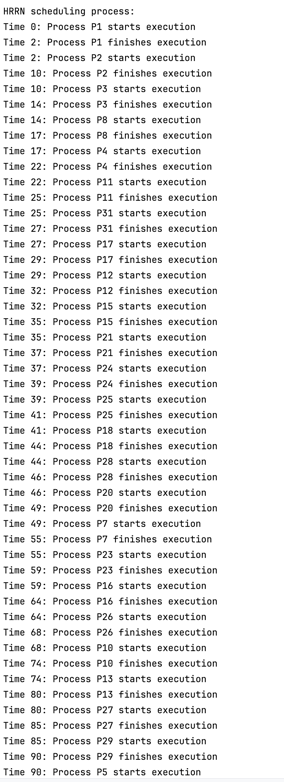


Highest Response Ratio Next:

图形用户界面, 文本, 应用程序

描述已自动生成文本

描述已自动生成



1. Compare all 5 of your chosen scheduling and show the comparison with a conclusion of pros and cons for each scheduling.

HRRN has the lowest wait time at 41.65, indicating that processes wait less on average before executing. RR has the highest waiting time, reaching 141.26, which is significantly higher than other algorithms. HRRN also performed best, with an average turnaround time of 46.26.

FCFS (First Come First Served) is simple to implement and has clear logic. It processes in the order of arrival to ensure that no process is prioritized or delayed. Short processes may be blocked by long processes, resulting in increased overall waiting time. It cannot dynamically respond to the arrival of high-priority processes and has poor adaptability.

LJF (Longest Job First) can reduce the overall completion time of long processes by giving priority to them. Short processes may need to wait for a long time, affecting system responsiveness. If long processes continue to arrive, short processes may never be executed.

RR (Round Robin) each process gets an execution opportunity when it is its turn, preventing starvation. Suitable for time-sharing systems, processes can respond quickly. Frequent context switching may increase system overhead, especially when the time slice is too short. RR has the highest average waiting time.

HRRN (Highest Response Ratio First) takes into account both waiting time and execution time, reducing starvation and the "convoy effect". HRRN has the lowest waiting time and turnaround time. It requires real-time calculation of the response ratio, which increases scheduling overhead. It requires maintenance and updating of the response ratio, and the logic is relatively complex.

LRTF (Longest Remaining Time First) prioritizes processes with the longest remaining time, potentially reducing their turnaround time. It can dynamically respond to the arrival of new processes and optimize CPU utilization. Short processes may not be able to execute due to the continuous arrival of long processes. Frequent preemption and context switching increase system overhead.