EE3220 System-on-Chip Design

Tutorial Note 4 – Operating System Scheduling Algorithms

Department of
Electrical Engineering
香港城市大學
City University of Hong Kong

1. Introduction

Why do we need scheduling?

A typical process involves both I/O time and CPU time. In a programming system like MS-DOS, time spent waiting for I/O is wasted and CPU is free during this time. In multi-programming systems, one process can use CPU while another is waiting for I/O. This is possible only with process scheduling.

Generally, scheduling algorithms can be divided into non-preemptive and preemptive algorithms:

- **Non-preemptive** algorithms are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time,
- **Preemptive** scheduling is based on priority where a scheduler may preempt a low priority running process anytime when a high priority process enters a ready state.
- Both have advantage and disadvantage, we need to select according to our applications

Department of Electrical Engineering
香港城市大學
City University of Hong Kong

1. Introduction

Before introducing the algorithms. There are some important concepts in this tutorial.

- **Arrival time:** Time at which the process arrives in the ready queue.
- **Completion Time:** Time at which process completes its execution.
- **Burst Time:** Time required by a process for CPU execution. (Or named as execute time)
- **Turn Around Time:** Time Difference between completion time and arrival time. (equals to Completion Time – Arrival Time)
- **Service time:** Time at which the queue begins the process
- **Wait time:** equals to Turn Around Time – Burst time. (Or named as wait time)

Department of Electrical Engineering
香港城市大學
City University of Hong Kong

1. Introduction

For the objectives of selecting process scheduling algorithms:

- Max CPU utilization [Keep CPU as busy as possible]
- Fair allocation of CPU.
- Min turn around time [Time taken by a process to finish execution]
- Min waiting time [Time a process waits in ready queue]
- Max throughput [Number of processes that complete their execution per time unit]

Department of Electrical Engineering
香港城市大學
City University of Hong Kong

1. Introduction

Based on the objectives, in this tutorial, we mainly introduce 6 popular process scheduling algorithms to be assigned to the CPU.

- First-Come, First-Served (FCFS) Scheduling
- Shortest-Job-First (SJF) Scheduling
- Shortest Remaining Time
- Fixed Priority Scheduling
- Round Robin(RR) Scheduling
- Multiple-Level Queues Scheduling

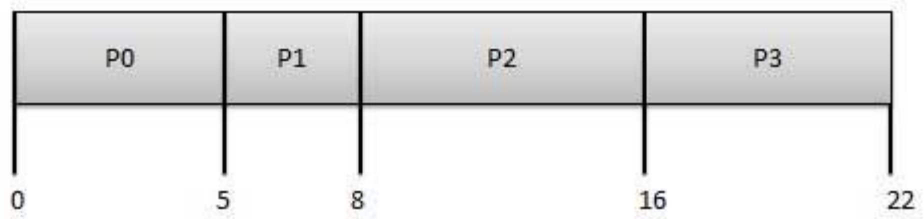
Department of Electrical Engineering
香港城市大學
City University of Hong Kong

2.1 First-Come, First-Served (FCFS) Scheduling

For this algorithm,

- Jobs are executed on first come, first serve basis.
- It is a **non-preemptive** algorithm.
- Easy to understand and implement.
- Poor in performance as average wait time is high

Process	Arrival Time	Burst Time	Service Time	Wait Time
P0	0	5	0	0
P1	1	3	5	4
P2	2	8	8	6
P3	3	6	16	13



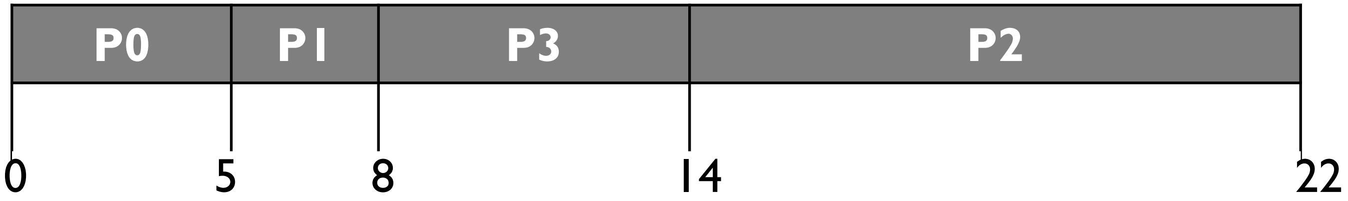
Average Wait Time: (0+4+6+13) / 4 = 5.75

CityU Department of Electrical Engineering
香港城市大學
City University of Hong Kong

2.2 Shortest-Job-First (SJF) Scheduling

- This is a **non-preemptive** 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.

Process	Arrival Time	Burst Time	Service Time	Wait Time
P0	0	5	0	0
P1	1	3	5	4
P2	2	8	14	12
P3	3	6	8	5

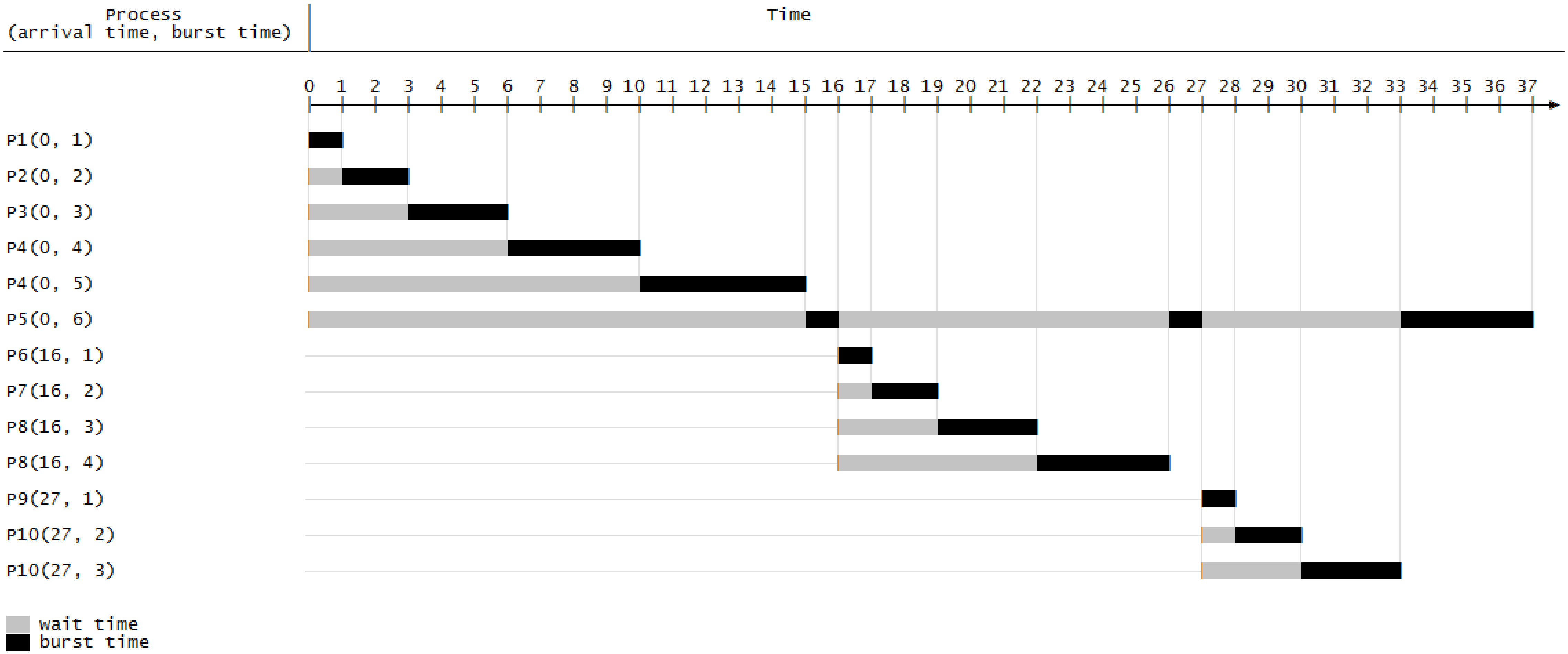


Average Wait Time: (0 + 4 + 12 + 5)/4 = 21 / 4 = 5.25

2.3 Shortest Remaining Time (SRT) Algorithm

- Shortest remaining time (SRT) is the **preemptive** version of the SJF algorithm.
- The processor is allocated to the job closest to completion but it can be preempted by a newer arrived job with shorter time to completion.
- Impossible to implement in interactive systems where required CPU time is not known.
- It is rarely used outside of specialized environments
- It is often used in batch environments where short jobs need to give preference.

2.3 Shortest Remaining Time (SRT) Algorithm



Department of Electrical Engineering
CityU
香港城市大學
City University of Hong Kong

2.3 Shortest Remaining Time (SRT) Algorithm

- **Questions**: Please finish the table, draw the process figure following the format in page 9 and calculate average waiting time based on following data.

Process	Arrival Time	Burst Time	Service Time	Wait Time
P0	0	8		
P1	1	3		
P2	2	6		
P3	3	5		

2.4 Fixed Priority Based Scheduling Algorithm

- Priority scheduling is a **non-preemptive** algorithm and one of the most common scheduling algorithms in batch systems.
- Each process is assigned a fixed 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

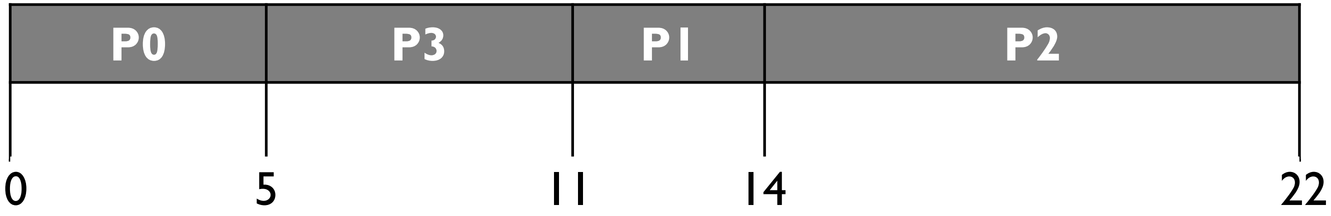
Process	Arrival Time	Burst Time	Priority	Service Time	Wait Time
P0	0	5	1	0	0
P1	1	3	2	11	10
P2	2	8	1	14	12
P3	3	6	3	5	2

Given: Table of processes, and their Arrival time, Execution time, and priority. Here we are considering 1 is the lowest priority

Department of Electrical Engineering
香港城市大學 City University of Hong Kong

2.4 Fixed Priority Based Scheduling Algorithm

Process	Arrival Time	Burst Time	Priority	Service Time	Wait Time
P0	0	5	1	0	0
P1	1	3	2	11	10
P2	2	8	1	14	12
P3	3	6	3	5	2



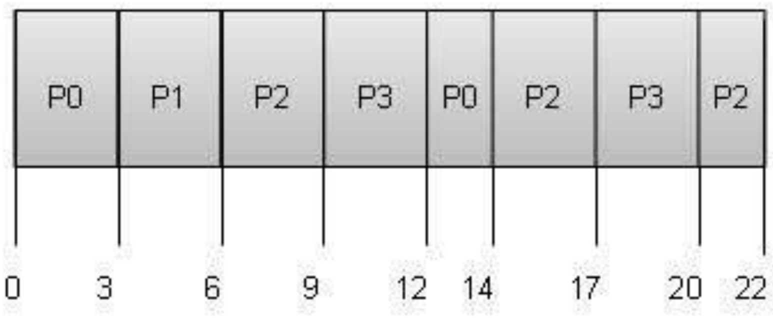
Average Wait Time: (0 + 10 + 12 + 2)/4 = 24 / 4 = 6

2.5 Round Robin Scheduling Algorithm

- 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.

Process	Arrival Time	Burst Time	Completion Time	Turn Around Time	Wait Time
P0	0	5	14	14	9
P1	1	3	6	5	2
P2	2	8	22	20	12
P3	3	6	20	17	11

Quantum = 3



Average Wait Time: (9+2+12+11) / 4 = 8.5

Department of Electrical Engineering
香港城市大學
City University of Hong Kong

2.6 Multiple-Level Queues Scheduling

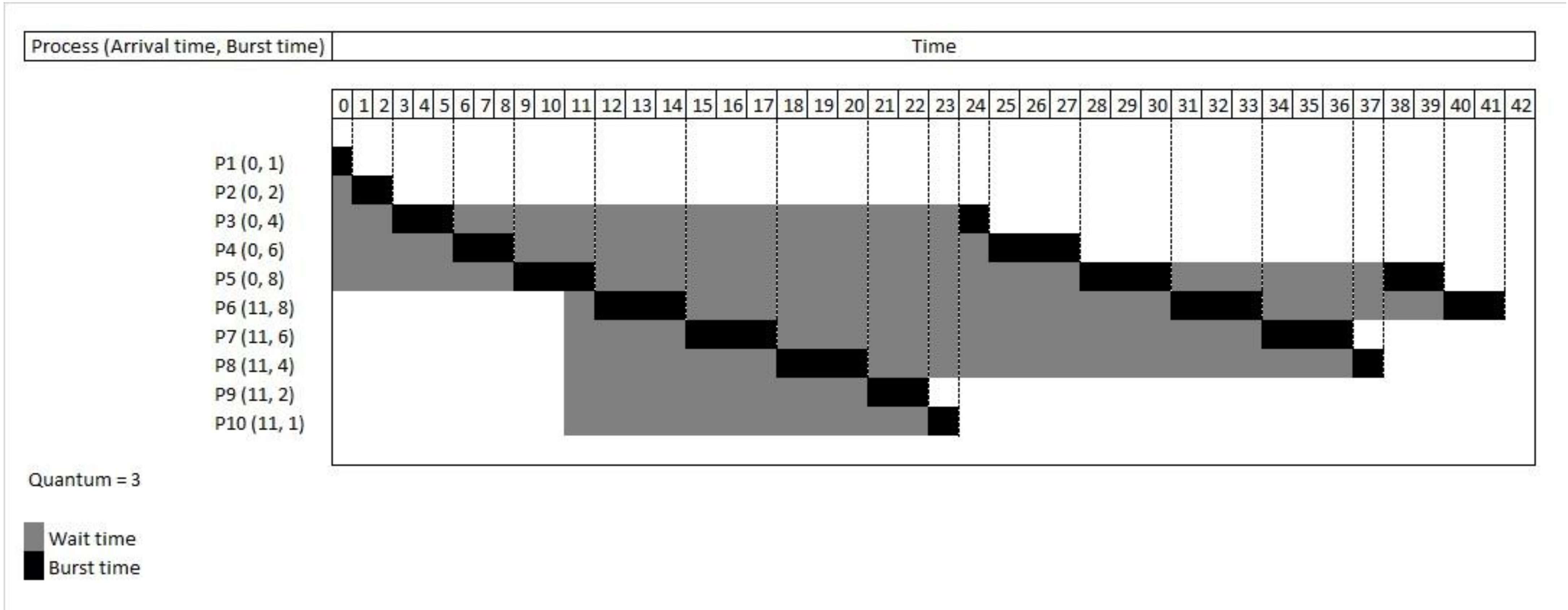
Multiple-level queues are not an independent scheduling algorithm. They make use of other existing algorithms to group and schedule jobs with common characteristics.

- 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

3. Conclusion

- Some features of algorithms:
  - FCFS can cause long waiting times, especially when the first job takes too much CPU time.
  - Both SJF and Shortest Remaining time first algorithms may cause starvation. Consider a situation when the long process is there in the ready queue and shorter processes keep coming.
  - If time quantum for Round Robin scheduling is very large, then it behaves same as FCFS scheduling.
  - SJF is optimal in terms of average waiting time for a given set of processes, i.e., average waiting time is minimum with this scheduling, but problems are, how to know/predict the time of next job.
- Besides, these are also other useful algorithms can be used according to our applications. Please learn the features of each algorithm and find the most suitable algorithm for your application.

4. Exercise



Calculate the average waiting time according to the figure of process and distinguish what kind of algorithm it is.

5. Reference

- https://www.tutorialspoint.com/operating_system/os_process_scheduling_algorithms.htm
- https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/
- https://en.wikipedia.org/wiki/Scheduling_(computing)