TA responsible for HW2: Feiyuan ZHANG (fzhangax@cse.ust.hk)

Fall 2024 COMP 3511 Homework Assignment #2

Handout Date: October 7, 2024 (Monday), Due Date: October 21, 2024 (Monday)

Name	LI,Yuntong
Student ID	20944800
ITSC email	ylino@connect.ust.hk

Please read the following instructions carefully before answering the questions:

- You must finish the homework assignment **individually**.
- This homework assignment contains **three** parts: (1) multiple choices, (2) short answer and (3) CPU scheduling
- **Homework Submission:** Please submit your homework to **Homework #2** on **Canvas**.
- TA responsible for HW2: Feiyuan ZHANG (fzhangax@cse.ust.hk)

1. (30 points) Multiple Choices

Write your answers in the boxes below:

MC1	MC2	MC3	MC4	MC5	MC6	MC7	MC8	MC9	MC10
D	B	A	C	B	B	C	D	B	A

(1) Which of the following resources cannot be shared across all the threads in the same process P?

A) The global variables.

B) An open file in P.

C) The text.

D) The stack memory.

(2) Which of the following statements on pipe communication is TRUE?

A) Named pipes are automatically deleted after the communication ends.

B) Communications are uni-directional for ordinary pipes

C) Named pipes cannot be used among a parent and child process.

D) Ordinary pipes can be used by communicating processes on different machines.

(3) If an application is 50% parallel and 50% serial, which of the following values is a possible speedup when moving the application from one core to four cores, according to Amdahl's Law?

A) 1.6

B) 2.0

C) 2.4

D) 2.8

TA responsible for HW2: Feiyuan ZHANG (fzhangax@cse.ust.hk)

(4) Which of the following statements is NOT TRUE between user threads and kernel threads?

A) User threads are only visible to programs

B) A user thread needs to be mapped a kernel thread before execution

C) Many-to-one mapping has been widely adopted by modern OS

D) OS manages kernel threads including allocating the required resources

(5) Which of the following process scheduling algorithm may lead to starvation?

A) FIFO

B) Shortest Job First

C) Round Robin

D) None of the above

(6) Which of the following process scheduling is considered non-preemptive?

1. Switches from running to waiting
2. Switches from running to ready
3. Switches from waiting to ready
4. Process terminates

A) 1 and 2

B) 1 and 4

C) 2 and 3

D) 1, 2 and 3

(7) Which of the following statement about the time quantum in RR algorithm is TRUE?

A) The time quantum can be as small as possible to achieve high average response time

B) The RR algorithm never suffers from convoy effect no matter how to set the time quantum

C) The average job turn-around time of RR might be worse than FCFS no matter how we set the time quantum

D) None of the above

(8) Which of the following statement on MLFQ scheduling is TRUE?

A) It is fair in the sense that all CPU-bound processes can make progress

B) It might deliver better performance than RR

C) Its performance resembles SJF and SRTF scheduling without the need to estimate the next CPU burst time

D) All of the above

(9) Under symmetric multiprocessing, or SMP, what is the advantage of using per-core ready queue,i.e., each CPU core has its own ready queue(s)?

A) It provides better CPU utilization in all cases

B) It naturally provides processor affinity

C) It makes CPU scheduling easier

D) None of the above

(10) Which of the following statement of rate-monotonic and EDF scheduling is TRUE?

A) A static priority is assigned in rate-monotonic scheduling

B) The rate-monotonic scheduling algorithm follows anon-preemption method

C) EDC scheduling uses static priority

D) Both rate-monotonic and EDF scheduling require processes to be periodic

2. (30 points) Please answer the following questions in a few sentences

(1) (5 points) Why is a thread referred to as a lightweight process?

A thread is considered a lightweight process because it shares the memory and resources of its parent process while having its own execution context, including a stack, program counter, and registers. This shared environment enables threads to be more efficient in creation and context switching than full processes, which involve greater overhead for memory management. As a result, threads are better suited for tasks that demand high concurrency with lower resource usage.

(2) (5 points) Please describe data parallelism and task parallelism.

Data parallelism is a form of parallel computing where the same operation is applied simultaneously across multiple data elements. This approach splits large datasets into smaller chunks that can be processed independently.

Task parallelism, on the other hand, involves distributing different tasks or operations, which may be completely independent, across multiple threads or processors. Each task can perform different computations or functions, making it suitable for applications with distinct concurrent processes.

(3) (5 points) Explain the concept of context switching and further explain why the overhead is more if switching to a thread belong to other processes?

Context switching is the process where the CPU switches from executing one thread or process to another. This involves saving the state of the currently running thread and loading the saved state of the next thread to execute.

The overhead increases when switching to threads in different processes because the operating system must save and restore the thread state and switch the entire memory context of the processes(address space will change) . This includes updating memory management structures and invalidating outdated caching data, resulting in greater time consumption compared to switching between threads within the same process.

TA responsible for HW2: Feiyuan ZHANG (fzhangax@cse.ust.hk)

(4) (5 points) Please describe the pros and cons of one-to-one thread mapping scheme.

Pros: Each user thread can be scheduled individually by the kernel, which allows for better CPU utilization and responsiveness. Each thread has its own kernel thread, simplifying management; all threads can take advantage of multiprocessor systems.

Cons: Each kernel thread requires additional resources, leading to higher memory consumption and possibly more overhead when creating and destroying threads. If the number of user threads exceeds the number of kernel threads the operating system can manage, it could lead to inefficiencies and increased context switching overhead.

(5) (5 points) Please describe the concept of CPU utilization, waiting time, turnaround time, response time and throughput.

CPU Utilization: The percentage of time the CPU is actively working on tasks versus being idle; higher utilization indicates better resource usage.

Waiting Time: The total time a process spends in the ready queue waiting for CPU time to execute. It's a key component of performance metrics, especially in scheduling algorithms.

Turnaround Time: The total time taken from the submission of a process to its completion. It encompasses waiting time, execution time, and any I/O operations.

Response Time: The time between a user request being submitted and the first response being produced by the system. It is crucial for interactive systems where quick responses are necessary.

Throughput: The number of processes or tasks completed in a unit of time. It's a measure of how effectively a system processes workloads.

(6) (5 points) Please describe the advantages in MLFQ scheduling.

1.Fairness: It provides good response times to interactive processes by giving preferential treatment to shorter tasks, which can reduce turnaround time for user interactions.

2.Flexibility: MLFQ can adapt to the behavior of processes by moving them between queues based on their observed execution characteristics, allowing it to optimize for both short and long processes.

3.Prioritization: By utilizing multiple queues with different priority levels, MLFQ ensures CPU-bound and I/O-bound processes get appropriate treatment, enhancing overall system responsiveness and throughput.

4.Efficiency: The dynamic nature of MLFQ allows it to efficiently utilize CPU while minimizing starvation of tasks, tackling one of the significant issues found in other static scheduling algorithms.

TA responsible for HW2: Feiyuan ZHANG (fzhangax@cse.ust.hk)

3. (40 points) CPU Scheduling.

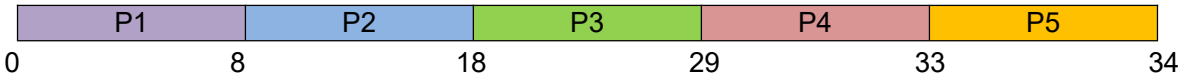
(1) (20 points) Consider the following single-thread process, arrival times, and CPU process requirements:

Process	Arrival Time	Burst Time
P_1	0	8
P_2	3	10
P_3	7	11
P_4	12	4
P_5	13	1

- We consider 4 algorithms: FCFS, RR, SJF, SRTF (preemptive).
- The time quantum of the Round-Robin (RR) is 4.
- Assume that context switch overhead is 0.
- When a process arrives, it is immediately eligible for scheduling, e.g., process 2 that arrives at time 2 can be scheduled during time unit 2.
- Whenever there is a tie among processors (same arrival time, same remaining time, etc), they are inserted into the ready queue in the ascending order of process id. That is, if process 1 and process 2 arrive at the same time, process 1 is inserted first, and process 2 second in the ready queue.

For each scheduling algorithm, draw the Gantt charts depicting the sequence of the process execution, and calculate the average turnaround time and average waiting time.

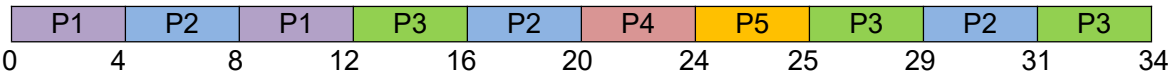
1.FCFS



Process	Waiting	Turnaround
P1	0-0 = 0	8-0 = 8
P2	8-3 = 5	18-3 = 15
P3	18-7 = 11	29-7 = 22
P4	29-12 = 17	33-12 = 21
P5	33-13 = 20	34-13 = 21

Average turnaround time = (8+15+22+21+21) / 5 = 17.4
Average waiting time = (0+5+11+17+20) / 5 = 10.6

2.RR



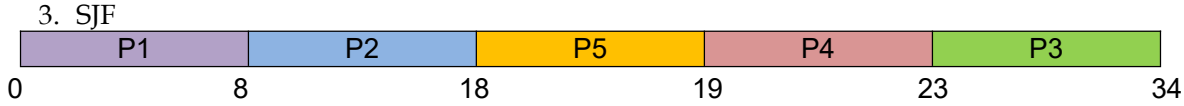
Process	Waiting	Turnaround
P1	(0-0)+(8-4)=4	12
P2	(4-3)+(16-8)+(29-20)=18	28
P3	(12-7)+(25-16)+(31-29)=16	27
P4	(20-12)=8	12
P5	(24-13)=11	12

Average turnaround time = (12+28+27+12+12) / 5 = 18.2
Average waiting time = (4+18+16+8+11) / 5 = 11.4

TA responsible for HW2: Feiyuan ZHANG (fzhangax@cse.ust.hk)

(continue your answers for Q3.1)

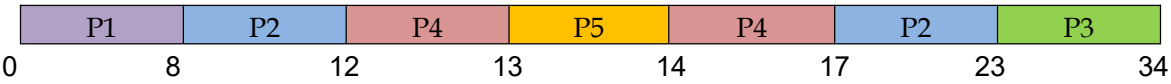
3. SJF



Process	Waiting	Turnaround
P1	(0-0)=0	8
P2	(8-3)=5	15
P3	(23-7)=16	27
P4	(19-12)=7	11
P5	(18-13)=5	6

Average turnaround time = (8+15+27+11+6) / 5 = 13.4
Average waiting time = (0+5+16+7+5) / 5 = 6.6

4. SRTF



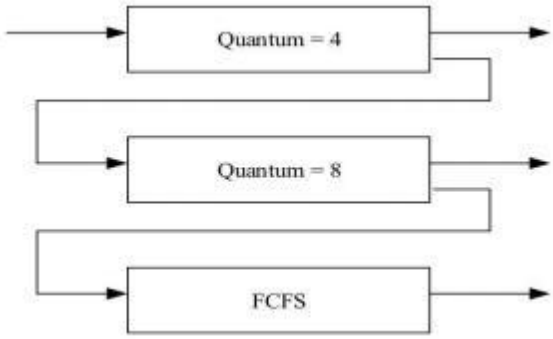
Process	Waiting	Turnaround
P1	(0-0)=0	8
P2	(8-3)+(17-12)=10	20
P3	(23-7)=16	27
P4	(12-12)+(14-13)=1	5
P5	(13-13)=0	1

Average turnaround time = (8+20+27+5+1) / 5 = 12.2
Average waiting time = (0+10+16+1+0) / 5 = 5.4

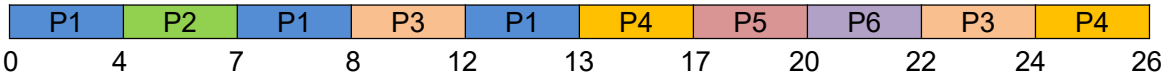
(2) (20 points) **Multi-Level Feedback Queue**

Draw Gantt charts for a MLFQ scheduling and compute the average turnaround time and the average waiting time. The three queues are defined as follows.

1) Q0: RR with time quantum 4
2) Q1: RR with time quantum 8
3) Q2: FCFS



Process	Arrival Time	Burst Time
P1	0	6
P2	1	3
P3	8	6
P4	13	6
P5	16	3
P6	20	2



Process	Waiting	Turnaround
P1	(0-0)+(7-4)+(12-8)=7	13
P2	(4-1)=3	6
P3	(8-8)+(22-12)=10	16
P4	(13-13)+(24-17)=7	13
P5	(17-16)=1	4
P6	(20-20)=0	2

Average turnaround time = (13+6+16+13+4+2) / 6 = 9
Average waiting time = (7+3+10+7+1+0) / 6 = 4.666.... = 4.67