CSC 4320: Operating Systems — 55 Points

Homework 3

© Instructor: Dr. Md. Mahfuzur Rahman — **Spring 2025**

Work on the following problems and submit your own answers. You are allowed to discuss with other students. However, do not copy the solutions from peers or other sources. If the assignment has programming component, your program(s) must compile with **gcc** and execute on **snowball.cs.gsu.edu**! Please see https://cscit.cs.gsu.edu/sp/guide/snowball for more details.

Instructions:

- Upload an electronic copy (MS word or pdf) of your answer sheet to the folder named "HW3" in iCollege.
- Please add the course number, homework number, and your name at the top of your answer sheet.
- Please write down your answers with the question number only in the answer sheet.
- Also submit your .c file (c program), if you are asked to write a program.
- Name your file in the format of CSC4320_HW3_FisrtnameLastname (.docx/.pdf)
- Deadline: Submit by March 08, 2025, 11:59 pm

1. (5 points) Write down the Amdahl's law and describe its parameters. Using Amdahl's Law, calculate the speedup gain of an application that has a 90 percent parallel with (a) four processing cores and (b) eight pro-cessing cores.

2. What (if any) relation holds between the following pairs of algorithm sets?
   (a) (2 points) Priority and SJF
   (b) (2 points) Multilevel feedback queues and FCFS
   (c) (2 points) Priority and FCFS
   (d) (2 points) RR and SJF

3. Consider the following code segment:

```
pid_t pid;
pid = fork();
if (pid == 0) { /* child process */
    fork();
    thread_create( . . .);
}
fork();
```

   (a) (5 points) How many unique processes are created? Explain in detail.
   (b) (5 points) How many unique threads are created? Explain in detail.

4. (6 points) Consider the following program using Pthreads API:

```
#include <pthread.h>
#include <stdio.h>

int value = 0;
```

```
void *runner(void *param); /* the thread */

int main(int argc, char *argv[])
{
   pid_t pid;
   pthread_t tid;
   pthread_attr_t attr;
   pid = fork();
   if (pid == 0) { /* child process */
      pthread_attr_init(&attr);
      pthread_create(&tid,&attr,runner,NULL);
      pthread_join(tid,NULL);
      printf("CHILD: value = %d",value); /* LINE C */
   }
   else if (pid > 0) { /* parent process */
      wait(NULL);
      printf("PARENT: value = %d",value); /* LINE P */
   }
}
void *runner(void *param) {
   value = 5;
   pthread_exit(0);
}
```

Describe what is happening in the program. What would be the output from the program at LINE C and LINE P?

5. Consider the following set of processes, with the length of the CPU burst time given in milliseconds:

Process	Burst Time	Priority
P1	5	4
P2	3	1
P3	1	2
P4	7	2
P5	4	3

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

(a) (4 points) Draw four Gantt charts that illustrate the execution of these pro-cesses using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).

(b) (4 points) What is the turnaround time of each process for each of the scheduling algorithms in part a?

(c) (4 points) What is the waiting time of each process for each of these scheduling algorithms?

(d) (4 points) Which of the algorithms results in the minimum average waiting time (over all processes)?
Please provide explanation of your work for each of the sub-questions a), b), c), d).

6. Consider two processes, P1 and P2,where p1 = 50, t1 = 25, p2 = 75, and t2 = 30.

(a) ($2\frac{1}{2}$ points) Can these two processes be scheduled using rate-monotonic scheduling? Illustrate your answer using a Gantt chart.

(b) ($2\frac{1}{2}$ points) Illustrate the scheduling of these two processes using earliest-deadline-first (EDF) scheduling.

7. (5 points) Consider a system running ten I/O-bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context-switching overhead is 0.1millisecond and that all processes

are long-running tasks. Describe the CPU utilization for a round-robin scheduler when the time quantum is 10 milliseconds .

Question:	1	2	3	4	5	6	7	Total
Points:	5	8	10	6	16	5	5	55
Bonus Points:	0	0	0	0	0	0	0	0
Score:								