Name: Sean Losseff RCS ID: losses @rpi.edu

CSCI 4210 — Operating Systems
Spring 2020 Quiz 4 (March 3, 2020)

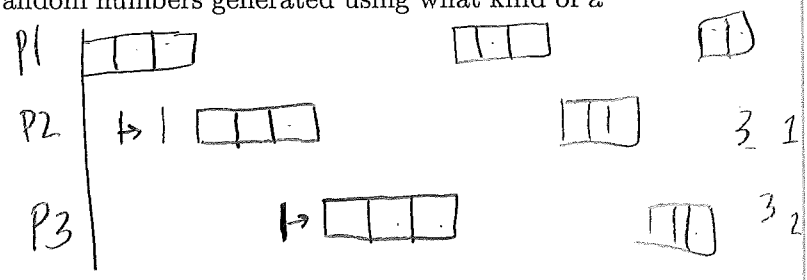
- Please silence and put away all laptops, notes, books, phones, electronic devices, etc.
- This quiz is designed to take 25 minutes; therefore, for 50% extra time accommodations, the expected time is 38 minutes and 100% extra time accommodations is 50 minutes (i.e., the end of class).
- Questions will not be answered except when there is a glaring mistake or ambiguity in a question. Please do your best to interpret and answer each question.

1. **(2 POINTS)** What can the `shmat()` system call be used for? Circle the best answer.

   (a) Attach to the shared memory segment given a shared memory key
   (b) Obtain the shared memory ID given a shared memory key
   (c) Create a shared memory segment with the designated size and permissions
   (d) Both (a) and (b), but not (c)
   (e) Both (a) and (c), but not (b)
   (f) Both (b) and (c), but not (a)

2. **(2 POINTS)** `drand48()` generates pseudo-random numbers generated using what kind of a distribution? Circle the best answer.

   (a) gaussian
   (b) Poisson
   (c) exponential
   (d) uniform



3. **(9 POINTS)** Apply the Round Robin (RR) scheduling algorithm to the process CPU bursts described in the table below. Use a timeslice of 3 ms. And for each process, calculate the individual turnaround time, wait time, and number of preemptions experienced by each given process. Ignore context switch times and other such overhead.

	CPU Burst	Arrival	Turnaround Time	Wait Time	Preemptions
P1	8 ms	0 ms	20	12	2
P2	10 ms	1 ms	26	16	3
P3	11 ms	5 ms	24	13	3

4. (a) **(4 POINTS)** What is the **exact** terminal output for the code below? If multiple outputs are possible, please use a diagram to show all possibilities. Note that there are no compilation warnings or errors in the given code. The `#include` directives are omitted to save space on the page. Assume that all system calls succeed. Also assume that there is no shared memory segment with key 9000 when this code is executed for the first time.

```
#define SHM_SHARED_KEY 9000

int main( int argc, char * argv[] )
{
  /* create the shared memory segment with a size of 4 bytes */
  int shmid = shmget( SHM_SHARED_KEY, sizeof( int ), IPC_CREAT | 0660 );
  printf( "shmget() returned %d\n", shmid );

  /* attach to the shared memory segment */
  int * data = shmat( shmid, NULL, 0 );

  int pid = fork();
  if ( pid > 0 ) waitpid( pid, NULL, 0 );

  int i, stop = 6;
  for ( i = 1 ; i <= stop ; i++ ) *data += i;

  printf( "%s: Sum is %d\n", pid > 0 ? "PARENT" : "CHILD", *data );
  shmdt( data );

  if ( pid > 0 ) shmctl( shmid, IPC_RMID, 0 );

  return EXIT_SUCCESS;
}
```

CHILD: Sum is 21
PARENT: Sum is 42

(b) **(3 POINTS)** What is the **exact** terminal output for the code above if it is executed a second time?

CHILD: sum is 21
PARENT: Sum is 42