CS 3305: Operating Systems
Department of Computer Science
The University of Western Ontario
Midterm
Spring, 2013

NAME : _____________________________________________________________

STUDENT NUMBER : ________________

This is a closed book exam. You have 100 minutes to complete 21 questions. Please write neatly and clearly. You should have 10 pages.

<table>
<thead>
<tr>
<th>Question</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-17</td>
<td>/34</td>
</tr>
<tr>
<td>18</td>
<td>/10</td>
</tr>
<tr>
<td>19</td>
<td>/24</td>
</tr>
<tr>
<td>20</td>
<td>/20</td>
</tr>
<tr>
<td>21</td>
<td>/12</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>/100</strong></td>
</tr>
</tbody>
</table>

Score : _____________ / 100
Multiple Choice

1. (2 points) Which of the following would lead you to believe that a given system is an SMP-type system?
   a) Each processor is assigned a specific task.
   b) There is a master–slave relationship between the processors.
   c) Each processor performs all tasks within the operating system.
   d) None of the above

2. (2 points) A _____ is an example of an operating systems service.
   a) command line
   b) web browser
   c) text formatter
   d) database system

3. (2 points) Which of the following statements is incorrect?
   a) An operating system provides an environment for the execution of programs.
   b) An operating system manages system resources.
   c) Operating systems must provide web browsers.
   d) Operating systems must provide both protection and security.

4. (2 points) The ____ of a process contains temporary data such as function parameters, return addresses, and local variables.
   a) text section
   b) data section
   c) program counter
   d) stack

5. (2 points) A process control block ____.
   a) includes information on the process's state
   b) stores the address of the next instruction to be processed by a different process
   c) determines which process is to be executed next
   d) is an example of a process queue

6. (2 points) When a child process is created, which of the following is a possibility in terms of the execution or address space of the child process?
   a) The child process runs concurrently with the parent.
   b) The child process has a new program loaded into it.
   c) The child is a duplicate of the parent.
   d) All of the above
7. (2 points) When a process creates a new process using the `fork()` function, which of the following is shared between the parent process and the child process?
   a) Stack
   b) Heap
   c) Text
   d) All of the above

8. (2 points) A process may transition to the Ready state by which of the following actions?
   a) Completion of an I/O event
   b) Awaiting its turn on the CPU
   c) Newly-admitted process
   d) All of the above

9. (2 points) ____ scheduling is approximated by predicting the next CPU burst with a weighted average of the measured lengths of previous CPU bursts.
   a) Multilevel queue
   b) Round-robin
   c) First-come-First-Serve
   d) Shortest-Job-First

10. (2 points) Which of the following scheduling algorithms is nonpreemptive?
    a) Shortest-Job-First
    b) Round-robin
    c) First-come-First-Serve
    d) Priority algorithms

11. (2 points) The default scheduling class for a process in Solaris is ____.
    a) time sharing
    b) system
    c) interactive
    d) real-time

12. (2 points) A significant problem with priority scheduling algorithms is ____.
    a) complexity
    b) starvation
    c) determining the length of the next CPU burst
    d) determining the length of the time quantum
True/False

13. (2 points) A system call is triggered by hardware (T/F)
14. (2 points) Interrupts may be triggered by either hardware or software (T/F)
15. (2 points) Virtually all modern operating systems provide support for SMP (T/F)
16. (2 points) System calls can be run in either user mode or kernel mode (T/F)
17. (2 points) In Round-Robin scheduling, the time quantum should be small with respect to the context-switch time (T/F)

18. System Calls (10 points)
   a) (4 points) Describe the relationship between an API, the system-call interface, and the operating system.

   b) (6 points) Describe three general methods used to pass parameters to the operating system during system calls.
19. Multiprogramming and Scheduling (24 points)

a) (6 points) Provide three reasons why a process may leave the Ready state?

b) (3 points) Explain the concept of a CPU–I/O burst cycle.

c) (5 points) In Windows 7, how does the dispatcher determine the order of process execution?

d) (5 points) Why does Solaris assign a lower priority for a process that finishes its time slice?

e) (3 points) In Linux how is the average sleep time used to determine if a process is interactive?

f) (2 points) When does Linux re-calculate priority for a user process?
20. (20 points) Processes, fork, pipes

a) (5 points) How many processes does the following program create?

```c
int main(void) {
    pid = fork();
    if (pid != 0)
        fork();
    if (pid == 0)
        fork();
}
```

Answer:_____

b) (5 points) Consider the following program. What is a possible output?

```c
void main() {
    pid_t pid;

    printf("Hello there\n");
    pid = fork();
    printf("What is up?\n");

    if (pid == 0) {
        printf("Nothing\n");
    } else {
        wait(NULL);
        printf("That is a relief\n");
    }
}
```

Answer:
c) (4 points) Consider the following program. What should the values be at the question marks on lines A, B, C and D?

```
#include <unistd.h>
#include <stdio.h>
int main(void) {
  
  int n;
  int fd[2];
  pid_t pid;
  char line[80];

  if (pipe(fd) < 0)
    perror("pipe error");

  if ((pid = fork()) < 0) {
    perror("fork error");
  } else if (pid == 0) {
    close(fd[?]); /*Line A*/
    write(fd[?], "hello world\n", 12); /* Line B*/
  } else if (pid > 0) {
    close(fd[?]); /*Line C*/
    n = read(?, line, 80); /*Line D*/
    write(1, line, n);
  }
}
```

Line A: _______________ Line B: _______________
Line C: _______________ Line D: _______________
d) (6 points) Consider the following program. How would you fill in the question marks in `dup2()` if you want the output of “ls” to go to “foobar.txt”?

```c
int main(void){
    FILE * fd1;
    char * prog1_argv[4];

    prog1_argv[0] = "ls";
    prog1_argv[1] = NULL;

    fd1 = open("foobar.txt", "w");
    dup2(?,
    execvp(prog1_argv[0], prog1_argv);
    exit(0);
}
```

Answer: The first question mark should be _______ and the second question mark should be _______.
21. (12 points) Assuming no context switching time, the following table consists of a set of jobs to be processed on a single CPU.

<table>
<thead>
<tr>
<th>Job</th>
<th>Burst</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

e) (4 points) If a round-robin scheduling algorithm with a time slice of 2 milliseconds is assumed, then at approximately what time is job 3 completed?

f) (4 points) Compute the waiting time of job 2 assuming that the scheduling policy is first-come first serve.

g) (4 points) What is the order of process execution assuming that the scheduling policy is shortest-job first, there is no pre-emption and no requests for I/O?
This page left intentionally blank