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

NAME : Answer Key

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.

Question	**Grade**
1-17	____/34
18	____/10
19	____/24
20	____/20
21	____/12
Total	____/100

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.

    Ans: The system-call interface of a programming language serves as a link to system calls made available by the operating system. This interface intercepts function calls in the API and invokes the necessary system call within the operating system. Thus, most of the details of the operating-system interface are hidden from the programmer by the API and are managed by the run-time support library.

    b) (6 points) Describe three general methods used to pass parameters to the operating system during system calls.

    Ans: The simplest approach is to pass the parameters in registers. In some cases, there may be more parameters than registers. In these cases, the parameters are generally stored in a block, or table, of memory, and the address of the block is passed as a parameter in a register. Parameters can also be placed, or pushed, onto the stack by the program and popped off the stack by the operating system.

19. Multiprogramming and Scheduling (24 points)

a) (6 points) Provide three reasons why a process may leave the Running state?
Ans: Terminates, Needs I/O, Its quantum is up, Device interrupt

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

Ans: The lifecycle of a process can be considered to consist of a number of bursts belonging to two different states. All processes consist of CPU cycles and I/O operations. Therefore, a process can be modeled as switching between bursts of CPU execution and I/O wait.

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

Ans: The dispatcher uses a 32-level priority scheme to determine the execution order. Priorities are divided into two classes. The variable class contains threads having priorities from 1 to 15, and the real-time class contains threads having priorities from 16 to 31. The dispatcher uses a queue for each scheduling priority, and traverses the set of queues from highest to lowest until it finds a thread that is ready to run. The dispatcher executes an idle thread if no ready thread is found.

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

Ans I/O bound processes spend most of their time waiting for data. When the CPU is needed it is feasible to give I/O bound processes highest priority. If a CPU bound process is assigned to the highest-priority queue then it is likely to consume the short time quantum. This is a good indicator that the process is CPU-bound. CPU-bound processes are given a lower priority and will not compete with I/O bound processes for the highest priority queue.

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

Ans: Interaction requires I/O and I/O usually means that the process sleeps more.

f) (2 points) When does Linux re-calculate priority for a user process?
Ans: When it is moved from the active array to the expired array.

20. (20 points) Processes, fork, pipes

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

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

Answer:___4__ Note: Give 2 points if answer is 3

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

```
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:

Hello there
What is up?
What is up?
Nothing
That is a relief

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:____0_____ Line B: _____1_____

Line C:____1_____ Line D: _____0_____

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"?

```
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 ___fd1____ and the second question mark should be __STDOUT ____.

21. (12 points) Assuming no context switching time, the following table consists of a set of jobs to be processed on a single CPU.

Job	Burst	Arrival
1	11	0
2	3	3
3	6	6
4	9	9

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?

Each line represents process id, time, old state and new state.

Time	0	1	2	3	4	5	6	7	8	9	10	11	12
Process running	1	1	1	1	2	2	1	1	2	3	3	1	1

Time	13	14	15	16	17	18	19	20	21	22	23	11	12
Process running	4	4	3	3	1	1	4	4	3	3			

If you assumed that 1 starts running at time 1 then 3 finishes at time 23.

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

Ans: Process 1 finishes at time 11. Process 2 arrives at time 3. 11-3 represents the waiting time.

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?

Ans: 1 2 3 4

Process 2 is the shortest but it arrives after process 1. This means that process 1 starts first.

This page left intentionally blank