



**University of Ottawa**  
**School of Information Technology and**  
**Engineering**

**Course:** CSI3131 – Operating Systems  
**SEMESTER:** Winter 2015

**Professor:**  
**Date:**  
**Hour:**  
**Room:**

Miguel A. Garzón  
Feb 26 2015  
14:30-16:00 (90 minutes)  
MRT 219

**Midterm Exam**

NAME: \_\_\_\_\_

STUDENT NUMBER: \_\_\_\_\_

**Please answer all questions on the questionnaire.**

**The exam consists of three (3) parts:**

<b>Part 1</b>	<b>Short Questions</b>	<b>10 points</b>	
<b>Part 2</b>	<b>Theory Questions</b>	<b>16 points</b>	
<b>Part 3</b>	<b>Application Questions</b>	<b>24 points</b>	
<b>Bonus</b>		<b>2 points</b>	
<b>Total</b>		<b>50 points + 2</b>	

## Part 1: Short Answer Questions (10 questions, each for one point):

1. Scheduling algorithms can starve a process by not allowing it to access the CPU when many processes are executing. Name a scheduling algorithm which can cause starvation?
2. Give an IPC mechanism that allows communication between two processes running on different computer systems?
3. In the context of Inteprocess communication, briefly describe what is a 'rendezvous'?
4. What is the name of the scheduler that is responsible for swapping processes?
5. Give an advantage of many to one thread model over the one to one thread model.
6. What are the two approaches to creating threads in Java?
7. Deferred cancellation is used to postpone the cancellation of a thread until its parent is ready to accept the exit value.

Is this statement TRUE ☐ or FALSE ☐ ?

8. On a multi-CPU computer, two threads of the same process in a pure user thread environment (many to one model) can be executed on different CPUs.

Is this statement TRUE ☐ or FALSE ☐ ?

9. Which of the following resources are shared by threads within the same process:
  - a. references to pipes
  - b. the program counter
  - c. program code
  - d. local variables on the stack
  - e. global variables
10. What are the three requirements that must be satisfied by valid solutions to the critical section problem?

## **Part 2: Theory Questions (16 points, answer in the provided space)**

**Question 1. (3 points)** List all states to which a process can directly move from the *running* state. Describe the circumstances under which these transitions occur. (Give at least 2 examples for each state.)

**Question 2. (3 points)** Briefly explain what is DMA and how it works. Why is it used?

**Question 3 (3 points)** Give a short definition of each of the 5 scheduling criteria. Place the criteria beneath the appropriate titles below.

Criteria to maximize:

Criteria to minimize:

**Question 4 (4 points)** Explain the main advantage and disadvantage of

a. many-to-one mapping of user threads to kernel threads

➤ Advantage:

➤ Disadvantage:

b. one-to-one mapping of user threads to kernel threads

➤ Advantage:

➤ Disadvantage:

**Question 5 (3 points).** Explain the difference between symmetric multiprocessing and asymmetric multiprocessing.

### Part 3: Problem Solving (24 points)

**Question 1. (9 points – 3 each)** Suppose that the following processes arrive the times indicated to run on the CPU for the indicated burst times.

Process	Arrival Time	Burst Time
P1	0.0	16
P2	1.5	2
P3	1.0	1

- a) What is the average turnaround time for these processes when using the FCFS (first-come first-serve) CPU scheduling algorithm?
- b) What is the average turnaround time for these processes when using the SJF (shortest job first) with **preemption** CPU scheduling algorithm?
- c) What is the average turnaround time for these processes when using the round robin CPU scheduling algorithm with a quantum of 3?

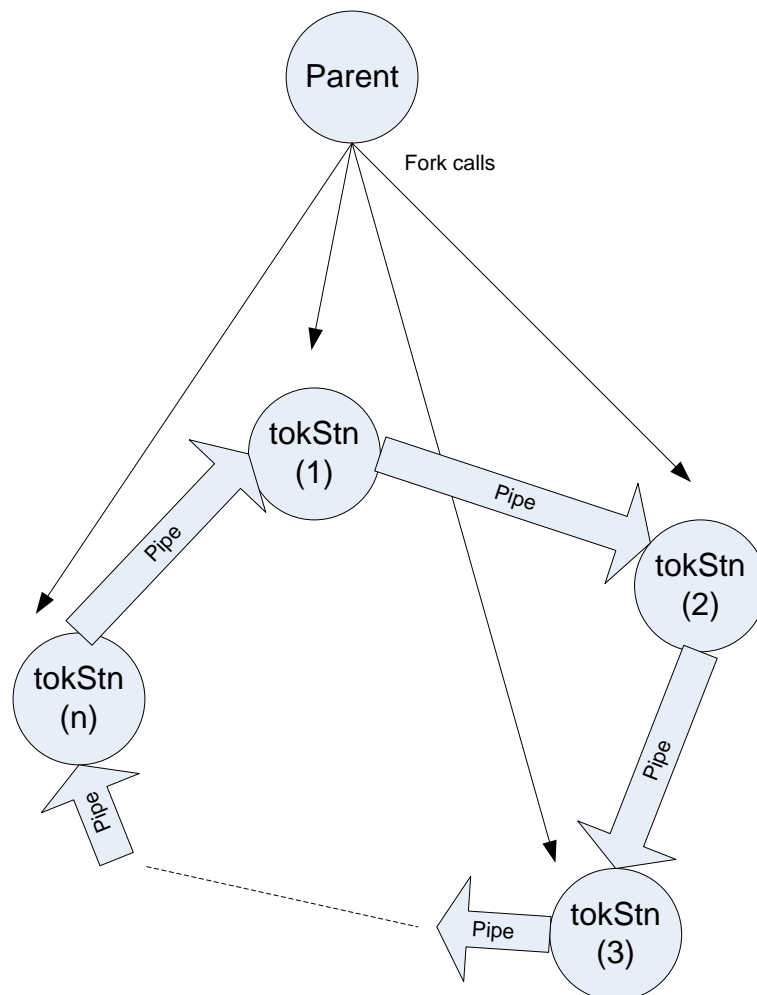
**Question 2 (15 points)** Create  $n$  child processes connected in a ring.

*Detailed specification:* The function `void createRing(int n)` is called in a parent process to create  $n$  child processes. A pipe connects the standard output of each child process to the standard input of another child process to form a ring as shown in the diagram below. Complete the function with the appropriate pipe, fork, dup2, and `execlp` calls to create such a ring of child processes. Child processes should only have `stdin`, `stdout`, and `stderr` file descriptors open when the `execlp()` is called. The child processes should run the `tokStn` program (with no arguments).

Hint:  $n-1$  children can be created in a loop, the last child must be treated as a special case.

Synopsis of system calls:

```
int fork()
int dup2(int oldfd, int newfd)
int execlp(char *file, char *arg,...,NULL)
int pipe(int filedes[2])
int close(int fd)
```



```
void createRing(int n) {
```

```
}
```



// Extra space if required

**Bonus - (2 points).** Including the initial parent process, how many processes are created in the program below?

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

int main()
{
    int i;

    for (i = 0; i < 4; i++)
        fork();

    return 0;
}
```