CISC 324 Winter 2017
Assignment 1
Due Friday, Jan 20, at the 11:30 lecture

[Information about labs is given at the end of this document. Labs 1 and 2 are due Monday, Jan 23, at the 1:30 lecture.]

**Handing in assignments and labs**
I do not accept late papers. If you cannot make it to lecture when an assignment or lab is due, then hand it in early:

- During business hours, bring your assignment/lab to Goodwin 557: that’s the School of Computing Office, located on the fifth floor of Goodwin Hall across from the elevators. Ask the receptionist, Karen Knight, to put your assignment into my mailbox.
- After business hours, slide the assignment/lab under the door of Goodwin 557. Clearly label it as “CISC324 for Prof. Blostein”, and Karen will put it into my mailbox the next day.

Your answers may be handwritten or typed. If you write by hand, please write neatly enough so that your answers are easy to read.

**Assignments and labs may be completed individually or with a partner**
If you work with a partner, write both names on the assignment and lab papers. Make sure that both of you participate fully. Working with a partner can help both of you learn: discuss the concepts, ask each other questions, explain to each other.

Carefully working through the assignments and labs is essential for learning the course material. My exam questions are aimed at testing whether students thoroughly understood the assignments and labs.

**Readings for Assignment 1**

CISC324 Course Reader. These readings are for questions (1) to (6) below. Most of this is review of concepts from previous courses.

- **Pages 5-8** Levels of Description of a Computer System
- **Pages 9-15** Overview of the Instruction Execution Cycle.
  For CISC324 exams you need to understand instruction execution in general, but you are not expected to know any details specific to the Pentium assembly language and machine language. In order to provide a clear review of instruction execution, the assignment questions below refer to Pentium details. You can solve these problems easily by referring to the course reader, and rest assured that you are not expected to memorize these details.
- **Pages 16-18** Processes versus Interrupt Handlers
  A thorough understanding of interrupts is essential for CISC324. If you need review of this concept:
  - Interruptions are summarized in step 4 on page 13.
  - Pages 96-97 show a context switch in detail. This provides details about how the interrupt handler, which is part of the operating system, saves the PC and SP values for the process that was executing on the CPU and then restores the PC and SP values for some other process that now gets a turn executing on the CPU.
  - Pages 16-18 discuss the difference between interrupt handlers and processes. These are very different concepts, but when you first learn about them it is possible to get them mixed up.

[Information about labs is given at the end of this document. Lab descriptions are in the course reader:]

- **Pages 68-74** Description of Lab 1: Introduction to Concurrency in Unix
- **Pages 75-76** Description of Lab 2: Introduction to Concurrency in Java

The textbook, *Operating System Concepts*, 9th edition (or 8th edition). These readings are for questions (7) to (9) below.
- **Chapter 1** Introduction
- **Chapter 2** Operating System Structures
  Chapters 1 and 2 contain a lot of information. Read/skim these chapters for a general introduction. The assignment questions draw your attention to the topics that are most important to this course.
Questions for Assignment 1

(1) Describe three reasons to use simulation in the design of a computer system.

(2) I have access to two computers, ComputerX and ComputerY, that run the same machine language and use the same operating system. I want to move a C program from ComputerX to ComputerY. Will it work for me to transfer the object file (that’s the output produced by the C compiler running on ComputerX) to ComputerY, or do I have to transfer the source code (the C code) and recompile on ComputerY? Briefly justify your answer.

Questions (3) to (6) review assembly language, registers, machine code, and instruction execution. You need to understand these concepts as background information for CIS324 material. Pentium assembly language is used to illustrate these concepts in the course reader. Read these examples carefully, but don’t try to memorize any details about the Pentium assembly language; I will not ask exam questions about this assembly language.

(3) Translate the following C code into Pentium assembly language. (Modify the example given on pages 9-10 of the course reader.)

```c
total = 12;
for (i = 2; i <= 50; i=i+1) {
    total = i + total + i;
}
```

(4) Show Pentium machine code for the assembly language statement `MOV BX, 23`. (Modify the example at the start of page 11.)

(5) (a) Briefly describe the roles played by PC (Program Counter) and IR (Instruction Register) during instruction execution.

(b) Which of the following operations takes place during execution of a JMP instruction? Briefly explain.
- update the value in IR using the value in PC
- update the value in PC using the value in IR
- update the value in SP using the value in PC
- update the value in PC using the value in SP

(6) (a) Why is it necessary for the hardware to save and restore register values during an interrupt?

(b) Which of the following registers must be saved? (Indicate all that apply)

AX BX CX DX PC IR MAR MDR Flags

(7) Briefly describe the meaning of the following terms.

(a) kilobyte, megabyte, gigabyte
(b) process
(c) An operating system responds to events that are signaled by an interrupt. (Section 1.2.1)
(d) processor, multiprocessor (Section 1.3)
(e) multiprogramming (Section 1.4) [Watch out: don't confuse the terms multiprogramming and multiprocessor.]
(f) Kernel of an operating system (Section 1.1.3 and 2.7.3)
(g) Virtual Machine (Section 16.1 in 9th edition; section 2.8 in 8th edition)

(8) Describe the steps involved in a DMA (Direct Memory Access) transfer. Discuss the following example. Process A is executing on the CPU, and requests the transfer of 100 bytes from disk to main memory. Describe how this data transfer is carried out. Include mention of how interrupts are used, and describe what happens to process A while the transfer is taking place.

[In case it helps you to visualize this scenario, here is a concrete example. Process A calls the unix function fread as follows `fread(ptr, 1, 100, myfile)`. This is how Process A requests the OS to transfer 100 bytes from file myfile to the block of main memory whose address is given by ptr.]
Choose one of the following answers and justify your choice. When several processes are executing,
- the operating system can let all processes use one shared stack.
- the operating system must maintain a separate stack for each process. The operating system does this by saving and restoring
  the value of the SP register at each context switch.

Hint: the second answer is correct. To see why the first answer is wrong, consider what happens when several processes are
executing code with procedure calls. For example, suppose process A calls a proc1; then there is a context switch to process B,
and process B calls proc2. After a context switch back to process A, process A tries to return from proc1 back to its main
program. What happens?

**Lab Work**

This course does not have scheduled lab times, so you can carry out the lab work whenever you wish. Lab instructions are given in
the CISC 324 Course Reader. **The files you need for the labs** are available here:
- on caslab machines at /cas/course/cisc324
- on the CISC324 web page  www.cs.queensu.ca/home/cisc324
- on the Caslab ftp site

The lab instructions in the CISC324 Course Reader tell you what material to hand in, to show that you have completed the lab. The
first two labs are straightforward introductions to concurrency. You should be able to complete each of them in less than an hour.

**Lab 1** provides an introduction to concurrency in unix, using the & and | commands to launch processes. This lab must be performed
on a computer that runs unix. You can use the caslab machines as follows. Log onto any of the caslab PCs in Goodwin Hall 248 or
Walter Light 310, and put them into unix mode by selecting XWin32 from the start menu. This opens up a window that runs unix.
When you are finished with the lab, please type **ps** to get a list of your processes, and make sure that you got rid of the a.out process
that you created in step 6 of the lab. Type **kill <process id number>** to kill a process. If you don't kill the a.out process, it will keep
running even after you log out, and will waste a lot of CPU time.

**Lab 2** provides an introduction to concurrency in Java, using threads. Lab 2, and all the rest of the labs in the course, can be
performed using any Java platform. The lab instructions tell you how to capture your Java output if you are running Java under unix;
if you are using another Java platform, you must find a way to capture your output. If you have problems with any of the labs, please
contact a TA.