McGill University
School of Computer Science
COMP-273A Introduction to Computer Systems

Final Exam

December 14, 2006 at 2PM – WONG
Examiner: Joseph Vybihal
Assoc. Examiner: Mike Langer

Student Name: ______________________________

Student ID: ________________________________

Instructions

- No notebooks, calculators or textbooks permitted in this exam.
- Language translation dictionaries are permitted.
- You are permitted to write your answers in either English or French.
- Attempt all questions.
- Parts marks are given for all questions - show your work.
- All answers must be written on the exam booklet provided.

Grading

Section	Grade	Your Mark
Question 1:	______ / 20	
Question 2:	______ / 20	
Question 3:	______ / 20	
Question 4:	______ / 20	
Question 5:	______ / 20	
	---------	--------------
Total	100 %	

Question 1: Recursive Assembler Programming

Write an assembler program that computes the factorial recursively.

Your program will not read input from the user. Define at least two data values. One data value will be called N and the other ANSWER. N will be the input value for your recursive program. ANSWER will store the final result. Your program will not print out the final result.

Your program must use a subroutine called FACT that will use a stack to recursively compute the factorial of N. Your program will have a main part that passes N to the subroutine and stores the answer into ANSWER when FACT terminates.

Question 2: Dynamic Assembler programming

Write an assembler program that creates a linked list and then deletes from it.

Your program will read integer numbers from the user in a loop until the user inputs a zero. Prompt the user for these numbers. For each number inputted – the program will create a node and insert the value at the end of a linked list. Your linked list will not reside in the computer's dynamic memory but instead you will use your program's DATA section to reserve space for the links. You should reserve space for at least 10 links. After the user has input zero the program will go into delete mode. Prompt the user for integer numbers in a loop that will be deleted from the linked list. This loop terminates when the user inputs zero. The program just terminates after this zero is input.

Question 3: Circuit Design Problem

Construct a circuit that solves the following problem:

A candy machine has three types of candies: $0.25, $0.50 and $1.00. The machine is designed to accept only quarters. The machine has three buttons: A for the candy in the $0.25 slot, B for the $0.50 slot and C for the $1.00 sot. If the user has input enough money and has selected the correct slot then the machine drops the candy. Assume a line can be attached to a motor A, B and C that triggers the dropping of the candy. Assume further that there is a device that senses the drop of a quarter and sends a signal into your system when it senses one. Any other change is automatically / mechanically rejected by the machine and does not enter the system. The machine has a change lever that when pulled will return the money that was input into the machine but only if the user has not pressed a button that successfully dropped some candy. The machine has an input power line and an optional clock (if you need this).

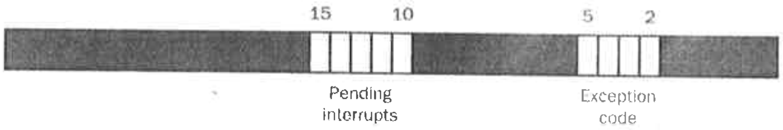
Question 4: Interrupt Programming

You do not need to write a complete application for this question. Instead assume that a financial program exists performing complex mathematical calculations. Assume further that various errors are possible that require special interrupt handling when they occur.

Given the following:

Number	Name	Description
0	INT	external interrupt
4	ADDRL	address error exception (load or instruction fetch)
5	ADDRS	address error exception (store)
6	IBUS	bus error on instruction fetch
7	DBUS	bus error on data load or store
8	SYSCALL	syscall exception
9	BKPT	breakpoint exception
10	RI	reserved instruction exception
12	OVF	arithmetic overflow exception

And



Write an interrupt handler that could manage the following errors: OVF and SYSCALL. In the case of OVF the program should terminate. In the case of SYSCALL a message should be printed to the user indicating that an unknown error has occurred, but in this case the handler will return execution to the program from the point where the error occurred.

Question 5: Interface Programming

Assume you've been asked to create a program that controls a mobile robot using MIPS. The robot's MIPS programming environment is organised as follows:

- It has a RAM memory space where your program will reside and execute.
- The MIPS CPU is a simple architecture and is not pipelined nor does it have a cache. Instead it is designed to operate directly with RAM. This makes it slower but the CPU's function is just to operate this robot and it is powerful enough for that.
- The robot's RAM space is memory mapped to special registers that control important capabilities of the robot. This memory mapped area ranges from address 0x000 to 0x999.
    - ♦ At address 0x100 is the robot's 8-bit status register. The status register records the following information about the robot:
        - ▪ Bit 0 to 4 is an integer number representing the battery power. Zero means no power. The robot should stop running when the power level drops to 1.
        - ▪ Bit 5 indicates that the turn right command was issued previously (it clears to zero automatically if the previous turn was not right).
        - ▪ Bit 6 indicates that the turn left command was issued previously (it clears to zero automatically if the previous turn was not left).
        - ▪ Bit 7 is not used.
    - ♦ At address 0x200 is the robot's 8-bit command register. The following commands can be issued:
        - ▪ Number 0 is stop motion
        - ▪ Number 1 is drive forward (it does this continuously until the command value changes)
        - ▪ Number 2 means turn right 45 degrees.
        - ▪ Number 3 means turn left 45 degrees.

- At address 0x300 is the robot's 32-bit sensor register. It sensor is a single infrared diode in the front of the machine and serves as its eye. If nothing is in front of the robot the value is in the register is zero. As something comes closer to the diode the integer value increases until a maximum of 100 which indicates that you have crashed into the object. There are no negative values.

Write a simple robot control program that makes this robot run about a space while avoiding obstacles. The robot will stop moving when its power level drops to 1. Design the program so that the robot turns right when it encounters its first obstacle, then left after the next and alternates in this fashion.