CPSC 313, 06w Term 2— Midterm 1

Date: February 9, 2007; Instructor: Norm Hutchinson

This is a closed book exam; no notes; no calculators. Answer in the space provided; use the backs of pages if needed.

There are **6** questions on **4** pages, totaling **42** marks.

You have **50 minutes** to complete the exam.

On the last two pages you will find summaries of the x86 instructions and address modes. You may find it profitable to (carefully) remove these pages from the exam.

You must write your name and student number on the exam and sign the exam in pen. You should write this exam in pen - I will not consider requests to regrade solutions that are written in pencil.

NAME: _____ **SCORE:** _____ / 42

STUDENT NUMBER: _____

SIGNATURE: _____

_____ **1.** **(8 marks)** Short answers.

1a. **(2 marks)** Is the address of a global variable in a C program determined *statically* (before the program runs) or *dynamically* (while the program is executing)? Briefly explain.

1b. **(2 marks)** Is the content of the jump table used in the execution of a switch statement determined *statically* or *dynamically*? Briefly explain.

1c. **(2 marks)** Does the IA32 instruction-set architecture require that `%eax` be used to hold the value returned from a C function call? Briefly explain.

1d. **(2 marks)** If `%ecx` holds the address of an integer array `a`, and `%edx` holds the integer `i`, give a single assembly-language statement that computes `&a[i]` and places it in the register `%eax`.

2. **(8 marks)** Consider the following C source file.

```
int g;
void f (int t) {
    int a; int *b;
    /* consider each statement as if it were here */
}
```

Give an assembly-code implementation of each of the following statements of function `f()`. Consider each statement in isolation (i.e., as if it were the only statement of f). Do not assume that variables start out in registers. Be sure to write results to the appropriate location in memory. Assume that the local variables are in memory on the stack (not in a register). A fully correct answer will use as few instructions as possible. **Comment your code.**

2a. **(2 marks)** `b = &g;`

2b. **(2 marks)** `a = *b;`

2c. **(2 marks)** `*b = 3;`

2d. **(2 marks)**

```
do {
  a = a - t;
} while (a > 0);
```

3. **(8 marks)** Assume that the following registers hold the indicated C language variables with the indicated types:

Register	Variable name	type
`%ebx`	a	int
`%ecx`	b	int
`%edx`	c	int *

For each of the indicated snippets of assembly code, write C-language statements that have the same effect. If your solution is significantly longer than mine, you will lose marks.

3a. **(2 marks)**

```
    leal    (%ebx, %ecx, 8), %eax
    ret
```

3b. **(2 marks)**

```
    addl    (%edx), %ecx
```

3c. **(2 marks)**

```
    movl    (%edx), %eax
    addl    (%edx), %eax
    movl    %eax, (%edx)
```

3d. **(2 marks)**

```
        cmpl    %ebx, %ecx
        jg  .L1
        movl    %ebx, %eax
        jmp .L0
.L1:    movl    %ecx, %eax
.L0:    ret
```

_____ **4.** **(6 marks)** Describe the input to and the work performed by each of the following three components of the process of converting a program in a high level language into an executable program.

4a. **(2 marks)** Compiler

4b. **(2 marks)** Assembler

4c. **(2 marks)** Linker

5. **(6 marks)**

5a. **(3 marks)** Describe the three major tasks accomplished by a function's prologue.

-
-
-

5b. **(3 marks)** Describe the three major tasks that must be accomplished by the code that makes a call to another function.

-
-
-

6. **(6 marks)** A C program contains a function `F` with one integer parameter which calls a function `G` with one integer parameter.

6a. **(4 marks)** Draw a picture of the runtime stack immediately before the `call` instruction in function `F` executes. Your stack should contain all of the activation record for the function F including its one integer parameter.

In your picture, clearly indicate the location of the saved frame pointers, parameters, and return addresses. You should indicate the general location of local variables, but need not show them in detail. Clearly indicate exactly where in the stack the registers `%esp` and `%ebp` point. Orient your stack so that lower addresses are at the top of the page.

6b. **(2 marks)** Give a single machine instruction that will load the address of the frame pointer for F's caller into the register `%eax`.

You may (carefully, so as to not destroy the staple) remove these last 2 pages from the exam and use them as a reference.

instruction	effect	description
leal s,d	d ← &s	load effective address
inc_ d	d ← d + 1	increment
dec_ d	d ← d - 1	decrement
neg_ d	d ← -d	negate
not_ d	d ← ~d	complement (bitwise)
add_ s,d	d ← d + s	add
sub_ s,d	d ← d - s	subtract
imul_ s,d	d ← d * s	multiply (32-bit)
xor_ s,d	d ← d ^ s	exclusive-or (bitwise)
or_ s,d	d ← d | s	or (bitwise)
and_ s,d	d ← d & s	and (bitwise)
sal_ k,d	d ← d << k	left shift
shl_ k,d	d ← d << k	left shift (same as sal_)
sar_ k,d	d ← d >> k	arithmetic right shift
shr_ k,d	d ← d >> k	logical right shift

type	gas form	operand value	addressing mode
immediate	$imm	imm	immediate
register	%r	R[r]	register
memory	imm	M[imm]	absolute
	(%r)	M[R[r]]	indirect
	imm(%r)	M[imm+R[r]]	base+displacement
	(%rb,%ri)	M[R[rb]+R[ri]]	indexed
	imm(%rb,%ri)	M[imm+R[rb]+R[ri]]	indexed
	(,%r,s)	M[R[r]*s]	scaled (by 1,2,4,8) indexed
	imm(,%r,s)	M[imm+R[r]*s]	scaled (by 1,2,4,8) indexed
	(%rb,%ri,s)	M[R[rb]+R[ri]*s]	scaled (by 1,2,4,8) indexed
	imm(%rb,%ri,s)	M[imm+R[rb]+R[ri]*s]	scaled (by 1,2,4,8) indexed

instruction		synonym	jump condition	description
jmp	label		1	direct jump
jmp	*operand		1	indirect jump
je	d	jz	zf	equal / zero
jne	d	jnz	~zf	not equal / not zero
js	d		sf	negative
jns	d		~sf	nonnegative
jg	d	jnle	~(sf ^ of) & ~zf	greater than (signed >)
jge	d	jnl	~(sf ^ of)	greater or equal (signed >=)
jl	d	jnge	sf ^ of	less than (signed <)
jle	d	jng	(sf ^ of) | zf	less or equal (signed <=)
ja	d	jnbe	~cf & ~zf	above (unsigned >)
jae	d	jnb	~cf	above or equal (unsigned >=)
jb	d	jnae	cf	below (unsigned <)
jbe	d	jna	cf | zf	below or equal (unsigned <=)