**COMP 2130, Final Test, Fall 2012**

Student Name: Student Number:

**Note 1: All the programming related questions assume C or IA32.**

1. (2 marks) List the four major hardware components in a computer system.
2. (2 marks) List the four phases in the compilation system.
3. (2 marks) List the two steps in a CPU instruction cycle.
4. (2 marks) Explain how CPU, PC (Program Counter), and MMU (Memory Management Unit) are used together?
5. (2 marks) When is the virtual address space used? When is the physical address space used? What address space is used by the instructions that are fetched and executed by CPU?
6. (2 marks) For what are stack and heap in the virtual memory space used?
7. (2 marks) Explain how a process and a computer program are different.
8. (2 marks) Write a definition of a function that is given a list of data items, finds the average, and returns the average.
9. (2 marks) Write a statement to print a student’s name, number and height that are stored in the next 3 variables.

float height = ...;

int number = ...;

char name[] = {‘G’, ‘o’, ‘l’, ‘i’, ‘a’, ‘t’, ‘h’};

1. (6 marks) Write a function to check even-parity of a given int parameter. That is that the function returns 1 when the number of bit 1's is even, otherwise 0.
2. (2 marks) Convert –115 to an 8-bit signed binary number, assuming the 2’s complement representation.
3. (2 marks) Convert a binary number 10110101 to a hexadecimal number.
4. (6 marks)
5. Convert two hexadecimal numbers, 0x6B and 0x3D, to binary numbers *Q* and *P*.
6. Compute the binary subtraction *R* = *P* – *Q*, where *P* and *Q* are the binary numbers in (a), assuming the 2’s complement representation.
7. What is the subtraction result *R* in (b) in decimal, assuming the 2’s complement representation?
8. (2 marks) Compute (1100 ^ 0110 & 0111 | 1000) >> 2, assuming the 4-bit representation. (Note that the precedence of bitwise operators is >>, <<; &; ^; |.)
9. (3 marks) What message will the next code segment print? (Note that signed integers use 2’s complement representation.)

char x, y, w, t;

unsigned char z;

int s;

x = 128;

y = 129;

z = x + y;

w = x + y;

s = x + y;

t = s;

printf(“%d, %d, %d, %d, %d, %d\n”, x, y, z, w, s, t);

1. (2 marks) Compute 1110 × 0110, assuming 4-bit 2’s complement representation. What is the result in decimal?
2. (4 marks) What will the following code print?

int x = 1; int y = 2; int z[10] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11};

int \*ip;

ip = &x;

printf(“%d, %d, %d, %d\n”, x, y, \*z, \*ip);

y = \*ip;

printf(“%d, %d, %d, %d\n”, x, y, z[0], ip[0]);

\*ip = 0;

printf(“%d, %d, %d, %d\n”, x, y, \*(z+1), \*ip);

ip = &z[1];

printf(“%d, %d, %d, %d\n”, x, y, z[1], ip[0]);

1. (2 marks) List two major jobs that linkers do.
2. For the following user defined data type,

struct Student {

 int number;

 char name[60];

 struct Student \*prev, \*next;

}

1. (4 marks) Write a function that

allocates the memory space for **struct Student**,

reads an integer and a string from the user and saves them into the memory space,

initializes **prev** and **next** with **NULL**, and

returns the reference of the memory space.

1. (2 marks) Write a function that has a parameter of **struct Student \***, and returns the name stored in the parameter.
2. (2 marks) A function fun() has the following code body:

\*p -= d;

return;

The IA32 code implementing the above code is as follows:

movl 12(%ebp), %eax

movl 8(%ebp), %edx; // the first parameter

subl %edx, (%eax);

Write the definition of function fun(), showing the types and ordering of the arguments p and d.

1. (2 marks) Explain how static libraries and shared libraries are different.
2. (6 marks) Assume the following values are stored at the indicated memory address and registers.

Address Value Register Value

0x200 0xAF %ebx 0x204

0x204 0xC8 %ecx 0x2

0x208 0x200 %edx 0x8

0x20C 0x204 %ebp 0x200

Fill the following table showing the values for the indicated operands:

 Operand Value

 %edx

 (%ebx)

 12(%ebp)

Fill the following table showing the values for the instructions:

 Instruction Value in

 movl %ebp, %eax %eax

 movl 8(%ebp), %eax %eax

 leal (%ebx, %ecx, 4), %eax %eax

1. (2 marks) What values will be in %eax, %ecx, and 12(%ebp) at the end of the following code:

*Initially 8(%ebp) has 9, and 12(%ebp) has 2.*

movl 8(%ebp), %ecx

movl %edx, %eax

movl 12(%ebp), %eax

shrl $2, %ecx

decl %eax

addl %ecx, %eax

subl %eax, 12(%ebp)

1. (2 marks) What values will be in %eax at the end of the following code:

*Initially 8(%ebp) has 5, and 12(%ebp) has 9.*

movl 8(%ebp), %ecx

movl 12(%ebp), %ebx

cmpl %ebx, %ecx // cmpl D, S -> S – D is compared

jle .L2

movl %ecx, %eax

jmp .L1

 .L2

movl %ebx, %eax

 .L1

1. (3 marks) Write a goto version of the following code: (You may convert for version to while version first.)

int loop\_for(int a, int b)

{

 int i;

 int sum = 0;

 for (i = 0; i < 5; i++)

 sum += a + i \* b;

 return sum;

}

1. (3 marks) Translate the following C code into IA32 assembly code:

// *Use %eax for r, %ebx for x, and %ecx for y.*

r = y – x;

if (x > y)

 r = x – y;

1. (2 marks) The following code has a logical error. Fix the error.

float \*search(float \*p, int val) { // search val in the data items pointed by p

 while (\*p && \*p != val)

 p = p + sizeof(float);

 return p;

}

1. (2 marks) What value will z have at the end of the following code:

x = 500; y = 1000;

\*p = y;

\*q = x;

z = \*p;

1. (2 marks) Explain how ft1() and ft2() in the following code can return different values:

int f();

int ft1() { return f() + f(); }

int ft2() { return 2 \* f(); }

1. (2 marks) Optimize the following code.

for (i = 0; i < m; i++)

 for (j = 0; j < n; j++)

 a[j] = b[j + i\*n];

1. (2 marks) Optimize the following code in two different ways.

 for (i=0; i < vec\_length(v); i++)

 \*sum = \*sum + val;

1. (2 marks) Compute the average time (in ms) to access a sector on the following disk:

Rotation rate: 7200 RPM; average seek time: 5 ms; average number of sectors / track: 1000

1. (2 marks) Improve the following code so that the code can have the locality property:

int sum(int a[K][L][M]) {

 int i, j, k, sum = 0;

 for (i = 0; i < L; i++)

 for (j = 0; j < K; j++)

 for (k = 0; k < M; k++)

 sum += a[j][i][k];

 return sum;

}

1. (2 marks) In the following code, what data references have spatial locality? What data references have temporal locality? (Note that there are data references and instruction references.)

sum = 0;

for (i = 0; i < n; i++)

 sum += a[i];

return sum;