CSCI-UA.0201

Computer Systems Organization

Final Exam Fall 2015 (time: 90 minutes)

Last name: First name: NetIT:

1. (5 points) Circle the correct answer among the choices given. If you circle more than one answer, you will lose the grade of the corresponding question.

(A) Which one of the following is NOT one of the jobs of the linker?

1. concatenate all modules together 2. adjust relocatable symbols (e.g.calls, jumps…)
3. add static libraries to the executable 4. add dynamic libraries to the executable

(B) The call to *malloc* in C:

1. will always lead to system call 2. will never lead to system call
3. will sometimes lead to system call 4. will cause an interrupt

(C) Assume we have a one-level page table. The size of a page affects the number of entries in the page table.

1. This statement is true. 2. This statement is false
3. This statement is true or false depending on the platform (i.e. hardware + OS)

(D) The TLB is accessed using:

1. virtual address 2. physical address 3. depends on the platform

(E) Assume we have the following statement in *C:* ****A = B;*** . Also assume A is in ebx and B is in ecx. Which of the following is a correct x86 translation to the C statement?

1. movl %ebx, %ecx 2. movl %ecx, %ebx 3. movl %ebx, (%ecx)
4. movl (%ecx), %ebx 5. movl (%ebx), %ecx 6. movl %ecx, (%ebx)

2. Suppose we have a cache memory of 16 blocks. The address length is 14 bits. If the address is divided as follows:

- **4 bits for the offset**
- **2 bits for the set**
- **8 bits for the tag**

(a) [2 points] What is the associativity of that cache (show all steps to get full credit)?

(b) [1 point] What is the block size (show all steps to get full credit)?

3. (5 points) Consider the following assembly representation of a function `foo` containing a `for` loop:

```
foo:
     movl %edi,%ebx
     leal 2(%ebx),%edx
     xorl %ecx,%ecx
     cmpl %ebx,%ecx
     jge .L4
.L6:
     leal 5(%ecx,%edx),%edx
     leal 3(%ecx),%eax
     imull %eax,%edx        # this means edx = edx * eax
     incl %ecx              # this means ecx++
     cmpl %ebx,%ecx
     jl .L6
.L4:
     movl %edx,%eax
     ret
```

Fill in the blanks in the following C code to provide the functionality of the loop:

```
int foo(int a){

     int i;
     int result = _____________;

     for( ________; ________; i++ )

     { ___________________;


       ___________________;
    }

     return result;

}
```

4. For the following piece of code:

```
int counter = 1;
int main() {
        if (fork() == 0) {
                counter--;
                exit(0);
        }
        else {
                wait(NULL);
                counter++;
                printf("counter = %d\n", counter);
        }
        exit(0);
}
```

a. [1 point] What will be printed on the screen?

b. [1 point] If the system that is executing the above code is using one-level page table as well as one TLB. How many page tables are alive in the system before non of the processes execute exit(0)?

c. [1 point] When the CPU is executing the statement *counter++;* how many processes are alive in the system?

d. [1 point] If the system that is executing the above code is using one-level page table as well as one TLB. How many TLBs exit in the system before non of the processes execute exit(0)?

5. [3 points] We said in class that sometimes a stack frame may not be needed when a function is called. State the conditions necessary for the compiler NOT to make a stack-frame when a function is called.