Fall term 2011
KAIST EE209 Programming Structures for EE

Final exam

Thursday Dec 15, 2011

Student's name: _____

Student ID: _____

The exam is closed book and notes. Read the questions carefully and focus your answers on what has been asked. You are allowed to ask the instructor/TAs for help only in understanding the questions, in case you find them not completely clear. Be concise and precise in your answers and state clearly any assumption you may have made. All your answers must be included in the attached sheets. You have 120 minutes to complete your exam. Be wise in managing your time.

Scores

Question 1	_____/10___
Question 2	_____/20___
Question 3	_____/15___
Question 4	_____/10___
Question 5	_____/10___
Question 6	_____/15___
Total	_____/80___

1. Briefly explain following terms (10 pt)

   (a) Page fault (2pt)

   (b) Spatial locality (2pt)

   (c) Stack frame (2pt)

   (d) System call (2pt)

   (e) Singal (2 pt)

2. Programming with fork() (20 pt)

a) What does this program print out? Write every possible output. (5pt)

```
static void f(void) {
 putchar('A');
 fflush(NULL);
 if (fork() == 0) {
      putchar('B');
      exit(0);
 }
 putchar('C');
 wait(NULL);
 putchar('D');
}

int main(void) {
 putchar('E');
 f();
 putchar('F');
 return 0;
}
```

b) What does this program print out? (5pt)

```
int main(void)
{
 int i;

 putchar('A');
 fflush(NULL);
 for (i = 0; i < 3; i++)
       fork();
 putchar('B');
 return 0;
}
```

c) What's the maximum number of characters that can be printed out to stdout in the following program? Please note that we got rid of fflush(NULL); from the program above (b). (5pt)

```
int main(void)
{
 int i;

 putchar('A');
 for (i = 0; i < 3; i++)
        fork();
 putchar('B');
 return 0;
}
```

d) What's the output of the second printf() (printf("parent ..."))? (3pt). How many times is the second printf(() called? (2pt)

```
int main(void)
{
 int k = 1;

 if (fork() == 0) {
        k++;
        printf("child k=%d\n", k);
        exit(0);
 }
 k += 2;
 wait(NULL);
 printf("parent k=%d\n", k);
 return 0;
}
```

3. The following assembly code was generated by gcc209 by compiling a simple C function (named f). It takes two integer parameters and returns an unsigned integer value. You may assume the passed-in parameters are in the range of 1 to 10. (15pt)

```
.file   "f.c"
.text
.globl f
.type   f, @function

f:
        pushl   %ebp
        movl    %esp, %ebp
        subl    $24, %esp
        cmpl    $1, 12(%ebp)
        jne     .L2
        movl    8(%ebp), %eax
        jmp     .L3
.L2:
        movl    12(%ebp), %eax
        movl    %eax, %edx
        shrl    %edx
        movl    8(%ebp), %eax
        imull   8(%ebp), %eax
        subl    $8, %esp
        pushl   %edx
        pushl   %eax
        call    f
        addl    $16, %esp
        movl    %eax, -12(%ebp)
        movl    12(%ebp), %eax
        andl    $1, %eax
        cmpl    $0, %eax
        jne     .L4
        movl    -12(%ebp), %eax
        jmp     .L3
.L4:
        movl    8(%ebp), %eax
        imull   -12(%ebp), %eax
.L3:
        leave
        ret
```

Some hints to understand this code:

- "leave" releases the stack frame, copying EBP to ESP
- imull multiplies two operands and stores the result to the second operand.

(a) Where are the two passed-in parameters in terms of %ebp? (4pt)

(b) What's the value of f(3, 3)? (4pt)

(c) Write the equivalent code in C. (5pt)

(d) Describe what the function does in one sentence. (2pt)

4. Memory management (10 pt)
   a) Given a virtual address, 0x34233230 on a lab machine, what is the virtual page number? What is the page offset? (4pt)

   b) Why could be the “best fit” memory allocation strategy a good method? (2pt)

   c) Why could be the “best fit” memory allocation strategy a bad method? (2pt)

   d) What does execvp() do? (1pt) Does it create a new process? (1pt)

5. Exceptions and Process control (10pt)

(a) Describe four types of exceptions, and give at least one example to each type. (4pt)

(b) What is context switch? List at least three different causes for context switch. (4pt)

c) I'd like to kill a process whose pid is 1234. What's the Linux command to kill the process? (Assume you have the privilege to kill the process) (2pt)

6. Big integer operations (15pt)
   We are writing a library that can add and multiply two unsigned large integer values that cannot be represented by the C's built-in integer type (unsigned int, unsigned long, unsigned long long). Assume that the largest integer in the library can be represented by 32 * sizeof(unsigned int) bytes. u_int is typedef'ed to be unsigned int.

   For example, 0x11111111222222223333333344444444 can be represented by an integer array of size 4. That is

   u_int a[16] = {0};

   a[0] = 0x44444444
   a[1] = 0x33333333
   a[2] = 0x22222222
   a[3] = 0x11111111

   represents 0x11111111222222223333333344444444. a[3] represents the most significant four bytes whereas a[0] represents the least significant four bytes.

   a) add_large() takes two unsigned big integers (a[16], b[16]) and write the sum to c[17]. Please fill out the function. (5pt)

      ```
      void add_large(u_int a[16], u_int b[16], u_int c[17])
      {



      }
      ```

b) multiply_large() takes two unsigned big integers (a[16], b[16]) and writes the result of the multiplication of the values into c[32]. Please fill out the function. (10pt)

```
void multiply_large(u_int a[16], u_int b[16], u_int c[32])
{



}
```