

CSci 4061: Introduction to Operating Systems (Spring 2013)
First Midterm Exam (March 7, 2013) (100 points)
Open Book and Lecture Notes
(Bring Your U Photo Id to the Exam)

This exam paper contains 8 questions.
Total 100 points.

Suggested guideline: A 20 point question should take about 15 minutes to answer.

Student Name:
Student ID:

Q #	1	2	3	4	5	6	7	8
Max Points	16	12	12	20	10	10	10	10
Points Scored								

Question 1 (16 points total – 4 points for each part):

*For all parts of this question assume that the file and directory names do not contain blankspaces or special characters. All directories contain only regular files and directories – **no links**.*

(a) Suppose that a directory contains one file, as shown below by the output produced by **ls -l** command:

```
-rw----- 1 dean cs4061 28 2013-02-22 12:00 somefile
```

Show the output produced by the following sequence of commands (where % is the shell prompt character).

```
% ln somefile datafile1
% ln -s somefile datafile2
% ls -l
% rm somefile
% ls -l
```

(b) Write either a shell command or a script program (using bash or tcsh) which will take a directory pathname as the argument. It will count the number of all directories contained in the directory subtree specified by the given argument pathname.

(c) Write either a shell command or a script program (using bash or tcsh) which will take a directory pathname as the argument. It will look for all files named “core” in the _directory subtree, and print the size of each file named “core”.

(d) Write a either a shell command or a script program using bash or tcsh which will set to true both the ‘s’ bit and the “x” bit permissions **for group** on all subdirectories nested in a given directory. Only the directory permissions should be changed and no other files must not be affected.

Question 2: (12 points)

(a) (3 points) List at least three hardware level mechanisms that cause the execution mode of a process to change from the USER mode to PRIVILEGED mode.

(b) (9 points) Select the correct answer by circling YES or NO.

1. Are system call functions executed in the PRIVILEGED mode using the stack in the address-space of the process? YES NO
2. Using the system call *wait*, can a user process wait for the completion of any process in the system? YES NO
3. Can a user process directly read from or write to an I/O device? YES NO
4. Is a link entry pointing to a file is deleted when the target file has been deleted? YES NO
5. Can a process executing in USER mode read data in the kernel memory space? YES NO
6. Can process executing in USER mode write data in the kernel memory space? YES NO
7. When you type the command “cd” to change the current working directory, shell creates a child process to execute the “cd” command? YES NO
8. When you type the command *ls* to list file or directory, shell creates a child process to execute the *ls* command? YES NO
9. When you define a new global environment variable using *export* in bash or *set* in tcsh, will that variable be still defined if you log out and log in again later? YES NO

Question 3: (12 points) For each of the following `egrep` regular expressions, indicate which of the lines **contain a string** that matches the given expression.

Write your choice of answers as a, b, c, d, or e. All correct answers should be indicated.

(1) `^[^:]*::$`

- (a) password::
- (b) ::user::
- (c) user::password
- (d) :user-password::
- (e) user::password::

(2) `(cats)+Z+`

- (a) catsZZZ
- (b) cats+Z+
- (c) ZZZZZ
- (d) catsZ
- (e) cats

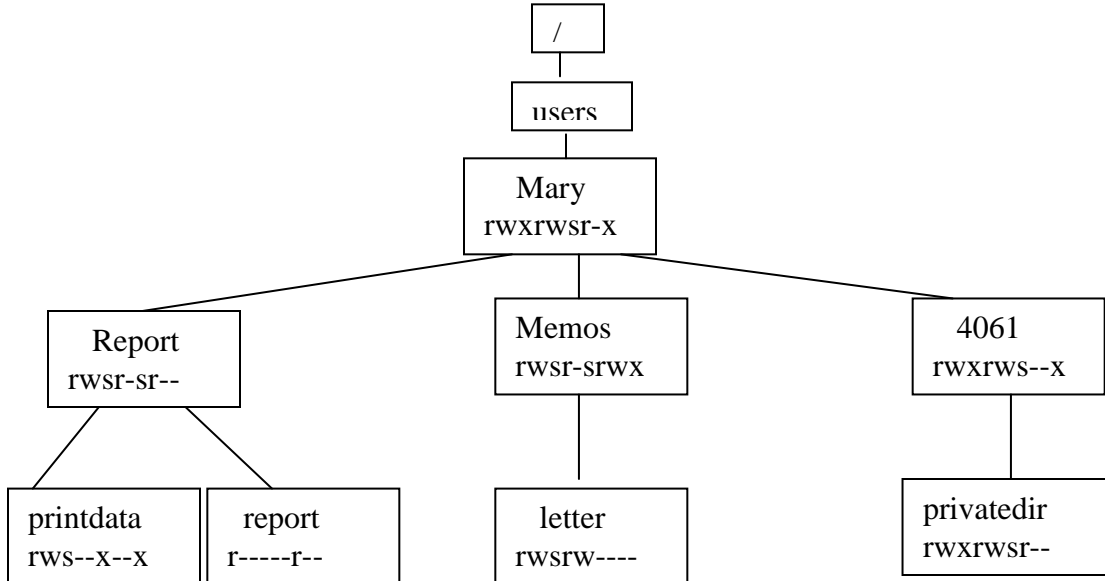
(3) `\<de.*ed`

- (a) decided option
- (b) predetermined
- (c) design editor
- (d) dean medical college
- (e) under-rated

(4) `a[^bd]+d?`

- (a) abd
- (b) aid
- (c) and
- (d) all
- (e) all added

Question 4: (20 points): Consider the following directory tree of files. The file permissions are shown along with each file name. Answer the following questions (2 points each) by circling **YES** or **NO**!



- (a) Can a user NOT in Mary's group read the file /users/Mary/Report/report? **YES** **NO**
- (b) Can a user in Mary's group read the file /users/Mary/Report/report? **YES** **NO**
- (c) Can a user in Mary's group write the file /users/Mary/Memos/letter? **YES** **NO**
- (d) Can a user in Mary's group delete the file /users/Mary/Memos/letter? **YES** **NO**
- (e) Can a user NOT in Mary's group delete the file /users/Mary/Memos/letter? **YES** **NO**
- (f) Can a user in Mary's group create a file in directory /users/Mary/Memos ? **YES** **NO**
- (g) Can a user NOT in Mary's group list the directory /users/Mary/4061? **YES** **NO**
- (h) Can a user NOT in Mary's group list the directory /users/Mary/cs4061/privatedir? **YES** **NO**

Suppose that the program /users/Mary/Report/printdata reads and prints the contents of the file /users/Mary/Report/report.

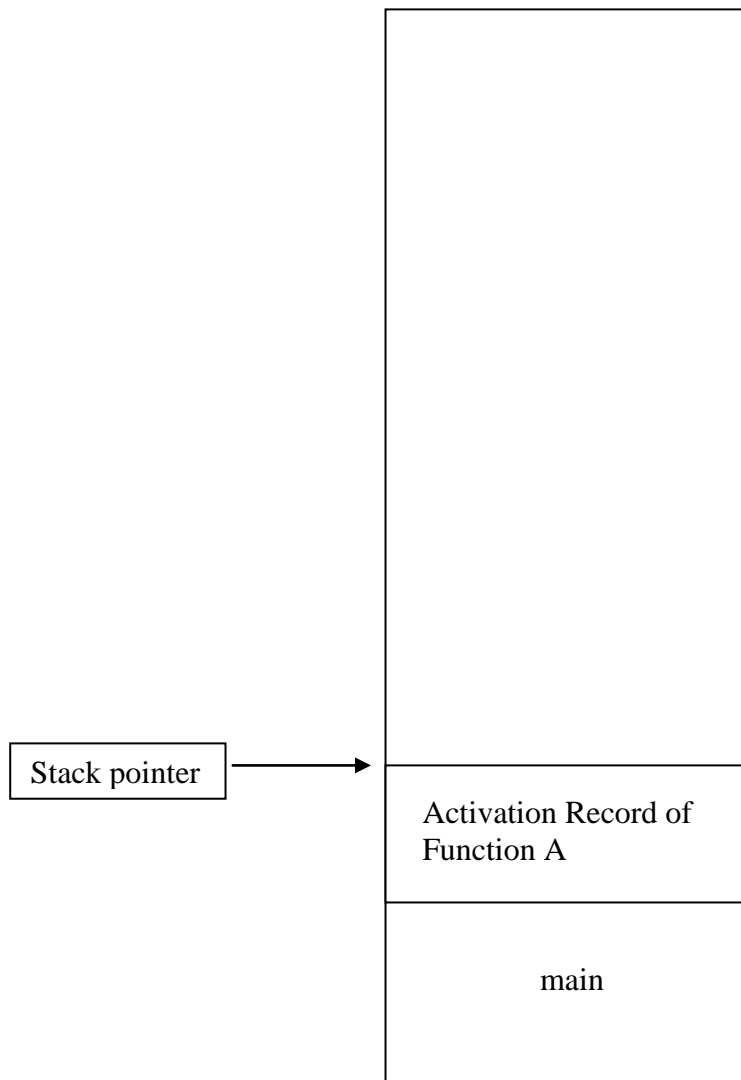
- (g) Can a user in Mary's group execute the program /users/Mary/Report/printdata to print the report file? **YES** **NO**
- (h) Can a user NOT in Mary's group execute the program /users/Mary/Report/printdata to print the report file? **YES** **NO**

Question 5: (10 points): Show the tree of processes (according to the parent-child relationship), that will get created when the following code is executed. Label the processes created in iteration i by label i . Label the process executing the code below by label 0.

```
void main ( void ) {  
    int  n = 4;  
    int  i, result;  
    for ( i=1;  i < n;  i++ ) {  
        result = fork();  
        if ( (result == 0) && (i == 2) ) {  
            break;  
        }  
    }  
}
```

Question 6: (8 points): Suppose that a program is currently executing function A, and its activation record is on the stack as show below. Function A now calls the following function B. Show the activation record that will be created on the stack when call to B is made. Assume that the return address requires 4 bytes. Assume that the stack grows from high to low address. Show the total length of the record, and space corresponding to different variables. Show stack pointer after the record is created.

```
Int function B (int x, in y) {  
    int status;  
    char buffer[8];  
        /* code of B */  
}
```



Question 7 (10 points): Each parts has 5 points.

(a) Consider the following code, which writes to a file named “somefile”.

```
fdesc = open ("datafile", (O_WRONLY | O_CREAT |O_TRUNC), 0740);
data = "Hello";
write( fdesc, data, strlen(data) );
lseek( fdesc, 8, SEEK_END );
data = "World";
write( fdesc, data, strlen(data) );
close(fdesc);
```

- What will be the length of “datafile” after executing the above code?
- Show the contents of the “datafile”.

(b) Consider the following Bash program script.

```
#!/bin/bash
listdata=`find TestDir -type f -exec stat -c%s {} \;`
var1=0
for data in listdata
do
    var1=`expr $var1 + $data`
done
echo $var1;
```

Suppose that TestDir contains the following files:

```
drwxrwx--- 2 tripathi tripathi 4096 Mar 5 19:20 Bye/
-rw-rw---- 1 tripathi tripathi 12 Mar 5 19:17 hello
```

Directory TestDir/Bye contains the following file:

```
-rw-rw---- 1 tripathi tripathi 17 Mar 5 19:18 bye
```

What will be the output produced when the above shell script program is executed?

Question 8 (10 points): Write a C function that takes a directory pathname as an argument, and prints the names of all regular files in that directory that have SUID bit true and also have both write and execute permissions for other category users.

- This function does NOT look into the nested directories.
- No need to perform error checking.

```
function void answer8 ( char * dirname) {  
    DIR *dpntr;
```

```
    dpntr = opendir( dirname );
```

