Sri Lanka Institute of Information Technology

B.Sc. Special Honours Degree/ Higher Diploma
in
Information Technology

Final Examination
Year 2, Semester 2 (2017)

IT201- Systems Programming and Design

Duration: 3 Hours

Instruction to Candidates:

- This paper Contains 4 questions. Answer All Questions.
- All questions carry equal marks.
- Total marks 100.
- This paper contains 7 pages including Cover Page.

Question 1 — 25 marks

a. List two components of an Operating System (2 marks)

b. Describe the Power-On-Self-Test operation (2 marks)

c. For each of the following, identify whether it is an absolute pathnames or a relative pathname? (3 marks)

i. SPD/MyDir/file123.c
ii. /home/nimal/test.txt
iii. ../nimal/test.txt

d. Compare and contrast hard links and soft links. (3 marks)

e. Sunil has logged into the system and he is currently in **logged user's home directory**. Answer the questions using the figure 1.

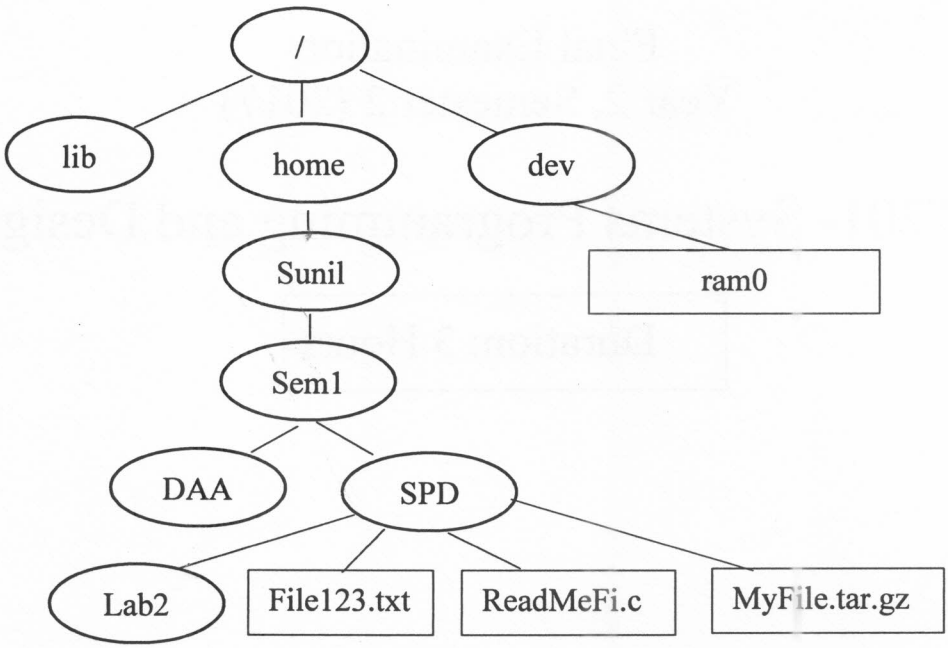

Figure 1: Directory Hierarchy

i. Sunil wanted to know the path to his working directory and he has entered **pwd** command. Write the expected output of the command. (1 mark)

ii. Write the path to **File123.txt** relative to the current working directory. (1 mark)

iii. Create a file called **Lab2File1.txt** in **Lab2** directory without changing the working directory. (2 marks)

iv. Write the command to get the output shown in figure 2 without changing the working directory. (2 marks)

```
brw-rw----. 1 Sunil root 1, 10 Aug 22 20:15 ram0
```

Figure 2: Output

v. Explain the output line shown in figure 2, in above part iv) (3 marks)

vi. Sunil has changed the working directory to **SPD** and executed **ls** command in UNIX along with the ***Fi*** file name substitution pattern **(ls *Fi*)** to list the content of the working directory. Write the expected output. (3 marks)

vii. Suppose that Sunil has created the archive file **MyFile.tar.gz** by using the **Lab2** directory and its content. Write the command to create that archive file and store in the directory which Sunil is currently working in. (3 marks)

Question 2 — 25 marks

a. List different types of variables in UNIX environment. (2 marks)

b. Sunil has executed list of UNIX commands as shown in figure 3.

```
Sunil@localhost ~]$ University=SLIIT
Sunil@localhost ~]$ bash
Sunil@localhost ~]$ echo $University

Sunil@localhost ~]$
```

Figure 3: UNIXCommands

i. What type of variable has Sunil created as University? (1 mark)

ii. What has Sunil done by executing **bash** UNIX command? (1 mark)

iii. Explain the reason of printing a blank line in the output of **echo** UNIX command. (2 marks)

iv. Explain (use UNIX command/s) how to overcome the problem you identified in above part iii). (2 marks)

c. Briefly explain following variables in bash shell. $$, $!, $2, $@ (4 marks)

d. Write a menu driven shell script to answer the following questions from i.) to v.) by referring the files shown in figure 4.

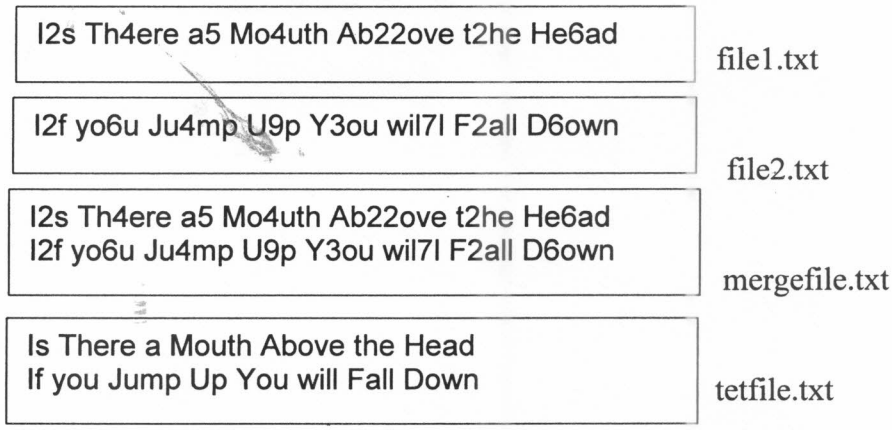

Figure 4: Files

i. Write a shell script code to print the following menu in the terminal. (2 marks)

[1] Merge
[2] Remove Digits
[3] Display File
[4] Exit/ Stop

Enter menu number: 1

ii. Write a shell function call mergeFiles ( ) to copy the content of file1.txt and file2.txt into the file call mergedfile.txt as in figure 4. *(Hint: pass the file names as parameters to the function)* (4 marks)

iii. Write a shell function call removeDigits ( ) to remove all digits in the content of mergedfile.txt file and pass the text into textfile.txt file. *(Hint: pass the file names as parameters to the function)* (3 marks)

iv. Write a shell function call print ( ) to print the content of textfile.txt file. *(Hint: pass the file name as parameters to the function)* (2 marks)

v. When the user selects option 1, invoke the mergeFiles function to merge file content. When user selects option 2, remove all digits of the mergedfile.txt and save in textfile.txt. User will be able to print the textfile.txt content by invoking the print function. Select option 4 to exit the program. (2 marks)

Question 3 — 25 marks

a. Define a process (2 marks)

b. Define a zombie process. (2 marks)

c. Compare and contrast system calls and library functions. Give two examples for each. (4 marks)

d. Answer the questions from i) to iii) using the C program given below figure 5.

```
#include <stdio.h>
int main()
{
        int no;
        printf("Hello World\n");
        Line A = fork();
        if(Line B )
        {
            printf("I am Child and Return Value=%d\n",Line C);
            printf("Child PID: %d\n", getpid());
            printf("Child's Parent PID: %d\n", getppid());
        }
        else if(Line D){
            printf("Parent PID: %d\n", getpid());
        }
        else {
            printf("Error in fork");

        }
                return 0;
}
```

Figure 5: C program

i. Complete the Line A, Line B, Line C and Line D of the above code. (4 marks)

ii. List three systems calls available in the above code. (3 marks)

iii. Modify the above code by printing the current date only by the parent process. (*Hint: Use only system call*) (2 marks)

e. Consider that you have a file called myfile.txt with some content. There exist another empty file called textdata.txt. Write a C program (prog.c) using file management system calls to do the followings. The program is accepting the files through command line arguments as follows.

./ prog myfile.txt textdata.txt

i. Accept the files to the program through command line arguments. Display an error message if they are not passed to the program. (2 marks)

ii. Open myfile.txt for reading. Display an error message if it is not success. (2 marks)

iii. Open textdata.txt file for writing. Display an error message if it is not success. (2 marks)

iv. Read myfile.txt file until the end and write the content of it inside the textdata.txt file. *(Hint: Assume the size of the file is 50 bytes)* (2 marks)

Question 4 — 25 marks

a. "Perl is a weakly typed language". Explain what weakly typed means. (2 marks)

b. Consider the following code segment.

```
@data = ("Harry", 250,"SLIIT", 35103, 6);
$a = @data;
$b = $data[4]++;
$c = --$data[3];
$d = $data[0]."potter"
```

What are the values of $a, $b , $c and $d after executing the code segment? (4 marks)

c. Write regular expressions to match. (2 marks)

i. Names of cricketers with 9 characters which starts with S or T (Ignore the case)

ii. Any Floating Point Number

d. Write a Perl subroutine named largerThan that accepts a number n, followed by a list of numbers as arguments and returns a list of numbers that are larger than n.
(Hint: if the subroutine is invoked as largerThan(3,5,7,2,6,8,9,4), then n=3 and it should return the list (5,7,6,8,9,4)) (5 marks)

e) A part of the UNIX password file is shown in below figure 6. It has following properties.

- No Column headings
- Each column is separated by a colon.
- Columns of the file are; Name of the user, Encrypted password, User ID, Group ID, Full name of the user, Home directory of the user and then the Shell of the user.

```
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
```

Figure 6: /etc/passwd file (password_file.txt)

You are required to write a PERL code to read the above file to obtain the home directory of the user and write the output to Output_file.txt as in figure 7. *(Hint: Consider only about the rows available in the figure 6 and write PERL syntax for the following requirements or write a complete code)*

```
User: bin, your home directory is root → bin
User: daemon, your home directory is  root → sbin
User: adm, your home directory is  root → var → adm
User: lp, your home directory is  root → var → spool → lpd
User: sync, your home directory is  root → sbin
User: shutdown, your home directory is  root → sbin
```

Figure 7: Output_file.txt

i. Open file "Password_file.txt" for reading. Display an appropriate error message if the file cannot be opened successfully. (2 marks)

ii. Open file "Output_file.txt" for writing. Display an appropriate error message if the file cannot be opened successfully. (2 marks)

iii. Read all the records in "Password_file.txt" line by line up to the end of the file. (1 mark)

iv. Remove any newline character in the current line. (1 mark)

v. Write the output to Output_file.txt as figure 7. (4 marks)

vi. Close the files. (2 marks)