Paper Code: IC2316

Roll No. 4052

M.Sc (Informatics) II Sem.-2016
Paper-IT-23, Operating Systems

Time: 3hrs

Maximum Marks:75

(Write your Roll No. on the top immediately on receipt of this question paper)
(Answer Question 1 Compulsory and any 4 from the rest)

Q1)

a) Briefly mention the similarities and differences bwteen a "PIPE" and a "NAMED PIPE".

b) You need to copy data of two source files to a destination file.

Write the steps needed to do this using only system calls (***open, read, write, close***). Mention any assumption that you have made.

c) Considering Linux Operating System, what is a Signal? Write an example of a Signal Handler.

d) What is a "zombie" process? How can we avoid creating zombie processes?

e) What are the advantages of a "Shared library" over a "Static library"? Mention the common file name extension for a shared and static library (Consider Linux Operating System)

[3 * 5]

Q2)

a) Consider the following code fragment and then answer the questions.

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
int myvar = 20;
int main()
{
        int ret = -1;
        ret = fork();
        if( ret == 0)
        {
            printf("Child : \n");
        myvar = myvar + 5;
        exit(myvar);
    }
    else
    {
        printf("Parent :\n");
        printf("The value in variable myvar is %d\n", myvar);
        exit(0);
    }
}
```

i) What is the purpose of the "fork()" system call? What are the return values of this system call? Mention the return value if this call fails, for some reason. [3]

ii) What will be the value of "myvar" printed by the parent? Briefly explain your answer. [2]

iii) Use this code fragment and add the following functionality to this

a) The parent waits for the completion of the child

b) The parent prints the Process Identifier of the child, after the child terminated and then it exits. [4]

b) "A parent process reads a file and sends the data over a ***pipe*** to a child process. The child process reads the data and displays on the screen. It should terminate when there is no more data to read". Write an algorithm to implement the above, using suitable system calls. [6]

Q3)

The file listing command "ls -li" is being executed in a "Linux" Shell and the output is shown below. Study the output and then answer the questions.

```
[localhost msc_iic]$ ls -li
total 4
656030 -rw-rw-r--. 2 user user    0 Apr 18 23:11 1.txt
656030 -rw-rw-r--. 2 user user    0 Apr 18 23:11 2.txt
655871 drwxrwxr-x. 2 user user 4096 Apr 18 23:10 dir
```

---

i)What is the significance of the first column of the output? For the first two rows the value of the first column is the same. Explain this observation. [4]

ii)What is the significance of the 3rd column of the output? Briefly mention how this will be different when you create a "regular file" and when you create a "directory". [4]

iii) if the output shown above is piped with standard Linux commands "grep" and "wc" in this manner *ls –l | grep "^d" | wc –l* , what will be the output? Briefly explain. [3]

b) Write a shell script, that will display the "***total file size of all regular files***" in your current working directory. [4]

Q4)

a) Briefly mention the purpose of a "make file". Write a makefile from the description given below.

"The name of your final executable should be ***myoutput***. It depends on four object files ***main.o, f1.o, f2.o, f3.o***. These object files depend of source files named as ***main.c, f1.c, f2.c, f3.c***. There is a single header file "***myheader.h***" which is referred by all these source files. The compiler that you need to use is "*gcc*". Also you need to specify a target which can remove all the object files and the final executable." [6]

b) Mention the difference between ***symbolic/soft*** link and ***hard*** link with an example. What is the purpose of ***readlink*** command in Linux? [3]

c) Can two processes running on two different machines (Having different Operating Systems) over a network communicate through a ***Message Queue***? Briefly explain. If you think this is not feasible, then write alternatives as to how these processes can communicate. [3]

d) Consider you have an executable file "***myoutput***" in your current directory (in your Linux System).The corresponding source code file is "*mycode.c*". You need to use "***gcc***" as the compiler and "gdb" as the debugger. Answer the following questions based on these information. [3]

1) Write the command to produce *myoutput* from *mycode.c* so that gdb can be used for debugging.

2) What are the gdb commands to invoke the executable, set a break point at the "main" function, execute the next line, set the value of a variable (say myvar) to 10.

Q5)

a) Considering a ***System V Message Queue***, answer the following questions

1) How can you create a system V message Queue?
2) Is there a specific format of message that can be used for a System V Message Queue?
3) Briefly explain the functions that can be used to send/receive messages.
4) How can you delete the message queue?

[8]

b) Suppose multiple writer processes are trying to write over the same shared memory, how can you ensure that one's data is not overwritten by another process? [3]

c) What is the difference between a ***semaphore*** and ***mutex***? [4]

Q6)

a) In the context of IPC (Inter Process Communication) objects, what is the difference between "***Process Persistence***" and "***Kernel Persistence***"?

Out of the given IPC objects classify them as "Process Persistent" or "Kernel Persistent" as appropriate.

IPC Objects – *PIPE, FIFO, SHARED MEMORY, MESSAGE QUEUE, SEMAPHORE.*

If you reboot your system which of the above IPC objects will still be available? (Consider that before the reboot there was one instance of each of these IPC objects) [8]

b) Consider two arbitrary commands, cmd1 and cmd2. You have entered ***cmd1|cmd2*** on the command prompt given by the shell. Based on this answer the following questions, [3]

i) What is the significance of the "|" character to the shell?

ii) Achieve the same effect with the help of ***redirection operators*** (< and >)

c) Considering a ***Shared Memory***, answer the following questions,

1) Is Shared memory a faster IPC mechanism than Message Queue?
2) What do you mean by ***attaching*** a Shared Memory to a process?
3) How can you delete a Shared Memory? [4]