DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF BARISHAL

4th Year, 1st Semester Final Examination

Admission Session: 2018-2019
Course Title: System Programming
Course Code: CSE-4105

Time: 02 Hours **Marks:** 60

N.B.: Answer any **FOUR** questions out of the followings. All parts of each question must be answered consecutively. Right side of the question shows the maximum marks.

1.a) Write short notes: 4
i) System Programming **ii)** Unix Kernel API **iii)** Shell Scripting **iv)** File Attributes

b) A file whose file descriptor is fd contains the following sequence of bytes: 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5. The following system calls are made: 6

```
lseek(fd, 3, SEEK_SET);
read(fd, &buffer, 4);
```

where the lseek call makes a seek to byte 3 of the file. What does buffer contain after the read has completed?

c) Depict the control flow of system calls. Do you agree/disagree that the system calls are time consuming? Give your statement. How do you change file permissions in Unix Explain with example. 5

2.a) **cd** [<**dir**>] means that change **pwd** to HOME directory, or <**dir**> if it is supplied. However, "Changing directories" and "being in a directory" are imprecise phrases. When you **cd** to a directory named **dir**, you may think of yourself as being "in **dir**", but this is not true. What is the true meaning for **cd** [<**dir**>] in order to resolve the relative pathnames? 4

b) In multi-user operating system, the operating system must protect users from each other and protect itself from users. However, while providing an operating environment for all users, the operating system creates this illusion (permission for read, write, execution, etc.) by creating data paths between user processes and devices and files. How does UNIX create this illusion? 6

c) Each user has a username and a number, the UID, why? Wouldn't it be simpler to record the username of the user as the owner of a file? Why not have a single identifier for each user? 5

3.a) A system call is like a conventional function call in that it causes a jump to a subroutine followed by a return to the caller. But it is significantly different. Distinguish between system call and function call based on the following keywords: *privileged/kernel mode, trap instruction, system dependent*. 5

b) Unix files have a full set of permission bits, including a `set-user-ID` bit and a `set-group-ID` bit. If you turn on the `set-group-ID` bit for a directory, does it have any effect? If so, what and why? If not, could you think of some use for this bit? 5

c) Analyze the following three commands: 5
i) ps –ax | grep cron
ii) ps -ax | tee processes.txt | more

4.a) The kernel had to locate a free `inode` and free disk blocks when it created a new file. How does the kernel know which blocks are free? How does the kernel know which `inodes` are free? What method does the file system on your machine use to keep track of unused blocks and `inodes`? 9

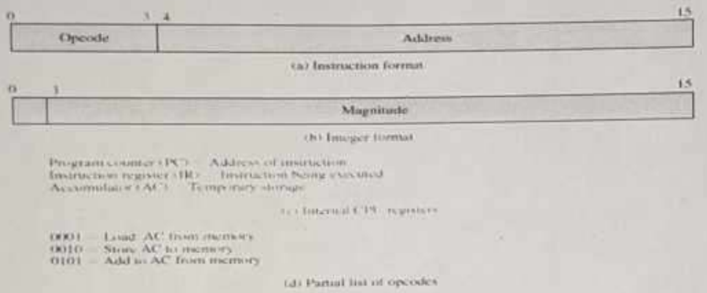
b) A directory is just a node in a set of linked nodes. Using the *pwd* command you can simply know the current directory. Based on the functionalities of *pwd* answer the following: **6**

i) What repetitive steps are performed to compute its current directory?
ii) How do we know when we read the top of the tree?
iii) How do we print the directory names in the correct order?

5.a) The hypothetical machine of **Fig. 1** also has two I/O instructions: **6**

0011 = Load AC from I/O	In these cases, the 12-bit address identifies a particular I/O device. Show the program execution (using the format of **Fig. 2**) for the following program:
0111 = Store AC to I/O	i) Load AC from device 5 ii) Add contents of memory location 940 iii) Store AC to device 6

Assume that the next value retrieved from device 5 is 3 and location 940 contains a value of 2.



(a) Instruction format

(b) Integer format

Program counter (PC) = Address of instruction
Instruction register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

(d) Partial list of opcodes

Fig. 1

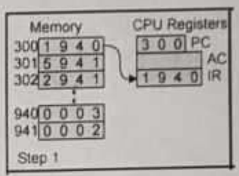


Fig. 2

b) In UNIX, there is a single file hierarchy and every accessible file is in this single file hierarchy, no matter how many disks are attached. Similar to windows, there is no such thing as the "C" drive" or "E" drive" in UNIX. If any new file system arrive then how UNIX adapt itself? Give example(s). **4**

c) How does `ls -l` work? Briefly explain the data flow in *who* command with figures. **5**

6.a) The stat system call is passed the name of a file and a pointer to a struct and fills the struct with information about the file. Explain, using the directory, inode, and data model, how stat works? **5**

b) How can we design a multithreaded program to count and print the total number of words in three files? Give two use case scenarios for clarification. **5**

c) Write short notes on the following: **5**

i) `pthread_create, pthread_join`
ii) `pthread_mutex_lock, pthread_mutex_unlock`

<u>Good Luck!!!</u>