1. What does ***ps aux*** command do?
2. What is the functionality of ***tee*** command?
3. How do you redirect the standard error to a file?
4. What will be the output of the command at a bash shell ? (Suppose your login name is cse)
5. $ echo ‘My name is $USER’
6. If a file foo has permission set at rwxr- - r --, how to change the permission to rwxrw-r--?
7. List all the filer command?
8. What is the functionality of ***ls [A-Z]\*[0-9]*** command?
9. What is the output after the following command sequence?

$ i=1

$ test $i -gt 0

$ echo $?

1. What is the result after the following expression sequence?

a=10

echo $((a<5))

1. what are the following positional parameters?

$\*, $?, $@, $#

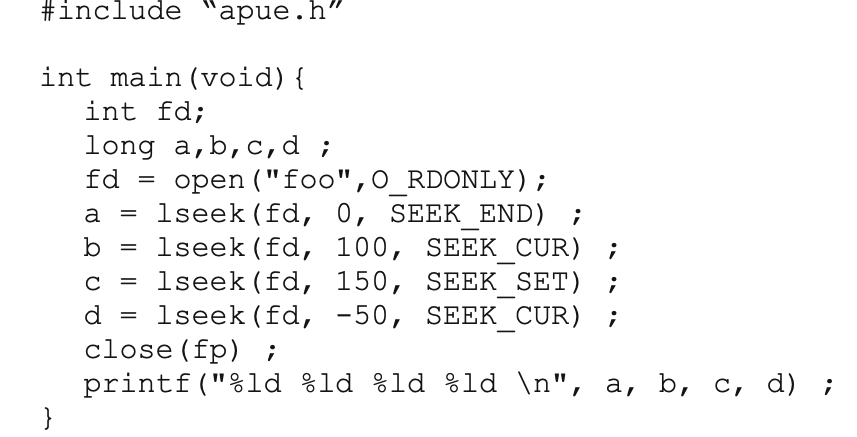
1. If you type the command who > /dev/null, what happens?
2. Explain the following pipeline

***Who | cut -c-8 | sort***

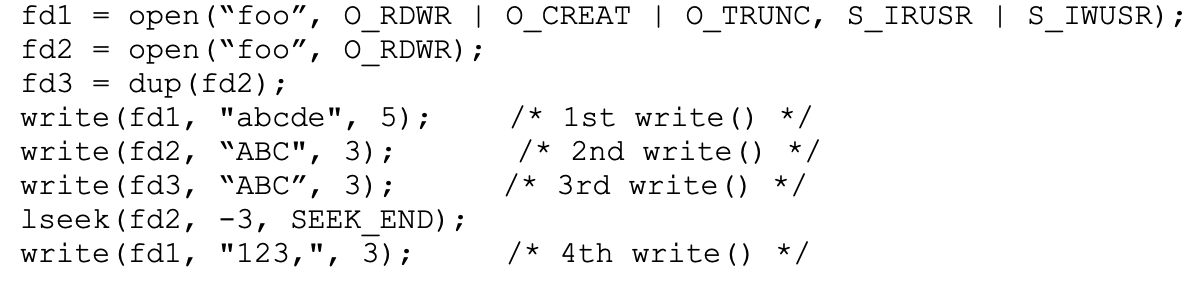
1. What is the difference between a program and a process?
2. What is the difference between the two positional parameters “$@” and “$\*”?
3. Use expr to write an expression equivalent to the built-in-arithmetic expression a=$((b\*10))
4. What is the file descriptor for STDOUT\_FILENO?
5. List the return value of read system call?
6. If we want to change the user ID of a file, which system function should we use?
7. By using umask() function prevents the write permission of group members and others, what number should use as parameter?
8. Which command can we use to show the inode number in Linux system?
9. What is the difference between stat() and lstat() system functions?
10. What is the difference between hard link and symbolic link?
11. Assume the file permission for the file “foo” is rwxr--r--, what is the file permission (in the same symbol format) after the following command?

Chmod u+s, g+s foo

1. What would be the output of the following program? The program is in a file called foo and has a size of exactly 200 bytes.



1. After each of the write() calls in the following code segment, explain what the content of the output file would be, and why



1. What is the default buffering scheme for standard output? How about standard error?
2. What system call is used to get the parent process ID of the calling process?
3. See the code below, and write how many times will “Hello” be printed and explain why:

**Int main(void){**

**Printf(“Hello\n”);**

**Fork();**

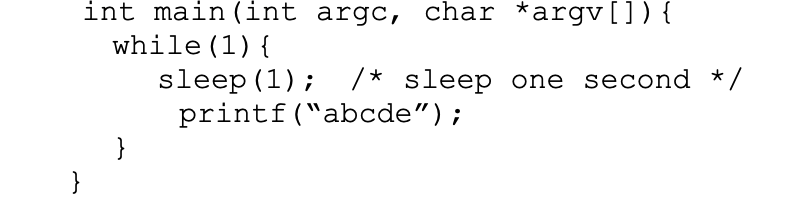
**Printf(“Hello\n”);**

**Fork();**

**Printf(“Hello”);**

**}**

1. **What are the return values of the fork() call?**
2. **Assume the buffer size is 500 bytes and the clock only runs on sleep() call, for the following piece of code, how many seconds will pass before you can see something on the screen? Please justify your answer**

****

**If the printf() is changed to printf(“abcde\n”), how many seconds will pass before you can see something on the screen? If the result is different with previous one, explain why.**