File Low level System Calls

Program A	Program B	Program C	

#include<stdio.h>
 #include<fcntl.h>
 main( )
 { int fd,i;char c;
  fd=open("xyz",O_RDONLY);
  i=lseek(fd,6,SEEK_SET);
  read(fd,&c,1);
  printf("%d%c",i,c);
 }
	int fd,i,j,k;char c;
	 fd=open("xyz",O_RDONLY);
	 i=lseek(fd,6,SEEK_SET);
	 j=lseek(fd,-2,SEEK_CUR);
	 lseek(fd,j,SEEK_SET);
	 read(fd,&c,1);
	 printf("%d%c\n",j,c);
	 k=lseek(fd,-2,SEEK_END);
	 lseek(fd,k,SEEK_SET);
	 read(fd,&c,1);
	 printf("%d%c\n",k,c);
		int fd,i;char fn[20];
		 printf("Give filename");
		 gets(fn);
		 fd=open(fn,O_WRONLY);
		 write(fd,"BCD",3);
		 lseek(fd,2,SEEK_CUR);
		 write(fd,"KLC",2);	

0	1	2	3	4	5	6	7	8	9
a	b	c	d	e	f	g	h	i	\n

Program A: Opens a file named xyz and prints its 6th letter. Let the contents of file xyz be abcdefghi then output is 6g. The lseek system call carries the file pointer at 6th location.

The file pointer can be moved by using lseek system call. SEEK_SET means relative to the beginning to the file. SEEK_CUR means relative to the current position and SEEK_END means from end of the file. In place of SEEK_SET, SEEK_CUR and SEEK_END 0,1 and 2 respectively can also be used.

Program B: The 1st lseek carries the file pointer to 6th location. The 2nd to (6-2)=4th. o/p 4e.

There are 10 letters in the file hence the last lseek will carry the file pointer to (10-2)=8th position. o/p 8i.

Program C: Let initial contents are abcdefghi then after the execution of the program the contents will be BCDdeKLhi. The third argument in write system call says how many characters should be written. In write(fd,"KLC",2) though three characters are given, yet only first two will be written. write(fd,a,4) will write a[0], a[1], a[2], and a[3].

int d;
 d=open("s",O_CREAT|O_WRONLY,0666);
 printf("%d",d);write(d,"ab",2);
 lseek(3,1,0); write(3,"yz",2);
	File "s" initially	pqrst	not exists

	finally the file "s" is	ayzst	ayz
	O_CREAT|O_WRONLY|O_APPEND	pqrstabyz	abyz
chmod +r s	O_EXCL|O_WRONLY	ayzst	error
	O_TRUNC|O_WRONLY	ayz	error

Program D	Program E	Program F	

int fd,i;char a[10],fn[20];
 fd=open("xyz",O_RDWR);
 i=lseek(fd,-4,SEEK_END);
 lseek(fd,i,SEEK_SET);
 read(fd,a,3);
 lseek(fd,0,SEEK_SET);
 write(fd,a,3);
	int fd1,fd2;char c;
	 fd1=open("xyz",O_RDONLY);
	 fd2=open("pqr",O_WRONLY);
	 lseek(fd1,6,SEEK_SET);
	 read(fd1,&c,1);
	 write(fd2,&c,1);
		int fd,i;char a[10];
		 a[3]=78;
		 fd=open("exp",O_RDONLY);
		 i=read(fd,a,5);
		 printf("%d%d%d",i,a[0],a[1]);
		 printf("%d %d",a[2],a[3]);	

Program D: copies last three characters of file "xyz" in the beginning. If initial contents of file "xyz" are abcdefgh then after execution of the above program the contents will be fghdefgh.

Program E: Reads 6th character from file xyz and puts it in the beginning of file pqr. If in above we replace file name pqr by xyz then in the same file the above change will be made. There are two open instances of file xyz.

Program F: The read system call returns the number of the letters actually read. In a file named "exp" if we put three letters abc, the output will be 4 97 98 99 10. If in file exp we write only two characters BC the output will be 3 66 67 10 78. Only 1 lettter K o/p 2 75 10 _ 78. No letter 0 _ _ _ 78. When File exp does

not exist then i is –1. The read system call returns –1 if read can not be performed. fd=open("exp",O_WRONLY); i=read(fd,a,5);

To find size of a file (when <100000).   s=read(fd,a,100000).

1. Write program, which finds the location of first a. Let a file has erapouseraty then output is 3.
2. Program to find the size of a file. no SEEK_END. [Hint: Read every letters till i=read(fd,&c,1) returns 0.]
3. Write program to print the entire file. It is like cat command.
4. Program to replace 4th letter of a file by 'K'. (eraKouseraty.)   (A) Write program to replace all a's by K's.
5. Program reads two file names. In first file the contents of the second file are copied. It is like (cp) copy.
6. Program reads two file names. In first file the contents of the second file are put at the end. Join.
7. Write program to print a file in reverse order (Do not use array).
8. Write a program, which reads an integer (k). The program exchanges kth and kth last letters of the file.
9. Program to reverse a file (no array). After the execution of this program the above file look as ytaresuopare.
10. Write program, which reads a file name and a string. The string is put at the end of the file.
11. Write program, which puts a character 'x' in the beginning of a file. (A) Delete first letter of file
12. Inserts 'x' in between adjacent characters. The above file will look as exrxaxpxoxuxsxexrxaxtxy.
13. Assume that file has numbers. Print the last number.