VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

OPERATING SYSTEMS

Experiment 10

AMIT KUMAR

19BCE1281

Aim: Determine the file read time for sequential and random access based on varying sizes of the file. Draw a graph log plot of Size of files vs Average per block time.

CODE-1:

```
#include <stdio.h>
int main()
{
 FILE *fh;
 int ch;
 fh=fopen("file.txt","r");
 if(fh==NULL)
 {
 puts("Can't open that file!");
 exit(1);
 }
 while((ch=fgetc(fh))!=EOF)
 putchar(ch);
 fclose(fh);
 return(0);
}
```

OUTPUT:

```
amit@amit-kumar-operat-VirtualBox:~$ gcc c1.c
c1.c: In function ‘main’:
c1.c:10:3: warning: implicit declaration of function ‘exit’ [-Wimplicit-functio
n-declaration]
   10 |   exit(1);
      |   ^~~~
c1.c:10:3: warning: incompatible implicit declaration of built-in function ‘exi
t’
c1.c:2:1: note: include ‘<stdlib.h>’ or provide a declaration of ‘exit’
    1 | #include<stdio.h>
  +++ |+#include <stdlib.h>
    2 | int main()
amit@amit-kumar-operat-VirtualBox:~$ ./a.out
can't open that file!
amit@amit-kumar-operat-VirtualBox:~$
```

CODE-2:

```
    #include <stdio.h>
    int main ()
    {
      FILE *fp;
      int c;
      fp = fopen("file.txt","w+");
      fputs("This is study.com", fp);
      // we are using fseek to shift the file pointer to the 7th
position
      fseek( fp, 16, SEEK_SET );

    //Now we overwrite C programming in the 7th position
      fputs(" C Programming", fp);

    //now we print the current position of the file pointer using
ftell
      printf("The current position of the file pointer is before
rewind: %ld\n", ftell(fp));
    //we take the file pointer to the beginning of the file
      rewind(fp);

    //now we verify if rewind() worked using ftell
      printf("The current position of the file pointer is - after
rewind: %ld\n", ftell(fp));
      while(1) {
       c = fgetc(fp);
       if( feof(fp) ) {
        break;
       }
       printf("%c", c);
      }
       printf("\n");
      fclose(fp);
```

```
      return(0);
    }
```

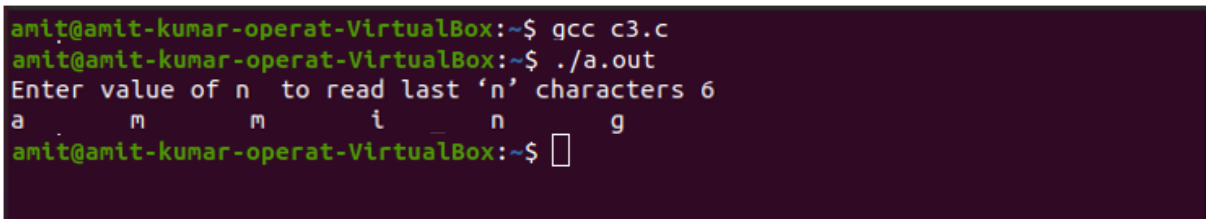
OUTPUT:

```
amit@amit-kumar-operat-VirtualBox:~$ gcc c2.c
amit@amit-kumar-operat-VirtualBox:~$ ./a.out
The current position of the file pointer is before rewind: 30
The current position of the file pointer is - after rewind: 0
This is study.co C Programming
amit@amit-kumar-operat-VirtualBox:~$
```

CODE-3:

```
     #include<stdio.h>
void main()
{
     FILE *fp;
     char ch;
     int n;
     fp=fopen("hello1.txt", "r");
     if(fp==NULL)
      printf("file cannot be opened");
     else
     {
        printf("Enter value of n  to read last â€˜nâ€™ characters");
        scanf("%d",&n);
        fseek(fp,-n,2);
        while((ch=fgetc(fp))!=EOF)
        {
             printf("%c\t",ch);
          }
      }
  fclose(fp);
 // getch();
}
```

OUTPUT:



CODE:

```
#include<stdio.h>
#include<string.h>
#include<time.h>
#include<unistd.h>
#include<sys/time.h>
int main()
{
	char data[40]="This is os lab. We are in 10th exp";
	char data1[600],ch;
	int i,j,res,seq[15],ran[15],n;
	long int size[15];
	printf("enter the value of n to read last 'n' charecters");
	scanf("%d",&n);
	struct timeval t1,t2;
	FILE *f[15];
	//creating 15 text files files
	f[1]=fopen("file1.txt","w+");
	f[2]=fopen("file2.txt","w+");
	f[3]=fopen("file3.txt","w+");
	f[4]=fopen("file4.txt","w+");
	f[5]=fopen("file5.txt","w+");
	f[6]=fopen("file6.txt","w+");
	f[7]=fopen("file7.txt","w+");
	f[8]=fopen("file8.txt","w+");
	f[9]=fopen("file9.txt","w+");
	f[10]=fopen("file10.txt","w+");
	f[11]=fopen("file11.txt","w+");
	f[12]=fopen("file12.txt","w+");
	f[13]=fopen("file13.txt","w+");
	f[14]=fopen("file14.txt","w+");
	f[0]=fopen("file0.txt","w+");
	//writing into files
	for(i=0;i<15;i++)
	{
		if(f[i]==NULL)
		{
			printf("file%d file failed to open",i);
		}
		else
		{
			if(strlen(data)>0)
			{
```

```
                        for(j=0;j<i;j++)
                        {
                                fputs(data,f[i]);
                                fputs("\n",f[i]);
                        }
                }
        }
        }
        //finding file size
        for(i=0;i<15;i++)
        {
                fseek(f[i],0L,SEEK_END);
                size[i]=ftell(f[i]);
                printf("size[%d]: %ld\n",i,size[i]);
        }
        //finding time for sequential access
        for(i=0;i<15;i++)
        {
                gettimeofday(&t1,NULL);
                while(fgets(data1,250,f[i])!=NULL)
                {

                }
                gettimeofday(&t2,NULL);
                long seconds=(t2.tv_sec - t1.tv_sec);
                seq[i]=(((seconds*1000000) + t2.tv_usec)-
(t1.tv_usec));
                printf("seq[%d]: %d\n",i,seq[i]);
        }
        //finding time for random access
        for(i=0;i<15;i++)
        {
                if(f[i]==NULL)
                {
                        printf("file%d file failed to open",i);
                }
                else
                {
                        gettimeofday(&t1,NULL);
                fseek(f[i],-n,2);
                while((ch=fgetc(f[i]))!=EOF)
                {

                }
        }
        gettimeofday(&t2,NULL);
        long seconds=(t2.tv_sec - t1.tv_sec);
        ran[i]=(((seconds*1000000) + t2.tv_usec)-(t1.tv_usec));
        printf("ran[%d]: %d\n",i,ran[i]);
        fclose(f[i]);
        }
        return(0);
}
```

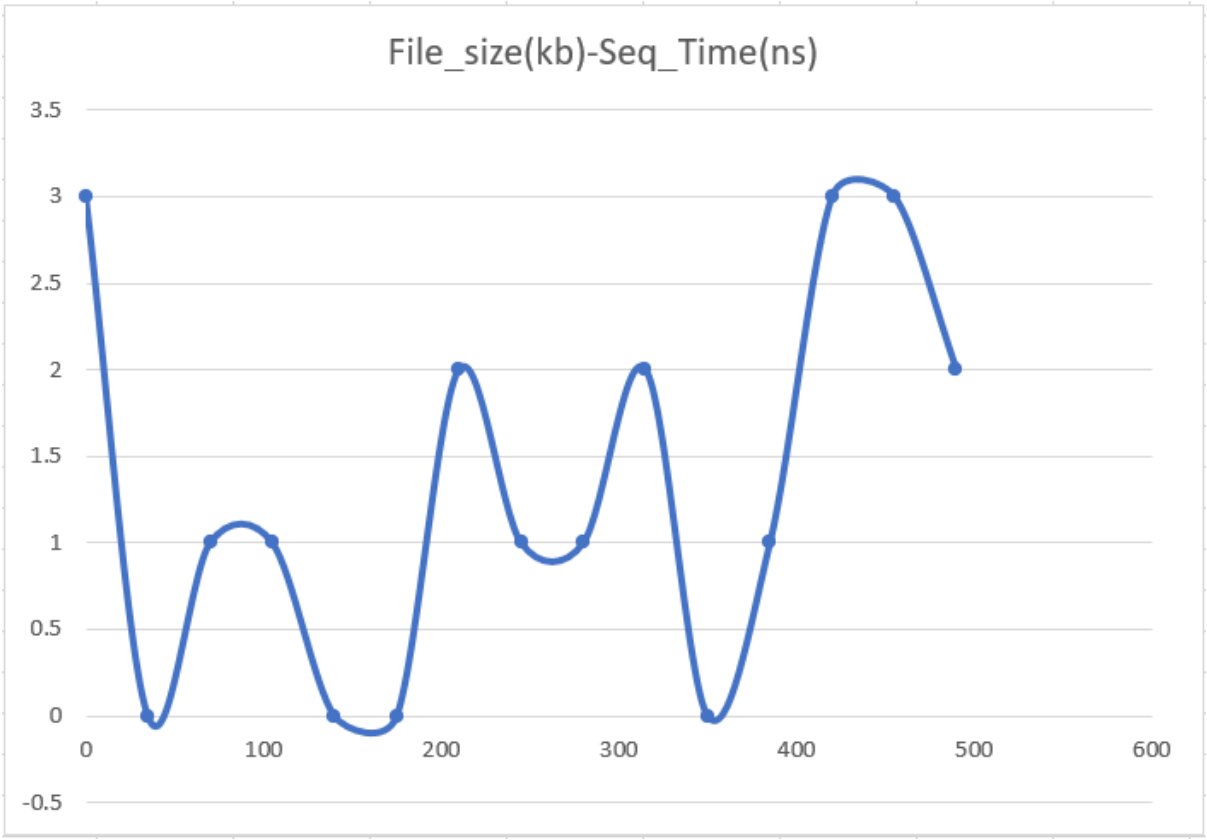
OUTPUT:

```
amit@amit-kumar-operat-VirtualBox:~$ gcc sam1.c
amit@amit-kumar-operat-VirtualBox:~$ ./a.out
enter the value of n to read last 'n' charecters 20
size[0]: 0
size[1]: 35
size[2]: 70
size[3]: 105
size[4]: 140
size[5]: 175
size[6]: 210
size[7]: 245
size[8]: 280
size[9]: 315
size[10]: 350
size[11]: 385
size[12]: 420
size[13]: 455
size[14]: 490
seq[0]: 3
seq[1]: 0
seq[2]: 1
seq[3]: 1
seq[4]: 0
seq[5]: 0
seq[6]: 2
seq[7]: 1
seq[8]: 1
seq[9]: 2
seq[10]: 0
seq[11]: 1
seq[12]: 3
seq[13]: 3
seq[14]: 2
ran[0]: 7
ran[1]: 3
ran[2]: 2
ran[3]: 2
ran[4]: 1
ran[5]: 2
ran[6]: 1
ran[7]: 2
ran[8]: 2
ran[9]: 2
ran[10]: 1
ran[11]: 2
ran[12]: 2
ran[13]: 2
ran[14]: 1
amit@amit-kumar-operat-VirtualBox:~$
```

GRAPH LOG PLOT:

	File_size	Seq_Time	Ran_Time
1	0	3	7
2	35	0	3
3	70	1	2
4	105	1	2
5	140	0	1
6	175	0	2
7	210	2	1
8	245	1	2
9	280	1	2
10	315	2	2
11	350	0	1
12	385	1	2
13	420	3	2
14	455	3	2
15	490	2	1

FILE-SIZE VS AVERAGE-SEQUENTIAL_TIME:



FILE-SIZE VS AVERAGE-RANDOM_TIME: