

## ADVANCED UNIX PROGRAMMING 2017 - LAB 1

Akash Sarda 111403003

Aditya Malu 111403023

Jassim Adbul Rehman 111403019

---

1. Assume that you have to read 10 characters from the beginning of an existing file and then to write “hello” to the end of the file. Write a program to achieve this without using lseek function.

Test Case1:

```
>./a.out
```

Not enough arguments.

```
./a.out <filename>
```

```
>./a.out sample2
```

```
I wandered lonely as a cloud  
That floats on high o'er vales and hills,  
When all at once I saw a crowd,  
A host, of golden daffodils;  
Beside the lake, beneath the trees,  
Fluttering and dancing in the breeze.  
Hello
```

```
>./a.out sample2
```

file is less than 10 characters.

: Success

```
>cat sample2
```

```
I w  
hello
```

```
>./a.out aaa
```

Cannot open: Is a directory

```
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question1
File Edit View Search Terminal Help
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question1 $ ./a.out
Not enough arguments.
./a.out <filename>
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question1 $ ./a.out sample2
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question1 $ cat sample2
I wandered lonely as a cloud
That floats on high o'er vales and hills,
When all at once I saw a crowd,
A host, of golden daffodils;
Beside the lake, beneath the trees,
Fluttering and dancing in the breeze.
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question1 $ ./a.out sample2
file is less than 10 characters.
: Success
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question1 $ cat sample2
I w
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question1 $ ./a.out aaa
Cannot open: Is a directory
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question1 $
```

2. Linux provides a function as given below to truncate file to specific length. `int truncate (const char *path, off_t len);` return 0 on success. On error, return -1, Write a program to emulate this function. Use `cat` command to demonstrate.

### Case 1: Normal Character file truncation.

```
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2
File Edit View Search Terminal Help
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ ls -l
total 28
-rw-r--r-- 1 sauron sauron 786 Aug 12 15:58 2.c
drwxr-xr-x 2 sauron sauron 4096 Aug 12 15:56 aaa
-rwxr-xr-x 1 sauron sauron 9056 Aug 12 15:57 a.out
-rw-r--r-- 1 sauron sauron 841 Aug 12 13:17 sample
-rw-r--r-- 1 sauron sauron 841 Aug 12 15:59 sample2
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ cat sample2
I wandered lonely as a cloud
That floats on high o'er vales and hills,
When all at once I saw a crowd,
A host, of golden daffodils;
Beside the lake, beneath the trees,
Fluttering and dancing in the breeze.

Continuous as the stars that shine
And twinkle on the milky way,
They stretched in never-ending line
Along the margin of a bay:
Ten thousand saw I at a glance,
Tossing their heads in sprightly dance.

The waves beside them danced; but they
Out-did the sparkling waves in glee:
A poet could not but be gay,
In such a jocund company:
I gazed—and gazed—but little thought
What wealth the show to me had brought:

For oft, when on my couch I lie
In vacant or in pensive mood,
They flash upon that inward eye
Which is the bliss of solitude;
And then my heart with pleasure fills,
And dances with the daffodils.
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ ./a.out sample2
```

## Case 2:

```
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2
File Edit View Search Terminal Help
Fluttering and dancing in sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ ./a.out sample2 600
file is less than 599 characters
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ cat sample2
I wandered lonely as a cloud
That floats on high o'er vales and hills,
When all at once I saw a crowd,
A host, of golden daffodils;
Beside the lake, beneath the trees,
Fluttering and dancing in sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ od sample2
0000000 020111 060567 062156 071145 062145 066040 067157 066145
0000020 020171 071541 060440 061440 067554 062165 005040 064124
0000040 072141 063040 067554 072141 020163 067157 064040 063551
0000060 020150 023557 071145 073040 066141 071545 060440 062156
0000080 064040 066151 071554 020054 053412 062550 020156 066141
0000100 020154 072141 067440 061556 020145 020111 060563 020167
0000120 020141 071143 073557 026144 005040 020101 067550 072163
0000140 020054 063157 063440 066157 062544 020156 060544 063146
0000160 062157 066151 035563 005040 062502 064563 062544 072040
0000180 062550 066040 065541 026145 061040 067145 060545 064164
0000200 072040 062550 072040 062562 071545 020054 043012 072554
0000220 072164 071145 067151 020147 067141 020144 060544 061556
0000240 067151 020147 067151 000040 000000 000000 000000 000000
0000260 000000 000000 000000 000000 000000 000000 000000 000000
*
0001120 000000 000000 000000 000000 2. Linux provides a function as given below to truncate file to specific
0001130 length. int truncate(const char *path, off_t len); return 0 on success.
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ od -c sample2
0000000
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ od -c sample2
0000000 I w a n d e r e d l o n e l
0000020 y a s a c l o u d \n T h
0000040 a t o f l o a t s o n h i g
0000060 h o ' e r v a l e s a n d
0000080 h i l l s , \n W h e n a l
0000100 l a t o n c e I s a w
0000120 a c r o w d , \n A h o s t
0000140 , o f g o l d e n d a f f
0000160 , o f g o l d e n d a f f
```

## Case 3:

```
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2
File Edit View Search Terminal Help
0000000 020111 060567 062156 071145 062145 066040 067157 066145
0000020 020171 071541 060440 061440 067554 062165 005040 064124
0000040 072141 063040 067554 072141 020163 067157 064040 063551
0000060 020150 023557 071145 073040 066141 071545 060440 062156
0000080 064040 066151 071554 020054 053412 062550 020156 066141
0000100 020154 072141 067440 061556 020145 020111 060563 020167
0000120 020141 071143 073557 026144 005040 020101 067550 072163
0000140 020054 063157 063440 066157 062544 020156 060544 063146
0000160 062157 066151 035563 005040 062502 064563 062544 072040
0000180 062550 066040 065541 026145 061040 067145 060545 064164
0000200 072040 062550 072040 062562 071545 020054 043012 072554
0000220 072164 071145 067151 020147 067141 020144 060544 061556
0000240 067151 020147 067151 000040 000000 000000 000000 000000
0000260 000000 000000 000000 000000 000000 000000 000000 000000
*
0001120 000000 000000 000000 000000
0001130
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ od -c
0000000
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab1/AUP/Lab1/question2 $ od -c sample2
0000000 I w a n d e r e d l o n e l
0000020 y a s a c l o u d \n T h
0000040 a t o f l o a t s o n h i g
0000060 h o ' e r v a l e s a n d
0000080 h i l l s , \n W h e n a l
0000100 l a t o n c e I s a w
0000120 a c r o w d , \n A h o s t
0000140 , o f g o l d e n d a f f
0000160 , o f g o l d e n d a f f
0000180 o d i l l s ; \n B e s i d e
0000200 h e l a k e , \n b e n e a t h
0000220 t h e t r e e s , \n F l u t
0000240 t t e r i n g a n d d a n c
0000260 i n g i n \0 \0 \0 \0 \0 \0 \0 \0
0000280 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
0000300 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
0000320 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
*
0001120 \0 \0 \0 \0 \0 \0 \0 \0
0001130
```

3.What will be the output for the program with following operation?

a.Create a new file “f1” and write “abcde” in it and close

b.Open the file “f1” for writing with O\_APPEND flag

c.lseek to the beginning of the file

d.Replace the existing data in the file with “12345”

Justify your answer.

Answer : The file will be "abcde12345" at the end of all the steps.

Since the file is opened with the O\_APPEND flag, the contents always get written at the end of the file.

-- A snippet from manpage of "write" (man 2 write) --

```
File Edit View Search Terminal Help
WRITE(2) Linux Programmer's Manual WRITE(2)
NAME
  write - write to a file descriptor
SYNOPSIS
  #include <unistd.h>

  ssize_t write(int fd, const void *buf, size_t count);
DESCRIPTION
  3.What will be the output for the program with following operation?
  write() writes up to count bytes from the buffer pointed buf to the file referred to by the file descriptor fd.

  The number of bytes written may be less than count if, for example, there is insufficient space on the underlying physical
  medium, or the RLIMIT_FSIZE resource limit is encountered (see setrlimit(2)), or the call was interrupted by a signal handler
  after having written less than count bytes. (See also pipe(7).)
  a.Replace the existing data in the file with "12345"
  For a seekable file (i.e., one to which lseek(2) may be applied, for example, a regular file) writing takes place at the cur-
  rent file offset, and the file offset is incremented by the number of bytes actually written. If the file was open(2)ed with
  O_APPEND, the file offset is first set to the end of the file before writing. The adjustment of the file offset and the
  write operation are performed as an atomic step.

  POSIX requires that a read(2) which can be proved to occur after a write() has returned returns the new data. Note that not
  all filesystems are POSIX conforming.
RETURN VALUE
  A snippet from manpage of "write" (man 2 write)
  On success, the number of bytes written is returned (zero indicates nothing was written). It is not an error if this number
  is smaller than the number of bytes requested; this may happen for example because the disk device was filled. See also
  NOTES.

  On error, -1 is returned, and errno is set appropriately.

  If count is zero and fd refers to a regular file, then write() may return a failure status if one of the errors below is
  detected. If no errors are detected, or error detection is not performed, 0 will be returned without causing any other
  effect. If count is zero and fd refers to a file other than a regular file, the results are not specified.
Manual page write(2) line 1 (press h for help or q to quit)
```

4.Write a program to create a file with a hole: write any 10 bytes at an offset of 10 and another 10 bytes at an offset of 30. Using

“system” function, invoke “od” command and view the contents. Later copy the contents of the file to another file without writing the bytes of 0. Once again verify the contents by invoking “system” with “od”.

