

William Shreeve

Matt Kennedy

CIS 452-10

Lab 1 - Introduction to UNIX
and System Development Tools

1. The SYNOPSIS section describes the necessary libraries to include, as well as all function signatures. It also shows any additional requirements to implement the function.

The RETURN VALUE section describes what the given function is intended to return, as well as any relevant information. In malloc, for example, the man page describes the differences between what each memory allocation function returns.

2. The user command 'write' sends a message to another user. The description states: **Write** allows you to communicate with other users, by copying lines from your terminal to theirs.

The system call 'write' writes to a file descriptor. The description states: **write()** writes up to a parameter number of bytes from the buffer starting at the parameter 'buf' referred to by the file description parameter 'fd'

3. SEEK_SET macro is used in fseek to search from the beginning of a file. This is detailed in the description part of the fseek man page.
4. ls -al; the -a flag means show all files, and the -l flag means show in long mode.
5. chmod 600; chmod sets accessibility according to 3 groups; owner, group, and other. This corresponds to the 3 numbers after chmod. Using 600 means only the owner can read and write.

6.

```
Script started on 2019-01-10 08:36:11-05:00 [TERM="xterm" TTY="/dev/pts/1" COLUMNS="76"
LINES="45"]
```

```
J0;shreevew@eos10:~/CIS452/Lab1[shreevew@eos10 Lab1]$ gcc Sample1.c -lm -g -Wall
```

```
J0;shreevew@eos10:~/CIS452/Lab1[shreevew@eos10 Lab1]$ ./a.out
```

```
Hello, world.
```

```
You are the 268435456.000000 person to write this program!
```

```
J0;shreevew@eos10:~/CIS452/Lab1[shreevew@eos10 Lab1]$ gdb a.out
```

```
GNU gdb (GDB) 8.2
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
```

```
This is free software: you are free to change and redistribute it.
```

```
There is NO WARRANTY, to the extent permitted by law.
```

```
Type "show copying" and "show warranty" for details.
```

```
This GDB was configured as "x86_64-pc-linux-gnu".
```

```
Type "show configuration" for configuration details.
```

```
For bug reporting instructions, please see:
```

```
<http://www.gnu.org/software/gdb/bugs/>.
```

```
Find the GDB manual and other documentation resources online at:
```

```
<http://www.gnu.org/software/gdb/documentation/>.
```

```
For help, type "help".
```

```
Type "apropos word" to search for commands related to "word"...
```

```
Reading symbols from a.out...done.
```

```
(gdb) b main
```

```
Breakpoint 1 at 0x1151: file Sample1.c, line 6.
```

```
(gdb) run
```

Starting program: /home/shreevew/CIS452/Lab1/a.out

Breakpoint 1, main () at Sample1.c:6

```
6      double num = 0.0;
```

(gdb) s

```
7      printf ("Hello, world.\n");
```

(gdb) p num

\$1 = 0

(gdb) s

Hello, world.

```
8      num = pow(2, 28);
```

(gdb) s

```
9      printf ("You are the %f person to write this program!\n", num);
```

(gdb) p num

\$2 = 268435456

(gdb) s

You are the 268435456.000000 person to write this program!

```
10     return 0;
```

(gdb) s

```
11 }
```

(gdb) s

0x00007ffff7c6d223 in __libc_start_main () from /usr/lib/libc.so.6

(gdb) s

Single stepping until exit from function __libc_start_main,

which has no line number information.

[Inferior 1 (process 3869) exited normally]

(gdb) q

@0:shreevew@eos10:~/CIS452/Lab1[shreevew@eos10 Lab1]\$ exit

Script done on 2019-01-10 08:36:46-05:00 [COMMAND_EXIT_CODE="0"]

7. The memory leak issue is with data2; the memory is never freed after printing, so memory is allocated every loop and never freed. The fix for this is just adding a free statement after the print statement. Additionally, if the user quits, the memory from data1 is never free'd.
8. The command used was strace -c ./a.out. The summary total number of calls after entering a name in the first loop, then exiting on the next was 3 write calls.
9. The primary C library subroutine that causes the system call to be invoked is printf.
10. See attached.