Matt Frretti and Hector Garcia-Pacheco

Lab 01 - Introduction to UNIX and System Development Tools:

Online Help:

1. Briefly describe the purpose of (what is found in) any two sections commonly found on a man page (e.g. the synopsis section of the malloc() system call)

   Section 2 - System Calls: chmod(): changes the file mode or Access Control Lists.
   Section 3 - Library Functions: atio(): this function converts a string to integer.

2. Describe the difference between the UNIX shell command write and the UNIX system call write()

   The UNIX shell command *write* allows a user to write to the terminal of another user while the UNIX system call *write()* writes up to a specified amount of bytes from the buffer to the referred file.

Include Files:

3. What is the meaning of the stream-based SEEK_SET macro?

   The *SEEK_SET* to macro moves the file pointer position to the beginning of the file.

Useful UNIX Fundamentals:

4. What is the command to list the contents of a directory in long mode, including hidden files?

   ```
   ls -la
   ```

5. What is the command Syntax to make a directory readable/writable/executable only by you?

   ```
   chmod 700 directoryName
   ```

6. Screenshots attached as separate files

7. Describe precisely (nature of problem, location) the memory leak(s) in sample program.

   If the user were to enter ‘quit’ instead of their username the space allocated for variable data1 and possibly data2 will never be freed as entering ‘quit’ breaks out of the loop. To fix this, free data1 and data2 if the user enters quit.

8. Fix the problem(s) and submit your corrected source code.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define SIZE 16
int main() {
  char *data1, *data2;
  int i;
  do {
    data1 = malloc (SIZE);
    printf ("Please input your eos username: ");
    scanf ("%s", data1);

    if (!strcmp (data1, "quit")){
      free (data1);
      free (data2);
      break;
    }

    data2 = malloc (SIZE);

    for (i=0; i<SIZE; i++)
      data2[i] = data1[i];

    printf ("data2 :%s:\n", data2);
    free(data2);
    free (data1);

  } while (1);


 return 0; }
```

9. How many times is the write() system call invoked when running sample program 2? Experiment with executing the loop a different number of times. Then express your answer as a formula.

   The write system call was invoked 3 times. The Formula to determine how many time the write system call will be invoked is: (#ofLoops * 3) - 3

10. Examine the source code and output to answer the question: what is the primary 'C' library subroutine that causes the write() system call to be invoked while executing sample program 2?

    The printf statement in the standard C library ‘stdio’ is the primary subroutine that causes the system call write().