**Buffers**

1. Commented Code

#include <fcntl.h>

#include <stdlib.h>

#include <unistd.h> //Define header files

#include <stdio.h>

#define BUF\_SIZE 500 //Define Buffer size as 500.

#define OUTPUT\_MODE 0700 //Define file permission.

int main(int argc, char \*argv[])

{

int in\_fd, out\_fd;

int rd\_size = 1, wr\_size;

char buf[BUF\_SIZE]; //Declare buffer.

if (argc != 3)

exit(1);

in\_fd = open(argv[1], O\_RDONLY); //Open input file.

if (in\_fd < 0)

exit(2);

out\_fd = creat(argv[2], OUTPUT\_MODE); //Create output file.

if (out\_fd < 0)

exit(3);

while (rd\_size > 0) {

rd\_size = read(in\_fd, buf, BUF\_SIZE); // Continuously read from input file into buffer.

if (rd\_size <0)

exit(4);

wr\_size = write(out\_fd, buf, rd\_size); // Continuously write from buffer into output file.

if (wr\_size<=0){

close(in\_fd); //Close both of the files.

close(out\_fd);

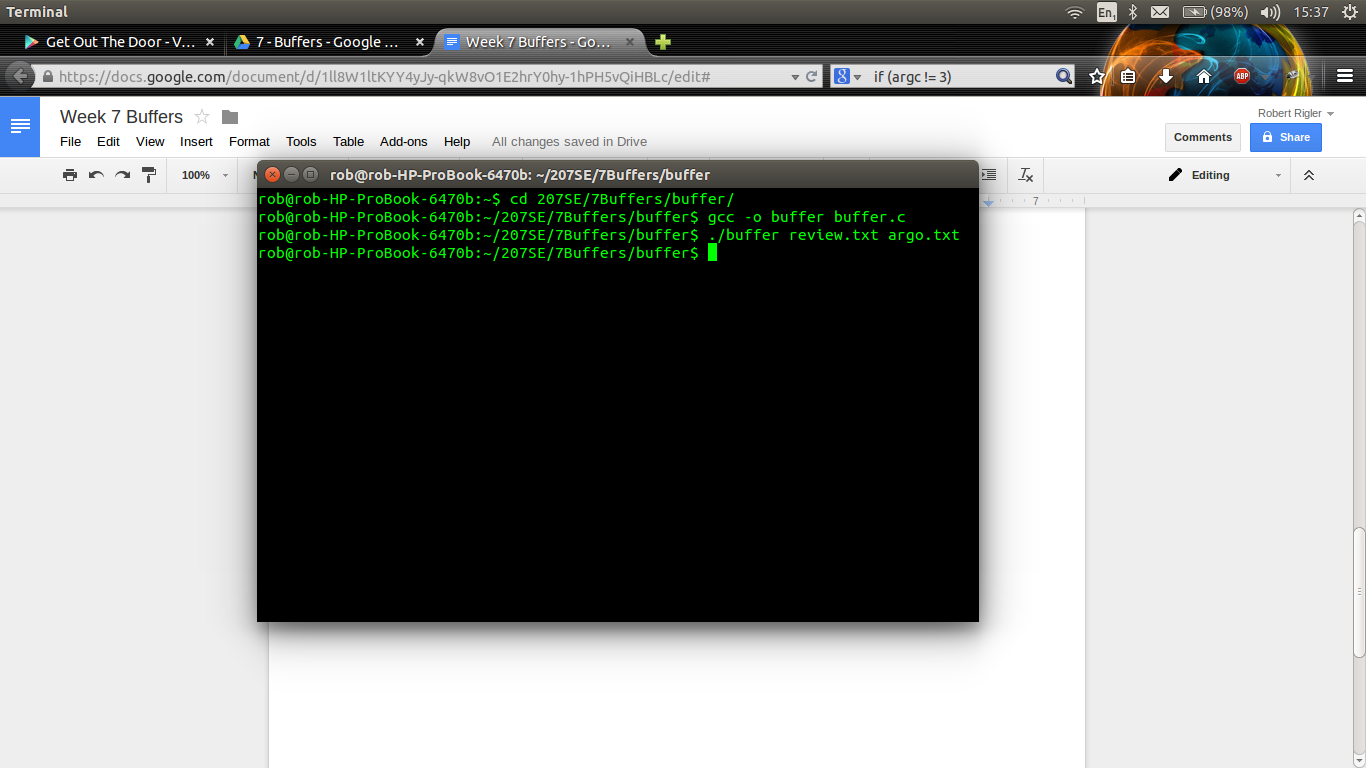
exit(5);

}

}

}

1. Compiled code



argo.txt contains the exact same text that was in review.txt .

1. Adapted code to show how many characters were read in total and how many times the buffer was filled. (Amended code underlined)

#include <fcntl.h>

#include <stdlib.h>

#include <unistd.h>

#include <stdio.h>

//Define header files

#define BUF\_SIZE 500 //Define Buffer size as 500.

#define OUTPUT\_MODE 0700 //Define file permission.

int main(int argc, char \*argv[])

{

int in\_fd, out\_fd;

**int buf\_count=0,rd\_count=0;**

int rd\_size = 1, wr\_size;

char buf[BUF\_SIZE]; // //Declare buffer.

if (argc != 3)

exit(1);

in\_fd = open(argv[1], O\_RDONLY); //Open input file.

if (in\_fd < 0)

exit(2);

out\_fd = creat(argv[2], OUTPUT\_MODE); //Create output file.

if (out\_fd < 0)

exit(3);

while (rd\_size > 0) {

rd\_size = read(in\_fd, buf, BUF\_SIZE); // Continuously read from input file into buffer.

**rd\_count+= rd\_size;**

**if(rd\_size > 0)**

**buf\_count +=1; //Counts the number of times the buffer is filled (only //if rd\_size is > 0**

exit(4);

wr\_size = write(out\_fd, buf, rd\_size); // Continuously write from buffer into output file.

if (wr\_size<=0){

close(in\_fd); //Close input file.

close(out\_fd); //Close output file

**printf("Number of characters read total: %d\n",rd\_count -1 ); //Prints how many Characters were read.**

**printf("Number of times the buffer was filled: %d\n",buf\_count);**

**//Prints how many times the buffer was filled**

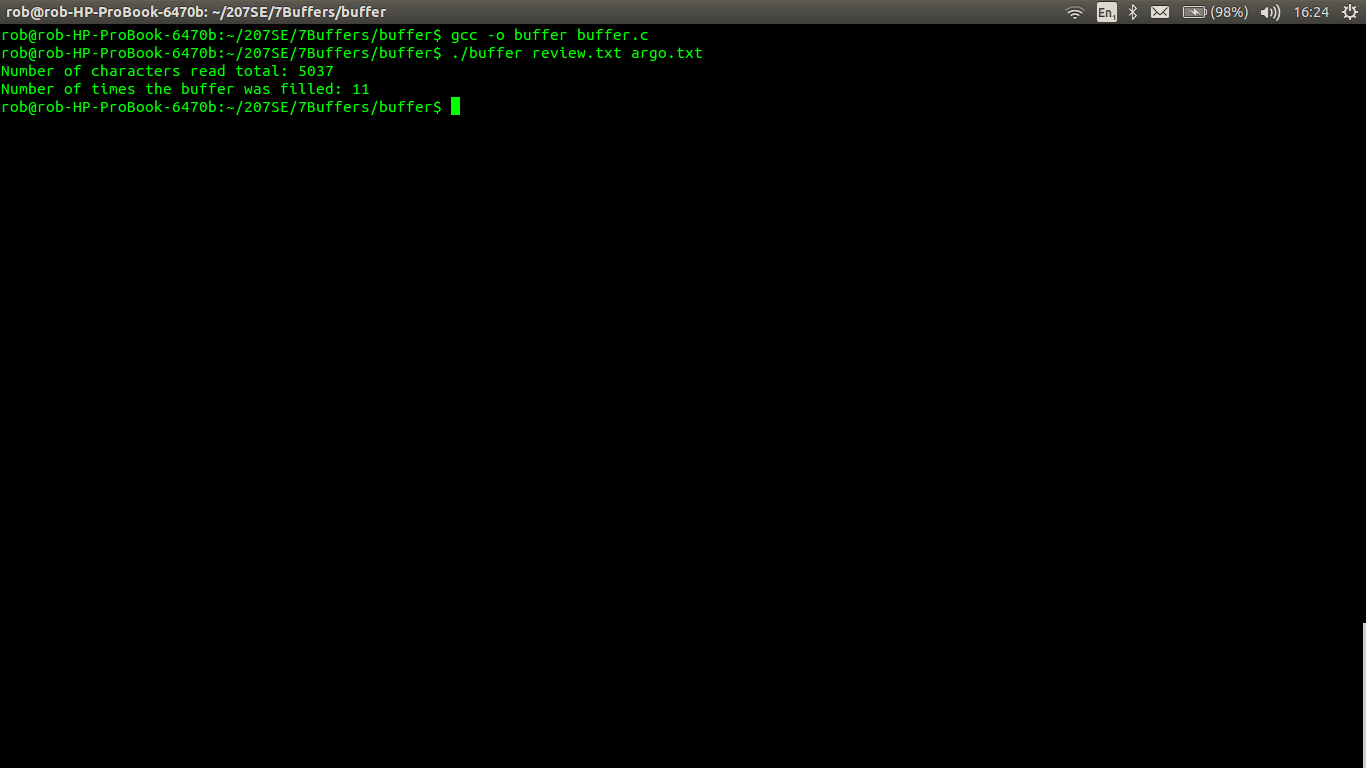
exit(5);

}

}

}

3a. Evidence



1. Alter the buffer size to 1000. How does this influence the number of times the buffer is filled.

* Doubling the buffer size to 1000, the program filled the buffer 6 times. This is half of the original value + 1.
* Doubling the buffer size again to 2000 , the program filled the buffer 3 times which is half of 6.
* Raising the the buffer size to 10000, the program filled the buffer 1 time, indicating that the entire text was placed into the buffer.

There is a direct linear correlation between the buffer size and the amount of times that the buffer was filled.

1. Adapt the code so that it is possible to compare if two files are the same.

5a. Sample Code

#include <fcntl.h>

#include <stdlib.h>

#include <unistd.h>

#include <stdio.h>

//Define header files

#define BUF\_SIZE 500 //Define Buffer size as 500.

#define OUTPUT\_MODE 0700 //Define file permission.

int main(int argc, char \*argv[])

{

int in\_fd, in0\_fd; // Create integers to hold file handles.

int rd\_size = 1; // Create integer to hold the amount of bytes in the //buffer.

char buf[BUF\_SIZE]; // //Declare 1st buffer.

char buf0[BUF\_SIZE]; //Declare 2nd buffer.

if (argc != 3)

exit(1);

in\_fd = open(argv[1], O\_RDONLY); //Open 1st file.

if (in\_fd < 0)

exit(2);

in0\_fd = open(argv[2], O\_RDONLY); //Open 2nd file.

if (in0\_fd < 0)

exit(3);

while (rd\_size > 0) {

int i;

rd\_size = read(in\_fd, buf, BUF\_SIZE); // Read From 1st file into 1st //buffer

if (rd\_size <0)

exit(4);

rd\_size = read(in0\_fd, buf0, rd\_size); //Read from 2nd file into 2nd //buffer

for (i =0; i < BUF\_SIZE; i++){ //Loop through the contents of each //buffer.

if (buf[i] == buf0[i]) // If buffer contents are equal, go to next buffer //element.

continue;

else { //If buffer contets are not the same, close the files and display //a message and exit the program.

close(in\_fd); //Close input file.

close(in0\_fd); //Close output file

printf("Files are not the same. \n");

exit(5);

} //end else

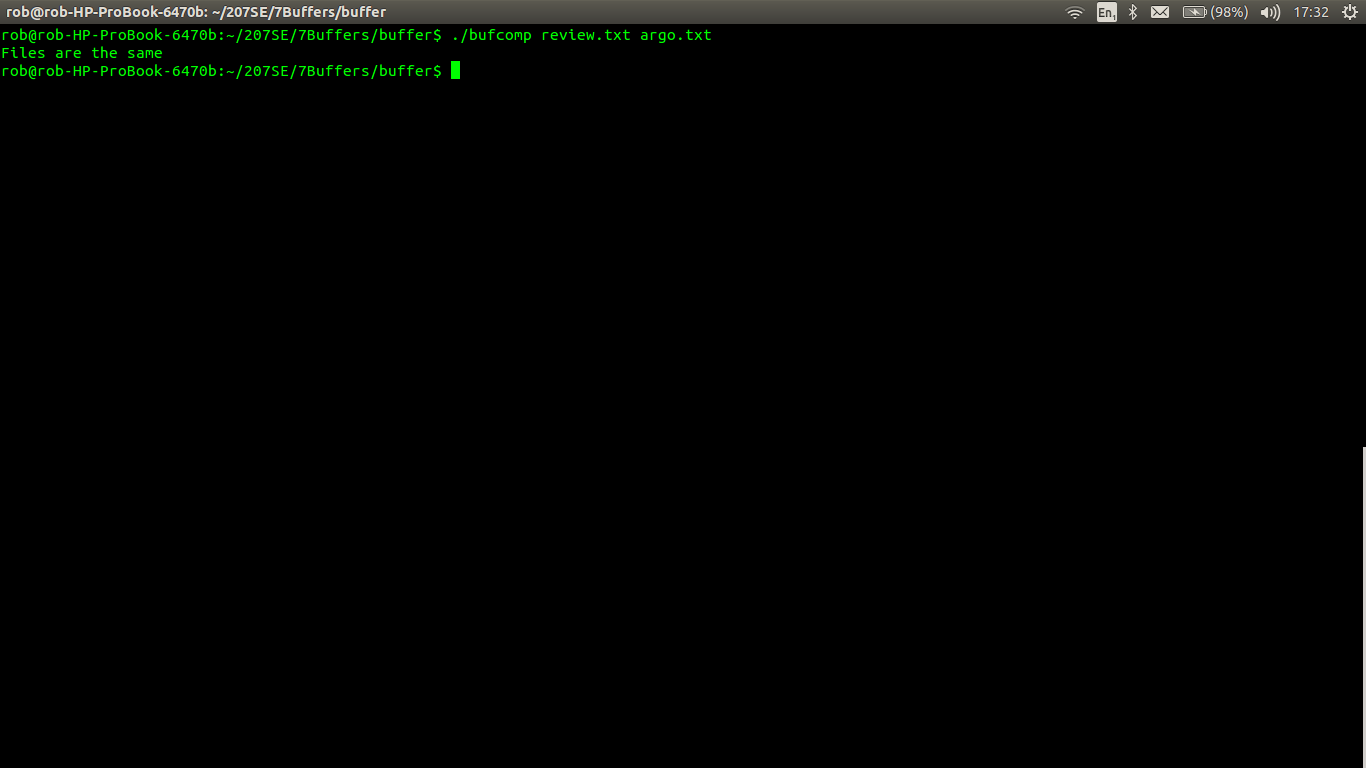
} //end for

} //end while

printf("Files are the same \n"); // Display this message if the files are the same

}//end main

5b. Comparison of review.txt and argo.txt



5c. Comparison of argo.txt and review\_observer.txt

