University of Southern
Maine

M. Fenton Libby
COS 350
4 March 2021

Program 2 Write-Up

Systems Programming

Summa

1- Program 2 .... **1**

Program 2

COS 350

On Honeybee, the following code was written:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <stdbool.h>
#include <getopt.h>

#define VERSION "1.0.0"

//option elements
static bool success_flag = false;
static bool verbose_flag = false;
static bool keep_flag = false;
static bool brief_flag = false;
static struct option long_options[] = {
    {"verbose", no_argument, 0, 'v'},
    {"brief",  no_argument, 0, 'b'},
    {"keep",   no_argument, 0, 'k'},
    {"version", no_argument, 0, 'V'}
};

int main(int argc, char **argv){
    FILE *input;

    //HANDLE OPTIONS&ARGS
    int c;
    int long_index = 0;

    while ((c = getopt_long(argc,argv,"vbkV::", long_options, &long_index)) != -1){
        switch(c){
            case 'v':
                verbose_flag = true;
                break;
            case 'b':
                brief_flag = true;
                verbose_flag = false;
                break;
            case 'k':
                keep_flag = true;
                break;
            case 'V':
                printf("z827 %s\n", VERSION);
                exit(0);
            case '?':
                fprintf(stderr, "Unknown option character %c\n", optopt);
                exit(1);
            default:
                continue;
        }
    }
```

```
//NON-OPTION ARGS
if (optind < argc){
    if ( ( input = fopen ( argv[optind], "rb" ) ) == NULL){
        fprintf(stderr, "The file \"%s\" could not be found.\n", argv[optind]);
        exit(1);
    }
    if (optind < argc-1){
        fprintf(stderr, "This program currently cannot compress multiple files in one
            go, ignoring additional args after \"%s\"\n", argv[optind]);
    }
} else {
    fprintf(stderr, "Need a file to (de)compress!\n");
    exit(1);
}

//END OPTIONS&ARGS HANDLING

//SET UP (DE)COMPRESSION ELEMENTS
unsigned int buff = 0;
unsigned char curr;
int bitc = 0;
int size = 0;
int procBuff = 0;
char fn[strlen(argv[optind])];

//FIND FILE SIZE
fseek(input,0L,SEEK_END);
size = ftell(input);
rewind(input);

strcpy(fn, argv[optind]);
if (strstr(fn,"z827") == NULL){
    //COMPRESS

    printf(brief_flag ? "" : "Compressing file...\n");

    printf(verbose_flag ? "Original file size: %d bytes\n" : "", size);

    fread(&curr, 1, 1, input);
    buff |= curr;
    bitc += 8;
    procBuff ++;
    FILE *output = fopen(strcat(argv[optind],".z827"), "w+b");

    fwrite(&size, 4, 1, output);

    while ( fread(&curr, 1, 1, input) == 1){
        procBuff++;
        buff |= curr << bitc-1;
        bitc += 7;

        while (bitc > 8){
            fwrite(&buff, 1, bitc/8, output);
            buff >>= 8*(bitc/8);
            bitc -= 8*(bitc/8);

        }

    }
```

```
        //CLEANUP
        fwrite(&buff, 1, 1,output);
        fclose(input);
        fclose(output);
        if (procBuff == size){
            printf(verbose_flag ? "Compression Successful!\nCleaning up...\n" : "");
            success_flag = true;
            if(!keep_flag)
                unlink(fn);
        }

    } else {
        //DECOMPRESS

        printf(brief_flag ? "" : "Decompressing file...\n");

        printf(verbose_flag ? "Original file size: %d bytes\n" : "", size);


        //ignore first 4 bits of file, which is compression info
        fseek(input, 4, SEEK_SET);
        procBuff+=4;
        //remove .z827 extension from input file string copy
        fn[strlen(fn)-5]='\0';

        FILE *output = fopen(fn, "w+b");

        while ( fread(&curr, 1, 1, input) == 1){
            buff |= curr << bitc; //append next char to the buffer
            bitc += 8;//add how many bits were just added to the buffer
            procBuff ++;

            //there are more than 7 bits left in the buffer, keep writing to the file
                until this is not so:
            while(bitc>7){
                curr = buff & 0x7f;
                fwrite(&curr, 1, 1, output);
                buff >>= 7;
                bitc -= 7;

            }

        }
        //CLEANUP
        fclose(input);
        fclose(output);
        if (procBuff == size){
            printf(verbose_flag ? "Decompression Successful!\nCleaning up...\n" : "");
            success_flag = true;
            if (!keep_flag)
                unlink(argv[optind]);
        }
    }

    printf(success_flag & !brief_flag ? "Done!\n" : "Compression Failed, make sure input
        file is ASCII compliant.\n");
    return 0;
}
```

I chose to move beyond the basic requirements of the program in order to further explore the implementation of options and structures in C. I implemented an option to keep the input file after successful (de)compression, two options to modify printing to the console during the (de)compression, and an option to simply display the version of the program. These options are discussed in the manual page.

There are no known bugs for this program, but an obvious "next step" would be allowing for the input of multiple files for sequential file compression, and input of a folder name to create a zipped folder of sorts.

Finally, the man page - which is included as both a separate PDF and DOC file, was modeled after the man page for the GZIP command. Since I missed the initial deadline for this submission and had some extra time, I drew up a LaTeXclass file for the easy creation of an aesthetically pleasing man page. The text from this can (and was) be copied and pasted into a text file, maintaining most of the formatting. I think next I will look into creating a program which takes the info from a PDF manual page and converts it to the proper format and moves it to the proper folder in **/usr/share/man**.

A log of the required testing results:

```
Script started on 2021-03-06 18:19:21-0500
mflibby@honeybee:~/cos350/prog/prog2$ cp short.txt short.orig
mflibby@honeybee:~/cos350/prog/prog2$ cp shake.txt shake.orig
mflibby@honeybee:~/cos350/prog/prog2$ z827 short.txt
Compressing file...
Done!
mflibby@honeybee:~/cos350/prog/prog2$ z827 shake.txt
Compressing file...
Done!
mflibby@honeybee:~/cos350/prog/prog2$ ls -l
total 9824
drwx------ 2 mflibby mflibby 4096 Mar 6 18:18 fol
-rw------- 1 mflibby mflibby    0 Mar  6 18:19 results.scr
-rwx------ 1 mflibby mflibby 5344213 Mar 6 18:19 shake.orig
-rw------- 1 mflibby mflibby 4676191 Mar 6 18:19 shake.txt.z827
-rwx------ 1 mflibby mflibby   67 Mar  6 18:19 short.orig
-rw------- 1 mflibby mflibby   63 Mar  6 18:19 short.txt.z827
-rwx------ 1 mflibby mflibby 13504 Mar 6 18:09 z827
-rw------- 1 mflibby mflibby 3797 Mar 6 18:09 z827.c
-rw------- 1 mflibby mflibby 3820 Mar 6 18:08 z827.c~
mflibby@honeybee:~/cos350/prog/prog2$ od -t x1 short.txt.z827
0000000 43 00 00 00 41 d0 3c dd 86 b3 cb 20 7a 19 4f 07
0000020 99 d3 ec 32 88 fe 06 89 cb a0 fa bc 4c 06 a5 dd
0000040 20 7a 79 4e 4f bb cf a0 77 9d 0e a2 a3 cb 20 3d
0000060 4e 76 03 8d df 6d b8 bc 3c 9f a7 df 6e 97 02
0000077
mflibby@honeybee:~/cos350/prog/prog2$ z827 short.txt.z827
Decompressing file...
Done!
mflibby@honeybee:~/cos350/prog/prog2$ z827 shake.txt.z827
Decompressing file...
Done!
mflibby@honeybee:~/cos350/prog/prog2$ ls -l
total 10476
drwx------ 2 mflibby mflibby 4096 Mar 6 18:18 fol
-rw------- 1 mflibby mflibby    0 Mar  6 18:19 results.scr
-rwx------ 1 mflibby mflibby 5344213 Mar 6 18:19 shake.orig
-rw------- 1 mflibby mflibby 5344213 Mar 6 18:20 shake.txt
-rwx------ 1 mflibby mflibby   67 Mar  6 18:19 short.orig
-rw------- 1 mflibby mflibby   67 Mar  6 18:20 short.txt
-rwx------ 1 mflibby mflibby 13504 Mar 6 18:09 z827
-rw------- 1 mflibby mflibby 3797 Mar 6 18:09 z827.c
-rw------- 1 mflibby mflibby 3820 Mar 6 18:08 z827.c~
mflibby@honeybee:~/cos350/prog/prog2$ cmp short.txt short.orig
mflibby@honeybee:~/cos350/prog/prog2$ cmp shake.txt shake.orig
mflibby@honeybee:~/cos350/prog/prog2$ z827 definitelyarealfile
The file "definitelyarealfile" could not be found.
mflibby@honeybee:~/cos350/prog/prog2$ z827
Need a file to (de)compress!
mflibby@honeybee:~/cos350/prog/prog2$ exit
exit

Script done on 2021-03-06 18:22:53-0500
```