60-330

Assignment 1

Dr. Ngom

I confirm that I will keep the content of this assignment confidential. I confirm that I have not received any unauthorized assistance in preparing for or writing this assignment. I acknowledge that a mark of 0 may be assigned for copied work.

Jason Choquette, 104337378

For this project I used ANSI C system calls; fopen(), fclose() and the FILE structure for all file operations.

Ex:

FILE \* getFilePointer(const char \* fileName, int fileType)

{

char buffer[MAX\_BUFFER];

FILE\* fp = NULL;

// format current file location

strncpy\_s(buffer, MAX\_BUFFER, "../", 3);

strcat\_s(buffer, MAX\_BUFFER, fileName);

// for input file type

if (fileType == INPUT)

{

// if input file doesn't exist, abort program...

if (!(fp = fopen(buffer, "r")))

{

printf("Source file does not exist. Aborting...");

fp = NULL;

}

}

// for output file type

else if (fileType == OUTPUT)

{

// if the output file exists, abort program...

if (fp = fopen(buffer, "r"))

{

printf("Destination file already exists. Aborting...");

fp = NULL;

}

else

fp = fopen(buffer, "w");

}

return fp;

}

For user input I use fgets().

Ex:

char \* getFileName(char \* fileType)

{

char buffer[MAX\_BUFFER];

// prompt the user to enter a file name

printf("Please enter the name of the %s file: \n", fileType);

char \* fileName = fgets(buffer, sizeof(buffer), stdin);

// ensure user does not enter an empty string

while (fileName == NULL || strcmp(fileName, "") == 0)

{

fileName = fgets(buffer, sizeof(buffer), stdin);

printf("Invalid filename. Please ensure you entered correct filename. /n");

}

// return file name without newline character

return strtok(buffer, "\n");

}

To copy the contents of the input file, I used a simple while loop and read in one character at a time until the End-Of-File was reached.

void copyContents(FILE \*\* in, FILE \*\* out)

{

char c = fgetc(\*in);

while (c != EOF)

{

fputc(c, \*out);

c = fgetc(\*in);

}

}

The main function contains the run program which is the logic of the copy file program:

int main() { return run(); }

int run()

{

// input and output filenames

char \* fileName;

// pointer to reference the file names

char \* f;

// File pointers for input/output files

FILE \* fpInput = NULL;

FILE \* fpOutput = NULL;

// allocate memory

if (((f = (char \*)malloc(MAX\_BUFFER)) == NULL))

{

printf("Error allocating memory. Aborting.../n");

return -1;

}

// get input file name

fileName = getFileName("input");

// copy input file name

strncpy\_s(f,256, fileName, MAX\_BUFFER);

// get input file pointer

if (!(fpInput = getFilePointer(f, INPUT))) return -1;

// reset/clear memory

memset(f, 0, MAX\_BUFFER);

// get output file name

fileName = getFileName("output");

// copy output file name

strncpy\_s(f, 256, fileName, MAX\_BUFFER);

// get input file pointer

if (!(fpOutput = getFilePointer(f, OUTPUT))) return -1;

// release memory pointer

free(f);

// copy contents from input file to output file

copyContents(&fpInput, &fpOutput);

// release file pointers

fclose(fpInput);

fclose(fpOutput);

printf("File copied successfully.\n");

return 0;

}