ForNextDay(9)

Stephen Cole

3553803

9a

printargs.c

#include<stdio.h>

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

{

int i;

printf("Number Of Arguments Passed: %d\n",argc);

for(i=0;i<argc;i++)

printf("argv[%d]: %p\n", i, &argv[i]);

return 0;

}

A screenshot of a cell phone

Description automatically generated

9b

Using the project’s Makefile, explain the purpose of:

* $@ symbol
* $(CFLAGS)

$@ is used to define the name of the executable mystring

$(CFLAGS) is used to define the intended flags for the compilation “-g -Wall -Wshadow”

Makefile

GCC = gcc

CFLAGS = -g -Wall -Wshadow

OBJS = mystring.o main.o

HDRS = mystring.h

VAL = valgrind --tool=memcheck --leak-check=full

VAL += --verbose --log-file=

%.o: %.c

$(GCC) $(CFLAGS) -c $\*.c

mystring: $(OBJS) $(HDRS)

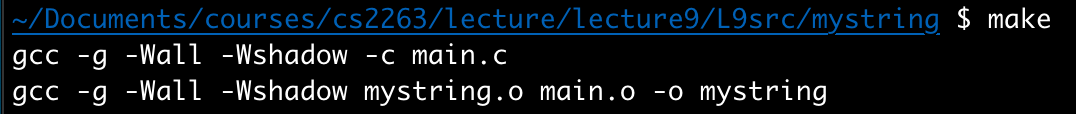
$(GCC) $(CFLAGS) $(OBJS) -o $@

clean:

rm -f mystring $(OBJS)

A picture containing food

Description automatically generated



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

if (argc != 4) {

printf("usage: %s command input output\n", argv[0]);

return EXIT\_FAILURE;

}

FILE \*infptr = fopen(argv[2], "r");

if (infptr == NULL) {

printf("unable to open file %s!\n", argv[2]);

return EXIT\_FAILURE;

}

FILE \*outfptr = fopen(argv[3], "w");

if (outfptr == NULL) {

printf("unable to open file %s!\n", argv[3]);

fclose(infptr);

return EXIT\_FAILURE;

}

int num\_lines = 0;

char buffer[LINE\_SIZE];

// count the number of lines in the file

while (fgets(buffer, LINE\_SIZE, infptr) != NULL)

num\_lines++;

// return to the beginning of the file

fseek(infptr, 0, SEEK\_SET);

char \*\*lines = malloc(sizeof(char \*) \* num\_lines);

int i;

for (i = 0; i < num\_lines; i++) {

if (feof(infptr)){

printf("not enough num\_lines in file!\n");

fclose(infptr);

fclose(outfptr);

return EXIT\_FAILURE;

}

lines[i] = malloc(sizeof(char) \* LINE\_SIZE);

fgets(lines[i], LINE\_SIZE, infptr);

}

fclose(infptr);

int total\_length = 0;

for (i = 0; i < num\_lines; i++)

total\_length += my\_strlen(lines[i]);

// count the length of each line

for (i = 0; i < num\_lines; i++){

fprintf(outfptr, "length: %d\n",

my\_strlen(lines[i]));

}

/\* for each line, count the occurrence of the first

letter in the line \*/

for (i = 0; i < num\_lines; i++){

fprintf(outfptr, "count(%c): %d\n", lines[i][0],

my\_countchar(lines[i], lines[i][0]));

}

for (i = 0; i < num\_lines; i++) {

my\_strupper(lines[i]);

fprintf(outfptr, "%s", lines[i]);

}

for (i = 0; i < num\_lines; i++){

char\* ret = my\_strchr(lines[i], 'a');

fprintf(outfptr, "first a located at: %s\n", ret);

}

for (i = 0; i < num\_lines; i++)

free(lines[i]);

free(lines);

fclose(outfptr);

return EXIT\_SUCCESS;

}