

Assignment 1 – Command Line Arguments

Description:

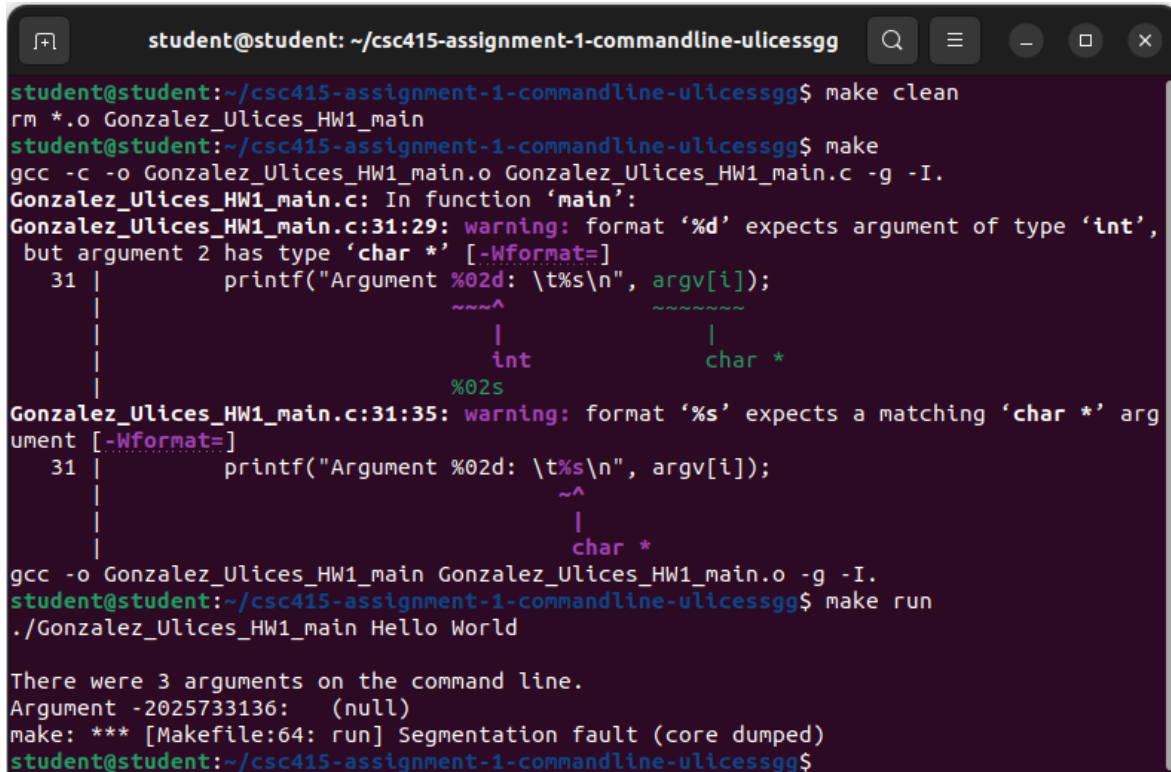
This assignment handles the fetching of command line arguments and displays them to the user with their correct index value based on their position in the command line.

Approach:

In this assignment, `int main` accepts the integer `argc` and the character pointer array `argv` from the command line. The `argc` integer holds the number of command line arguments and the character pointer array `argv` holds each argument entered on the command line. The program uses a `for` loop set to begin at the index of 0 which iterates until the set limit of `argc` is reached. The program is formatted through the use of newlines and tabs for lines of text, and format specifiers `02` for integers.

Issues and Resolutions:

My first and only issue was a segmentation fault which occurred due to the iterator in my `for` loop being left out of the `printf` statement.



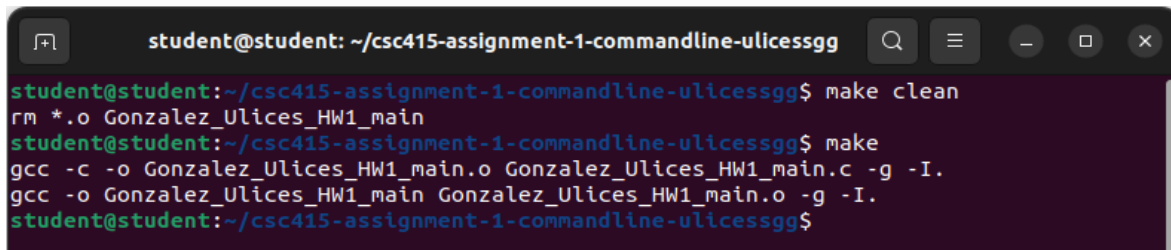
```
student@student: ~/csc415-assignment-1-commandline-ulicesgg
student@student:~/csc415-assignment-1-commandline-ulicesgg$ make clean
rm *.o Gonzalez_Ulices_HW1_main
student@student:~/csc415-assignment-1-commandline-ulicesgg$ make
gcc -c -o Gonzalez_Ulices_HW1_main.o Gonzalez_Ulices_HW1_main.c -g -I.
Gonzalez_Ulices_HW1_main.c: In function 'main':
Gonzalez_Ulices_HW1_main.c:31:29: warning: format '%d' expects argument of type 'int',
but argument 2 has type 'char *' [-Wformat=]
   31 |         printf("Argument %02d: \t%s\n", argv[i]);
      |                               ^
      |                               |
      |                           int      char *
      |                           %02s
Gonzalez_Ulices_HW1_main.c:31:35: warning: format '%s' expects a matching 'char *' arg
ument [-Wformat=]
   31 |         printf("Argument %02d: \t%s\n", argv[i]);
      |                               ^
      |                               |
      |                           char *
gcc -o Gonzalez_Ulices_HW1_main Gonzalez_Ulices_HW1_main.o -g -I.
student@student:~/csc415-assignment-1-commandline-ulicesgg$ make run
./Gonzalez_Ulices_HW1_main Hello World

There were 3 arguments on the command line.
Argument -2025733136: (null)
make: *** [Makefile:64: run] Segmentation fault (core dumped)
student@student:~/csc415-assignment-1-commandline-ulicesgg$
```

This was solved by simply changing `printf("Argument %02d: \t%s\n", argv[i]);` to `printf("Argument %02d: \t%s\n", i, argv[i]);`

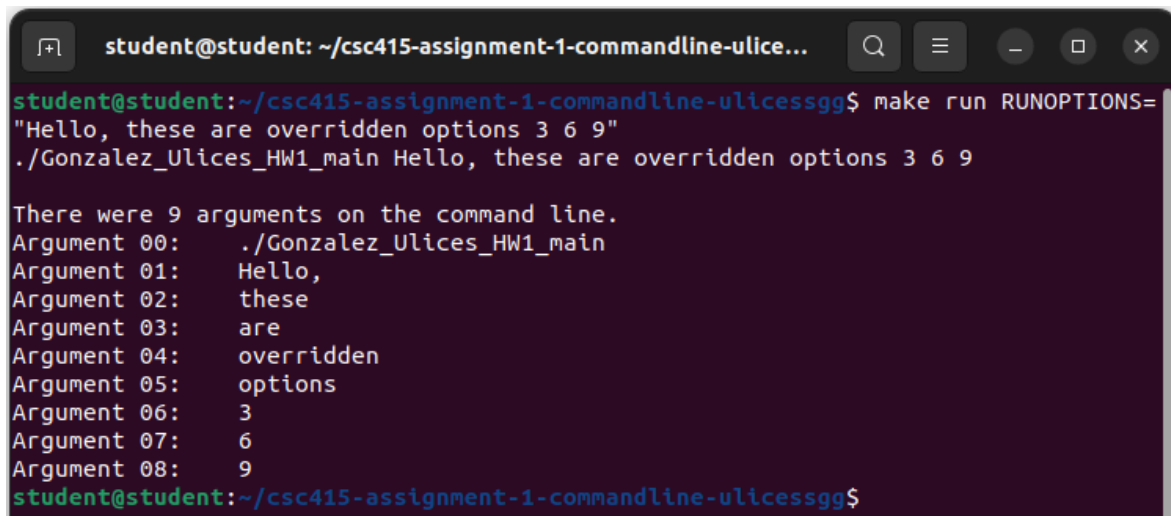
Analysis: (Not required for this assignment)

Screen shot of compilation:

A terminal window with a dark background and light-colored text. The window title is "student@student: ~/csc415-assignment-1-commandline-ulicessgg". The terminal shows the following commands and output:

```
student@student:~/csc415-assignment-1-commandline-ulicessgg$ make clean
rm *.o Gonzalez_Ulices_HW1_main
student@student:~/csc415-assignment-1-commandline-ulicessgg$ make
gcc -c -o Gonzalez_Ulices_HW1_main.o Gonzalez_Ulices_HW1_main.c -g -I.
gcc -o Gonzalez_Ulices_HW1_main Gonzalez_Ulices_HW1_main.o -g -I.
student@student:~/csc415-assignment-1-commandline-ulicessgg$
```

Screen shot(s) of the execution of the program:

A terminal window with a dark background and light-colored text. The window title is "student@student: ~/csc415-assignment-1-commandline-ulice...". The terminal shows the following commands and output:

```
student@student:~/csc415-assignment-1-commandline-ulicessgg$ make run RUNOPTIONS=
"Hello, these are overridden options 3 6 9"
./Gonzalez_Ulices_HW1_main Hello, these are overridden options 3 6 9

There were 9 arguments on the command line.
Argument 00:  ./Gonzalez_Ulices_HW1_main
Argument 01:  Hello,
Argument 02:  these
Argument 03:  are
Argument 04:  overridden
Argument 05:  options
Argument 06:  3
Argument 07:  6
Argument 08:  9
student@student:~/csc415-assignment-1-commandline-ulicessgg$
```