

Israel Alcantara

Homework 5

Problem 1

For this assignment I wrote a program that examines the memory storing the command line arguments by converting the variable `argv` into an unsigned char pointer so I could inspect its bytes one by one. The program prints the memory content of `argv` in eight byte groups in hexadecimal format and also prints the address where `argv` is stored. Then it iterates over each element in the `argv` array to print the pointer stored in each element as eight bytes and shows their respective addresses. After that I adjusted the pointer to an address that is aligned to an eight byte boundary and determined the finish point by including the null terminator and aligning that address to eight bytes. The program finally prints each eight byte block in hexadecimal and prints the corresponding ASCII characters while displaying a dot for any non printable characters.

Problem 2

In this assignment I wrote a program that uses `getopt` to parse the command line options in a way similar to the example shown in class. The program accepts options `c`, `d` and `u` as well as options `f` and `s` that require an additional argument. It prints the option when it is encountered and if the option requires an argument it prints that as well. The program checks for unknown options and missing arguments and prints an error message if it finds any problems. After processing the options the remaining arguments are treated as the input file and the output file and they are printed accordingly.

Problem 3

I developed a program that sorts the environment variables based on their names by directly using the `envp` array provided to `main`, the program first counts the number of environment variables and then implements a bubble sort algorithm to order them, it compares adjacent pairs by locating the equal sign in each variable and temporarily replacing it with a null character so that only the name is compared using a string comparison function, if the first variable name is greater than the second the program swaps the pointers in the array and then restores the equal sign, after sorting the program prints each environment variable in order.