

## Strings and Command Line Arguments

11-15-2002

---

---

---

---

---

---

---

---

## Opening Discussion

- What did we talk about last class?
- Do you have questions about the assignment?

---

---

---

---

---

---

---

---

## Questions on Minute Essays

- How to initialize and array of structures.
- What is "->"?
- Why use structs? When do you take their size?
- Dynamic memory inside structures.
- When to use \*\* instead of \*.
- When would you not use dynamic memory?

---

---

---

---

---

---

---

---

### More Minute Essay Questions

- How can we just “get more memory” and what role does free play?
- I don’t play any instruments.
- When do you use arrays vs. structs?
- There was also a question about using x+1 instead of x++ in the context of the last class that I didn’t understand.

---

---

---

---

---

---

---

---

### The string.h Library

- If you type in “man string” you can get a listing of all the functions that are part of the string.h library.
- That is the standard C library for dealing with strings, or null terminated arrays of characters.
- This library contains many helpful functions. You could write any of them yourself, but they save you from having to.

---

---

---

---

---

---

---

---

### Basic Functions

- strcmp(char \*s1,char \*s2) - compares two strings. It returns 0 if they are equal, <0 if the first is less and >0 if the second is less.
- strlen(char \*s) - returns the length of the string.
- strcpy(char \*dest,char \*src) - copies a string.
- strcat(char \*dest,char \*src) - concatenates.

---

---

---

---

---

---

---

---

## Other Functions

- `strcascmp` - case insensitive comparison.
- `strchr` - finds a character in a string.
- `strtok` - finds a token in a string.
- `strfry` - makes a random anagram of the string.
- `strncmp` - compare the first n characters of strings.
- `strncpy` - copy the first n character of a string.

---

---

---

---

---

---

---

## Command Line Arguments

- When you are working in Linux, you almost always type not just a command, but some type of arguments for that command.
- We'd like to be able to do this in our programs so that we can give our program extra information about how to run when we type the command.
- Example "a.out 3+5\*8"

---

---

---

---

---

---

---

## Arguments for main

- The way we get command line arguments is by providing special arguments for `main`. With the following signature, the `int` is the number of arguments, including the command, and the `char**` is an array of strings for those arguments.

```
int main(int argc, char **argv);
```

---

---

---

---

---

---

---

## Code using Command Line Arguments

- Now let's write some code that uses command line arguments. Since the arguments always come as strings we can use some parts of the string library as well.

---

---

---

---

---

---

---

---

## Minute Essay

- How are you coming on the assignment? Command line arguments are something that is rarely used in the windows environment. Do you see their value when working on a computer?
- Next class is sorting.

---

---

---

---

---

---

---

---