161-01

Reading and Writing Files

Friday November 1

Agenda

0. Quiz Feedback
1. Announcements
2. Command-line arguments (Wednesday’s lab)
3. Discussion
4. Lab

Quiz Feedback

Declaration of a string variable may omit the length.

```
char s[] = “flamingo”;
```

`s` is an array of characters. It is not assignable. Instead access individual elements by indexing `s[0], s[1]…`

```
char s[10];
s = “flamingos”;
```

Declares `s` to be a pointer.

```
char *s = “flamingo”;
```

Not enough room for the terminating null character. Will not work as desired with the string library.

```
char s[4] = “flamingos”;
```

Quiz Feedback

A *rule* in a Makefile has the following form

```
target:  prerequisites …
        recipe
        …
        …
```

A rule to clean the directory:

```
clean:
        rm –f exp1 exp2
```

Announcements

- Homework 6 is due on Monday
  - An assignment to encourage you to think about what is going well, and what changes you might consider making.
  - Any questions?
- Lab Report 8 due on Monday
- Token Events
  - Choose your own game show, tonight at 7pm in Harris. (Adarsh)
  - VSA event with food, Saturday 5-7pm in HSSC Atrium (Linh)
  - Other things?

Wednesday's Lab

Command-line arguments

To access command-line arguments we define main as a function with two parameters, conventionally named `argc` and `argv`

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

   ...

}
```

`argc` is the "argument count" – the number of command-line arguments, including the name of the program itself.

`argv` is the "argument vector" – an array of pointers to the command line arguments, each of which is stored as a string.

Wednesday’s Lab

Command-line arguments

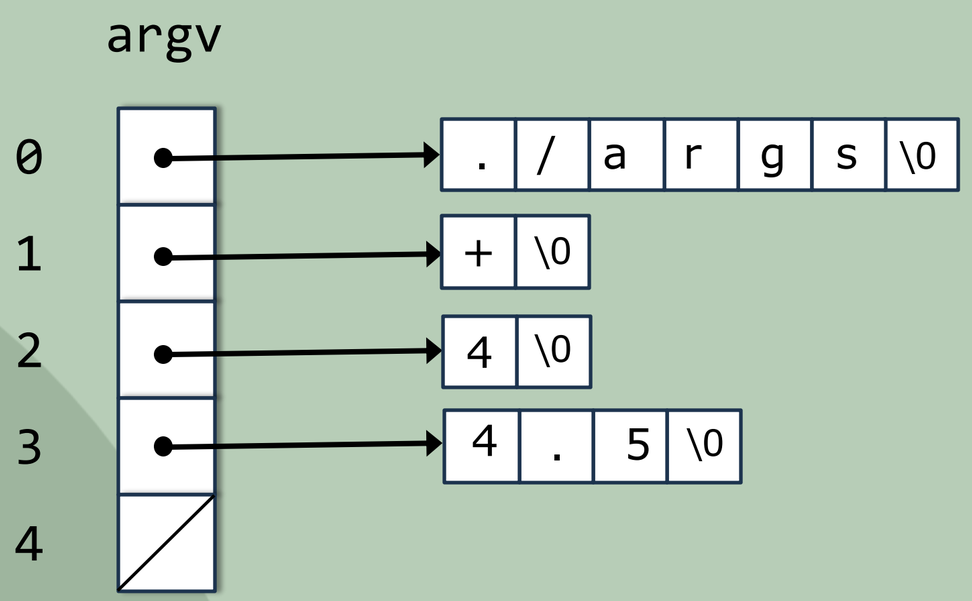
Notice: We’ve already been using command-line arguments!

```
$ cd csc161
```

Wednesday’s Lab

Command-line arguments

```
$./args + 4 4.5
```



Wednesday’s Lab

Command-line arguments

```
int sentenceLen = strlen(argv[1]);
int targetLen = strlen(argv[2]);


for(int i = 0; i < sentenceLen – targetLen + 1; i++){
    /* compare argv[1][i] – argv[1][i+targetLen-1]
     * to
     * argv[2][0] – argv[2][targetLen-1]
    */

}
```

Discussion

Reading and Writing Files

Discussion

Reading and Writing Files

1. What does it mean to *open* or *close* a file?
   What functions open/close files?

2. What does it mean to *read* or *write* a file?
   What are (some) functions to read/write?

3. Why do you think a program might use a file?

Questions?

Lab

Wrap up

For next time:

- Read King 16.1 – 16.3
- Lab Report 8 due Monday
- Homework 6 due Monday