

Lab 6

Command Line Arguments

October 13th, 2010

James Marshall

These are Your labs

- Bring questions
- Email me topics before hand

Bash Tips

- Useful commands
 - `history` – lists previous commands
 - `grep "string" file` – searches file for string
- Pipes
 - `|`
 - One command's output as input to another
 - `history | grep "gcc"`

Recursion Review

- Fibonacci Sequence
- 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
- n^{th} number?
 - Base case:
 - $n = 0, 1$: return 1
 - Recursive case:
 - Otherwise return $\text{fib}(n-1) + \text{fib}(n-2)$

The code:

```
#include "stdio.h"

int fib(int n) {
    if (n == 0 || n == 1) {
        return 1;
    } else {
        return fib(n-1) + fib(n-2);
    }
}

int main() {
    printf("10th fib number: %d\n", fib(10));
}
```

Motivation

- Trees
 - Not just Data
 - Decision trees / AI
 - Nodes describe state, edges describe transitions
- Command line arguments
 - Write useful programs

Useful Programs!

- You can
 - read / write files
 - Take command line args
 - Read from stdin
 - Ask user for input
 - Create small programs that do one thing well
 - Link programs via file redirection and pipes and scripts

That's how linux works!

- Forget the GUI, this is the real power of linux
- You are now power users
- If there isn't a tool for what you need, just write one
- And share it!

Command Line Arguments

- `int main(int argc, char * argv[]) {`
- `int argc`
 - **argument count**
 - Includes program name
- `char * argv[]`
 - **argument values**
 - Array of `char *`
 - 0th is name of program
 - `argc`th is NULL

Update Fib

```
int main(int argc, char * argv[]) {
    int n = 0;

    if (argc < 2) {
        printf("Please specify desired fib number\n");
        return 1;
    }

    sscanf(argv[1], "%d", &n);
    printf("%d fib number: %d\n", n, fib(n));
    return 0;
}
```

Please Note

- We only changed the `main` function
 - Modularity!
- Check `argc` before accessing `argv`
 - C won't check for you
 - `return 1; return 0;`
- Could be more robust
 - `sscanf`

Not Needed for Project

- What if you want to process from a pipe?
 - Read from stdin (scanf)

```
int main(int argc, char * argv[]) {
    int n = 0, res;
    while(1) {
        res = scanf("%d", &n);
        if (res == 0 || res == EOF) { return 0; }
        printf("%d fib number: %d\n", n, fib(n));
    }
}
```

Needed for Project

- Get file names from command line
- Open files
- Read data from messages file
 - Store? Struct?
- Read commands from query file

Read Message Data

FROM: ayoussef (<= 50 chars)

DATE: 09-15-2005 (MM-DD-YYYY)

SUBJECT: Do your homework (<= 60 chars)

BODY:

My advice to you is to start to do your programming assignment 1 today. -AY

(Always 10 '#'s)

Message Struct

- The operations performed hint at data structure
 - Need a way to store messages
 - And order them by date
 - Remove most recent
- The format of the message gives hints as well

Parse?

- How do we get data from the text file?
 - fscanf – used previously
 - strtok – next week
 - <http://www.opengroup.org/onlinepubs/000095399/fun>