

APS 105

Winter 2012

Jonathan Deber

jdeber -at- cs -dot- toronto -dot- edu

Lecture 2
January 18, 2012

Today

- (Quick) Admin
- Instructions
- C Basics

Admin

- Blackboard
- MarkUs
- A0
- Office hours

Office Hours

- 2 hours per week
- Possible times:
 - Mondays 12:00 - 1:00
 - Mondays 1:00 - 2:00
 - Wednesdays 12:00 - 1:00
 - Wednesdays 1:00 - 2:00
 - Something else?
- Location TBA

Instructions

- Also called algorithms
- We use them all the time

Beat two egg whites until they form stiff peaks, then gently fold into batter. Transfer to pan, and smooth top. Bake in a 450°F oven until golden brown.

Apply to hair. Lather, rinse, repeat.

Instructions

Add the numbers 2 and 3.
Print out the result.

Programming
(human)

```
int sum = 2 + 3;  
printf("%d", sum);
```

Compiling
(computer)

Machine Code

```
11111110111011011111101011001111  
0000000100000000000000000000111  
1000000000000000000000000000011  
0000000000000000000000000000010  
00000000000000000000000000001011  
000000000000000000000001100000000  
00000000000000000000000010000101  
00000000000000000000000000000000  
000000000000000000000000000011001  
00000000000000000000000001001000  
01000001010100000101111101011111
```

Programming Process

- Form the problem as an algorithm
 - Translate to a programming language
 - Translate to machine code
 - Part art, part science
- } you
- compiler

Languages

- Lots and lots (and lots) of them
- We're using C
- Different languages fall on different places in a continuum

Instructions

Add the numbers 2 and 3.
Print out the result.

Programming
(human)

```
int sum = 2 + 3;  
printf("%d", sum);
```

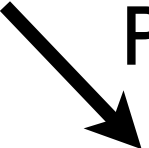
Compiling
(computer)

Machine Code

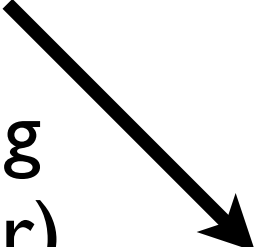
```
11111110111011011111101011001111  
0000000100000000000000000000111  
1000000000000000000000000000011  
0000000000000000000000000000010  
000000000000000000000000000001011  
0000000000000000000000001100000000  
000000000000000000000000010000101  
00000000000000000000000000000000  
0000000000000000000000000000011001  
00000000000000000000000001001000
```

Instructions

Add the numbers 2 and 3.
Print out the result.

 Programming
(human)

```
int sum = 2 + 3;  
printf("%d", sum);
```

 Compiling
(computer)

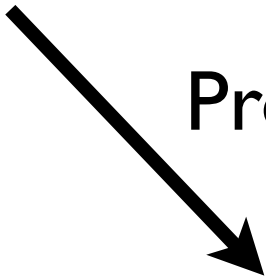
Machine Code

```
11111110111011011111101011001111  
0000000100000000000000000000111  
1000000000000000000000000000011  
0000000000000000000000000000010  
000000000000000000000000000001011  
0000000000000000000000001100000000  
000000000000000000000000010000101  
000000000000000000000000000000000  
0000000000000000000000000000011001  
00000000000000000000000001001000
```

Instructions

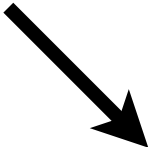
Add the numbers 2 and 3.
Print out the result.

Programming
(human)



```
int sum = 2 + 3;  
printf("%d", sum);
```

Compiling
(computer)



Machine Code

```
11111110111011011111101011001111  
0000000100000000000000000000111  
10000000000000000000000000000011  
00000000000000000000000000000010  
0000000000000000000000000000001011  
0000000000000000000000001100000000  
000000000000000000000000010000101  
0000000000000000000000000000000000  
00000000000000000000000000000011001  
00000000000000000000000001001000
```

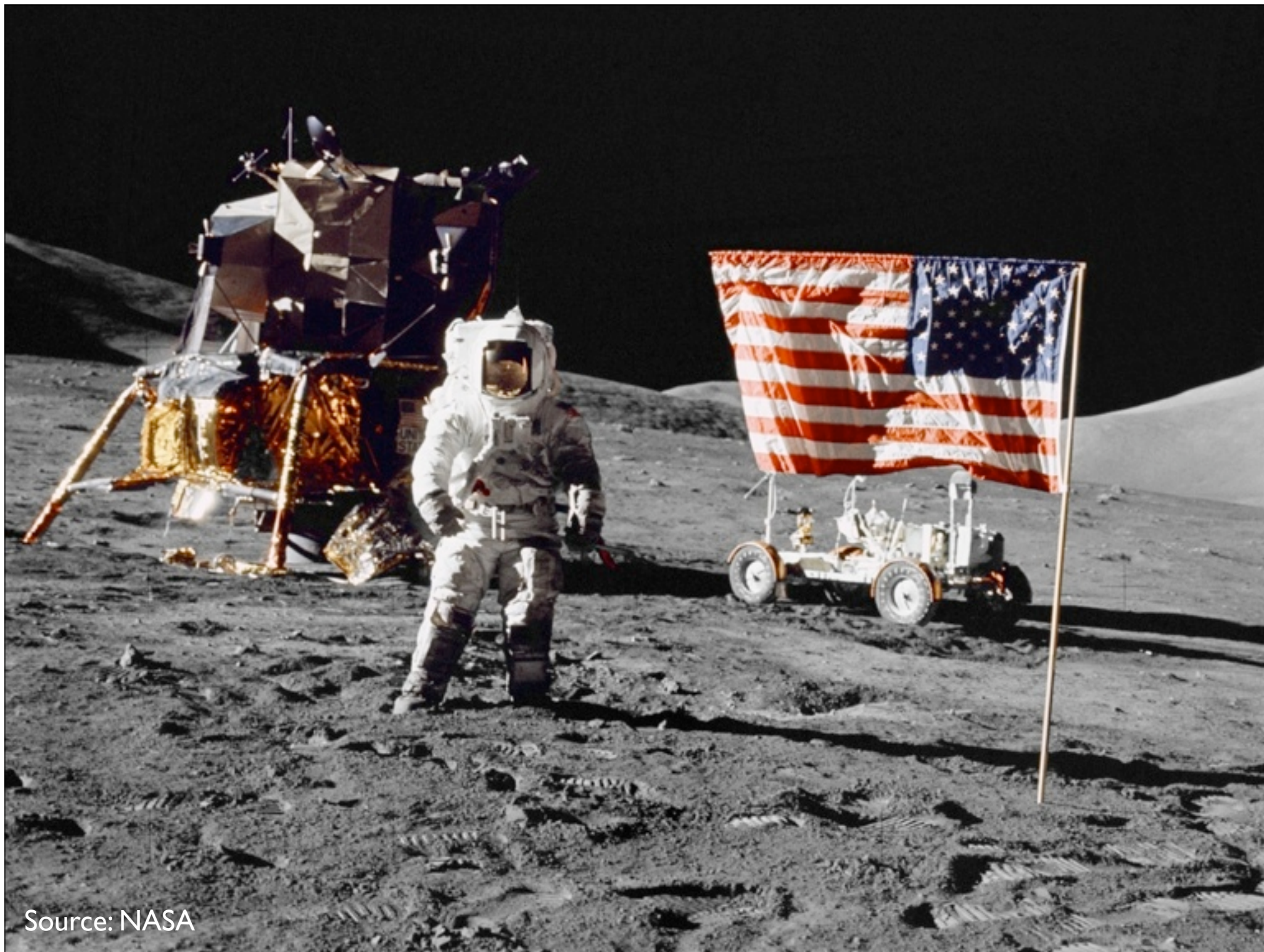
A Bit of History

- Developed in 1969-1973 by Dennis Ritchie at Bell Labs
- BCPL → B → C
- “Portable assembly language”

```
.cstring
LC0:
    .ascii "%d\0"
    .text
.globl _main
_main:
LFB3:
    pushq %rbp
LCFI0:
    movq %rsp, %rbp
LCFI1:
    subq $16, %rsp
LCFI2:
    movl $5, -4(%rbp)
    movl -4(%rbp), %esi
    leaq LC0(%rip), %rdi
    movl $0, %eax
```

A Bit of History

- Developed in 1969-1973 by Dennis Ritchie at Bell Labs
- BCPL → B → C
- “Portable assembly language”
- Four formal versions:
 - K&R (1978)
 - ANSI C aka C89 (1989)
 - C99 (1999)
 - C11 (2011)



Source: NASA



Original Design: Futurama. Image: futurama.wikia.com

In 1964 the band that changed everything came to America.
Now they're on iTunes.



Source: www.apple.com





Source: *The Toronto Star*



Source: <http://cm.bell-labs.com/cm/cs/who/dmr/picture.html>

Comparing Specs

(very back-of-the-envelope)



MacBook Pro 15"	PDP-11
2.2 GHz Intel Core i7 (Quad Core)	3.5 MHz KA11
4 GB RAM 8 GB Max	8 KB RAM 28 KB Max
\$1,799	~ \$340,000 ~ \$60,000
0.000000204 \$/Hz	~ 0.017 \$/Hz
0.000000419 \$/B	~7.30 \$/B

~~> 1,000,000~~
~~100,000~~
~~x 10,000~~
~~100,000,000~~
~~x 10,000,000~~

Implications

- Programmer time vs. computer time
 - Error checking
 - “Implementation dependent”
- Idiosyncrasies that made sense at the time
- Lots of things you *can* do, but shouldn't
- Very widely used, can help to understand hardware

UNIX History

- Operating System
- Class of systems (e.g., Mac OS X, Linux, Solaris, *etc.*)
- Grew up with C
 - Similar design goals
- Lots of little tools that each do one thing well

Let's Look at C

Two Types of Concepts

- Fundamental ideas of how we program computers
- How to express those things in C

A function is a reusable
chunk of code

```
int main(void)
{
    ...
}
```

A string is a
collection of text

```
"Hello, world"
```

A Simple Program

```
#include <stdio.h>

// This program prints: Hello, world
int main (void)
{
    printf("Hello, world\n");

    return 0;
}
```

A Simple Program

Tell compiler where to find
functions we want to use

```
#include <stdio.h>
```

```
// This program prints: Hello, world
```

```
int main (void)
```

```
{
```

```
    printf("Hello, world\n");
```

```
    return 0;
```

```
}
```

A Simple Program

```
#include <stdio.h>
```

Information for other humans

```
// This program prints: Hello, world
```

```
int main (void)
```

```
{
```

```
    printf("Hello, world\n");
```

```
    return 0;
```

```
}
```

A Simple Program

```
#include <stdio.h>
```

```
// This program prints: Hello, world
```

```
int main (void) Define a function
```

```
{
```

```
    printf("Hello, world\n");
```

```
    return 0;
```

```
}
```

}

A Simple Program

```
#include <stdio.h>
```

```
// This program prints: Hello, world
```

```
int main (void)
```

```
{
```

```
    printf("Hello, world\n");
```

Call the function “printf” with this data

```
    return 0;
```

```
}
```

A Simple Program

```
#include <stdio.h>
```

```
// This program prints: Hello, world
```

```
int main (void)
```

```
{
```

```
    printf("Hello, world\n");
```

```
    return 0;
```

```
}
```

Give the result “0” back to the OS

A Simple Program

```
#include <stdio.h>
```

```
// This program prints: Hello, world
```

```
int main (void)
```

```
{
```

```
    printf("Hello, world\n");
```

```
    return 0;
```

```
}
```

Running a Program

- Program in .c file (plain text)
- `$ gcc -o hello hello.c`
- `$ ./hello`
- No message if no errors

Warning!

```
$ gcc -o hello.c hello.c
```