

Integers & Command Line

Lecture 3

Jan 17th 2023 | COMP 211-002 | Joshua Bakita

Did you know...

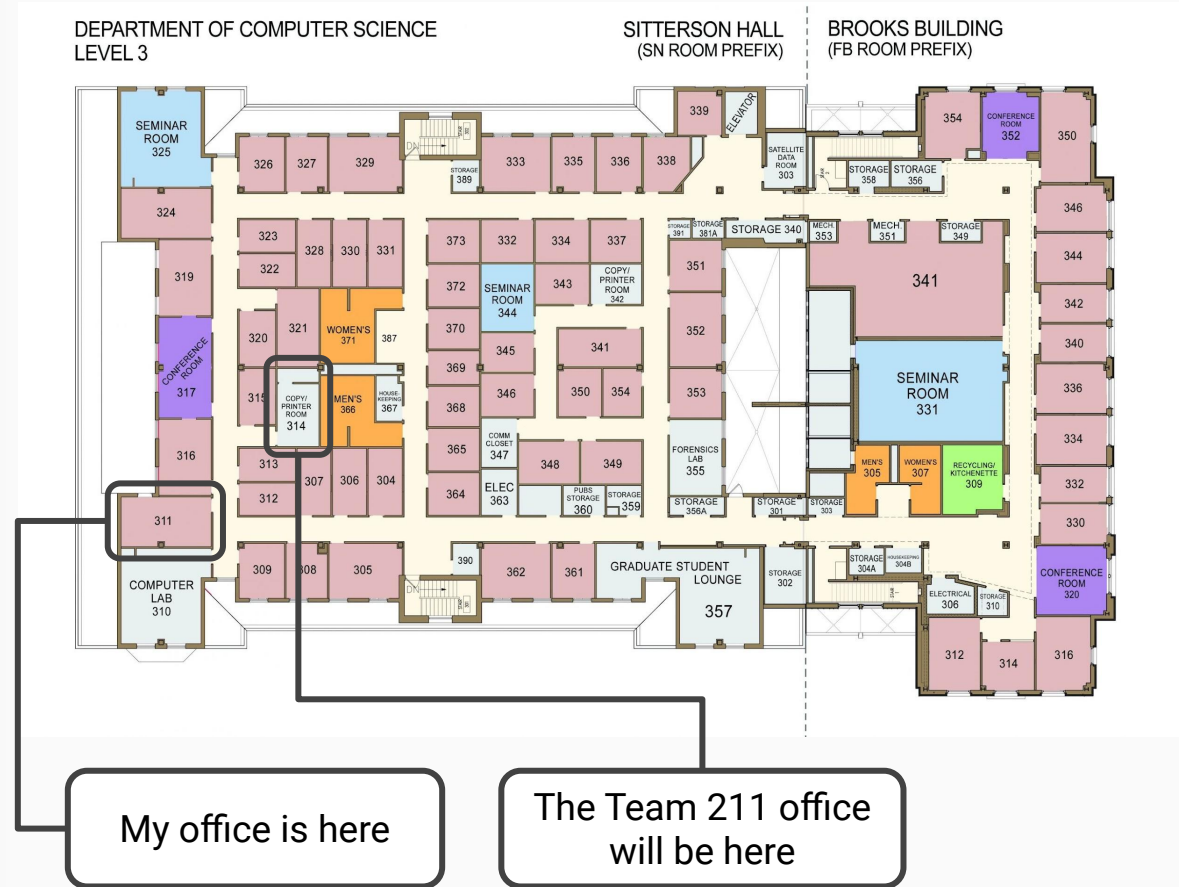
Welcome!

Today:

- Understanding Integers
- Command Line Workflow

Logistics:

- Choose a seat and stick with it! We'll be looking for you in the same location each week for attendance.
- Assign. 1 up on website; ~15% have started.
- Prefer Piazza, then email: s23-comp-211-002-staff-cs@cs.unc.edu



DEPARTMENT OF COMPUTER SCIENCE
LEVEL 3

SITTERSON HALL
(SN ROOM PREFIX)

BROOKS BUILDING
(FB ROOM PREFIX)



Prof. Bakita's Office

The Team 211 Office

Understanding Integers

Let's review the problem from last week...

Primitive Types

Detailed reference:



<https://c.emi.is/?page=data-types-x86-64>

2.2 Data Types and Sizes

There are only a few basic data types in C:

<code>char</code>	a single byte, capable of holding one character in the local character set.
<code>int</code>	an integer, typically reflecting the natural size of integers on the host machine.
<code>float</code>	single-precision floating point.
<code>double</code>	double-precision floating point.

In addition, there are a number of qualifiers that can be applied to these basic types. `short` and `long` apply to integers:

```
short int sh;  
long int counter;
```

The word `int` can be omitted in such declarations, and typically is.

From Chapter 2 of *The C Programming Language* ("K&R C")

```

    }
}

uint64_t end = get_ms();

// The below math relies on the L3 being at least 256KB (aka, buffer
double total_kbytes = (L3_SIZE/1024)*4*iterations;
if (total_kbytes/(1<<20) >= 1)
    fprintf(stdout, "Completed generating %.1fGiB of memory requests'
            total_kbytes/(1<<20));
else
    fprintf(stdout, "Completed generating %.1fMiB of memory requests'
            total_kbytes/(1<<10));
fprintf(stdout, " in %.2f seconds.\n", (end - start) / 1000.0);

free(buffer);
return 0;
}

uint64_t get_ms() {
    struct timeval tv = {0};
    if (gettimeofday(&tv, NULL) < 0) {
        perror("Unable to get current time. Terminating...");
        exit(3);
    }
    return tv.tv_sec * 1000 + tv.tv_usec/1000;
}

```

What exactly is happening here?

To see the code in context:



<https://github.com/JoshuaJB/imx6q-thrasher/blob/amd-3950x-thrasher/thrasher.c>

```

// The below math relies on the L3 being at least 256KB (aka, buffer
double total_kbytes = (L3_SIZE/1024)*4*iterations;
if (total_kbytes/(1<<20) >= 1)
    fprintf(stdout, "Completed generating %.1fGiB of memory requests'
            total_kbytes/(1<<20));
else
    fprintf(stdout, "Completed generating %.1fMiB of memory requests'
            total_kbytes/(1<<10));
fprintf(stdout, " in %.2f seconds.\n", (end - start) / 1000.0);

free(buffer);
return 0;
}

uint64_t get_ms() {
    struct timeval tv = {0};
    if (gettimeofday(&tv, NULL) < 0) {
        perror("Unable to get current time. Terminating...");
        exit(3);
    }
    return tv.tv_sec * 1000 + tv.tv_usec/1000;
}

```

Respond via Poll Everywhere



<https://PollEv.com/joshuabakita182>

I have a 32-bit integer variable
`total_mib_transferred` that I
want to check is less than a TiB.
How can I do this efficiently?

Command Line Workflow

First an example...

What parameters should I give `printf` to print a number as hex, like `0x0000BEEF` for decimal number 48879?

<https://PollEv.com/joshuabakita182>

Command Line

<code>ssh user@server</code>	Connect to server as user
<code>whatis func</code>	List manual pages for func
<code>man func</code>	View manual for func
<code>vim file</code>	Open file in vim text editor
<code>gcc file.c -o prog</code>	Compile file.c to executable file prog
<code>ls</code>	List files in current directory
<code>./prog</code>	Run executable prog
<code>exit</code>	Disconnect from server
Tab	Attempt to autocomplete command/file name

Vim Commands (Normal Mode)

<code>i</code>	Enter insert mode
<code>:w</code>	Save file
<code>:q</code>	Quit
<code>h/j/k/l</code>	Move left/down/up/right
<code>u</code>	Undo
<code>Ctrl+r</code>	Redo

Vim Commands (Insert Mode)

<code>Esc</code>	Return to normal mode
Arrow Keys	Move

Thanks! Questions?

Come chat with me now, or drop
by my office Wed. morning!
(Sitterson 311, 8:30-11:30 AM)

Assignment 1 due Jan 26.

Contact:

Email: hacker@unc.edu

Twitter: [@JJBakita](https://twitter.com/JJBakita)

Web: <https://cs.unc.edu/~jbakita>

