

CS2014

Systems Programming

Lectures:

Stephen Farrell

stephen.farrell@cs.tcd.ie, x2354

Room: WR3.4 (impossible to find!)

Teaching Assistant:

Christian Cabrera

cabrerac@scss.tcd.ie, xNNNN

See Christian @ labs

My Favourite LOC

```
#define malloc(__xxx__) \  
    (rand()%100<=30?0:malloc((__xxx__)))
```

Administrivia (1)

- Lectures:
 - Tuesday, 1500-1550, RoomTBD
 - Wednesday, 1300-1350, RoomTBD
- Labs:
 - Thursday, 1200-1600, LG12
 - Labs are for you to do/get-help-with assignments (more on those in a second)
 - It is also just fine to work on the assignment on your own computing devices
 - There are more of you than there are seats in the lab, so...

Administrivia (2)

- Lab slots:
 - Group 1: 12.00-12.50: Students whose surname begins with a letter between "O" and "Z".
 - Group 2: 13.00-13.50: Students whose surname begins with a letter between "I" and "N"
 - Group 3: 14.00-14.50: Students whose surname begins with a letter between "D" and "H"
 - Group 4: 15.00-15.50: Students whose surname has a prefix between "A" and "C"

Administrivia (3)

- 2017 timing:
 - Lecture#1: September 26th
 - Week of November 13th: no lectures!!
 - TBD: Or maybe Christian takes those slots!!
 - Last lecture: December 13th
 - Last lab: December 14th
 - Last assignment due: Jan 15th (start of 2nd semester)
 - Exams: later (April/May)

Administrivia (4)

- Exam/Assignments: 80/20

- That is: assignments are worth 20%
- You **MUST** pass both independently

```
int bummer=0; int pass;  
if (exam_mark < 0.4) {bummer=1;}  
if (assignment_mark < 0.4) {bummer=1;}  
if (!bummer) pass=1; else pass=0;
```

- There will be 6 marked assignments, each with a deadline

Administrivia (5, and the last!)

- Assignment deadlines:
 - Assignment1: deadline 2017-10-09
 - Assignment2: deadline 2017-10-16
 - Assignment3: deadline 2017-10-23
 - Assignment4: deadline 2017-11-13
 - Assignment5: deadline 2017-12-04
 - Assignment6: deadline 2018-01-15

Systems Programming: what's that?

- As always, it depends who you ask:-)
 - <https://duckduckgo.com/html?q=systems%20programming>
- Wikipedia definition#1 is most like this course, saying systems programming is lower level than application programming and can make assumptions about the system on which the code is run.
 - Often related to how the OS/system interacts with higher level code/users.
- DevOps trend though maybe changes this some: A lot of “systems programming” in future may involved python, Rust, Go, PHP/Node.js and JS etc as the boundaries between applications and systems blur.
- For now though, we'll stick with what Wikipedia says (today:-), on the basis that if you learn a couple of programming languages, the rest are mostly the same and you can learn them yourself later as needed (and fashions will change before you do that, probably;-)

Course Content

- Introduction
- A bit about using GNU/Linux
- Various bits of C programming
- Safety with dangerous implements
- A touch of C++ programming
- Something interesting for the end, that I've yet to decide upon
- And 6 programming assignments

What's our goal here?

- You'll learn C/C++ and a bit about the systems we use for doing assignments
- Real goal is that you can figure this stuff out for yourself when you later come across similar problems/challenges/opportunities:
 - On different systems
 - With different languages
 - Using different libraries
 - With or without handholding IDEs
- It is entirely fine that you have to look stuff up when you forget that, esp about libraries (I do that all the time)
 - Web search -> stackexchange etc. is almost entirely a fine thing
 - As always though, don't believe all you read

My Favourite LOC

```
#define malloc(__xxx__) \  
    (rand()%100<=30?0:malloc((__xxx__)))
```

- Let's shift from slideware to HTML:

<https://down.dsg.cs.tcd.ie/cs2014/examples/bm/README.html>