UPPSALA UNIVERSITET

Programming in C

You will learn the details and practice in the labs (important)!
We assume that you have at least 10hp in other programming courses.
(Some of you know C well, some don't)

Jarmo Rantakokko

1

UPPSALA UNIVERSITET

Linux Environment

We will not use any IDE but work directly in a terminal window issuing commands to the operating system Linux. (Use VScode, Xcode on your comp).
Linux: Open-source Unix-like operating system, first version 1991 by Linus Torvalds (Finnish-American software engineer)

Common Linux-commands:

pwd	- Show current directory name and path
cd	- change directory
mkdir	- make directory
ls	- list all files in directory
cat	- display file all at once
more	- display file filling one page
rm	- remove file
rmdir	- remove directory
cp	- copy file
mv	- move file
man	- show manual for a command

…. and many more commands

Linux nerd joke:
Q: What do computers and air conditioners have in common?
A: They both become useless when you open Windows!

You can also run scripts issuing commands directly to the OS, e.g., shellscripts (Python-like language)
https://www.shellscript.sh
Automatic checking of assignments will be done using shellscripts.

2

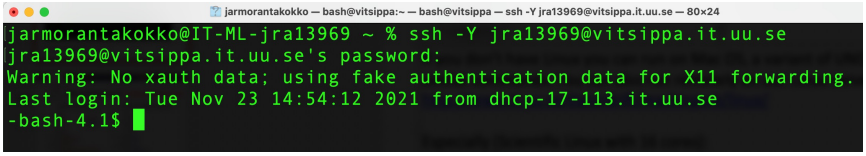
Linux Environment

If you don't have Linux you can run on Mac OS, a variant of UNIX with the same or similar commands. You can also run remotely on our Linux servers using ssh: http://www.it.uu.se/datordrift/maskinpark/linux/

Especially (Scientific Linux with 16 cores):

- gullviva.it.uu.se
- tussilago.it.uu.se
- vitsippa.it.uu.se

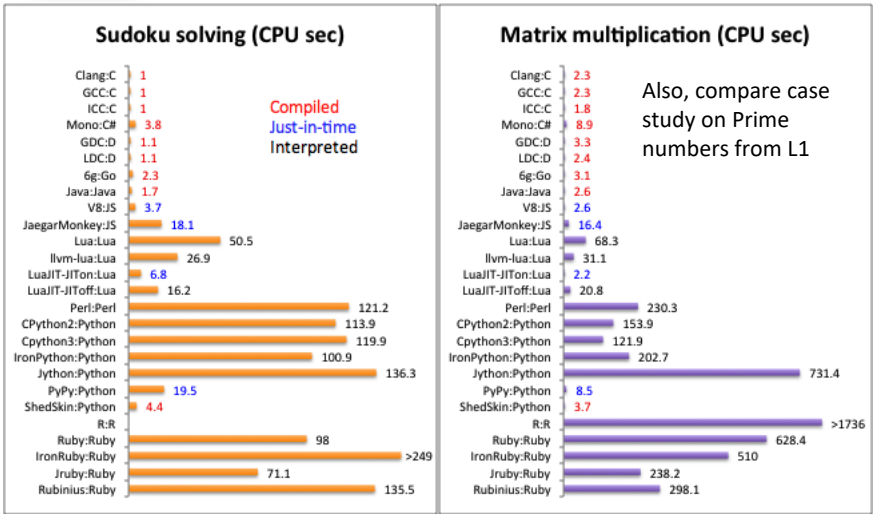
Don't misuse the systems by running large parallel programs for a long time



In PC labs, start the *thinlinc client* (server thinlinc.student.it.uu.se) or a terminal and use ssh to linux-servers. See lab1 for further details.

3

Why should we use C? #1 Reason Performance!



Also, compare case study on Prime numbers from L1

http://attractivechaos.github.io/plb/

4

UPPSALA UNIVERSITET

Programming languages 2022, PYPL index

Interpretated
Just In Time
(=slow)

Rank	Change	Language	Share	Trend
1		Python	28.27 %	-2.0 %
2		Java	18.03 %	+0.8 %
3		JavaScript	8.86 %	+0.4 %
4		C#	7.51 %	+0.6 %
5		C/C++	7.32 %	+0.6 %
6		PHP	5.71 %	-0.4 %
7		R	4.23 %	+0.5 %
8		Objective-C	2.28 %	-1.2 %
9	↑	TypeScript	2.11 %	+0.3 %
10	↓	Swift	2.01 %	+0.2 %

https://statisticsanddata.org/data/the-most-popular-programming-languages-1965-2021/

PYPL index, how often language tutorials are searched on Google.

5

UPPSALA UNIVERSITET

TIOBE Index for March 2021

Search...

Mar 2021	Programming Language	Ratings	Ratings chart	Change
1	C	15.33%		-1.00%
2	Java	10.45%		-7.33%
3	Python	10.31%		+0.20%
4	C++	6.52%		-0.27%
5	C#	4.97%		-0.35%
6	Visual Basic	4.85%		-0.40%
7	JavaScript	2.11%		+0.06%
8	PHP	2.07%		+0.05%
9	Assembly language	1.97%		+0.72%
10	SQL	1.87%		+0.03%

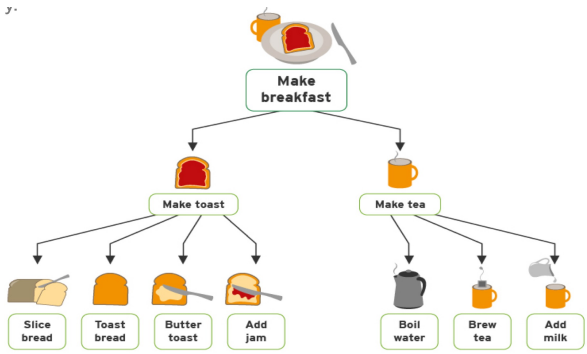
TIOBE ratings are calculated by counting hits of the most popular search engines.
(https://www.tiobe.com/tiobe-index/programming-languages-definition/)

6

UPPSALA UNIVERSITET

Characteristics of C

Functional programming: Programming paradigm where you decompose your problem in to smaller building blocks and implement as functions. C is functional – low level, compact, efficient. Key concept is functions.



7

UPPSALA UNIVERSITET

Characteristics of C

- Interpreted v/s **compiled**
- Object oriented v/s **functional**.
- Dynamic typing v/s **static typing**.
- Garbage collected v/s **no garbage collection**.
- Ease of development v/s **execution speed**.

8

UPPSALA UNIVERSITET

Brief history

- Originally developed by Dennis Ritchie for OS implementations 1972.
- Standardized in 1989 (ANSI-C or C90).
- Revised in 1999 (C99).
- And again in 2011 (C11).

Supported by most compilers

- And C17 exist, new version (C2x)

Image credit: Dennis Ritchie

9

UPPSALA UNIVERSITET

A first example

You will need an editor to write code in: emacs, vi, gedit, …, Xcode on Mac, VS on Windows.

C hello.c > …

```c
/* Prints 'Hello, world!' */
#include <stdio.h>

int main(){
    printf("Hello, world!\n");
    return 0;   // exit status
}
```

Compile (in terminal window):

```
> gcc -o hello hello.c
```

Run:

```
> ./hello
Hello World!
```

Comments in C11:

- multi-line with /* … */, single-line with //.

The `#include` pre-processor directive

- to include the input/output standard library of functions - `stdio.h`.

The function `main`:

- where code execution starts. Returns an integer exit status. Can include input arguments.
- Brackets { } defines a code block with local variables – local scope

A statement `printf`:

- directs text to the standard output.
- Semi colon ends statements in C

A statement `return`:

- returns the 'normal' exit status 0 to the OS, return 0 – no errors!

10

UPPSALA UNIVERSITET

C-language

C is a small and compact language with a few built in concepts:

- Standard types: char, int, long, float, double,…
- Other types: arrays, pointers, struct, typedef, void
- Loops: for, while
- Conditional statements: if, else, switch
- Functions
- …

Learning these will be enough for most applications and we will go deeper into different topics in this course. You will be skilled after doing all labs and assignments. Prepare for hard work!

Also, google is your friend but never copy code or sections of code without a reference. We will report plagiarism!

11

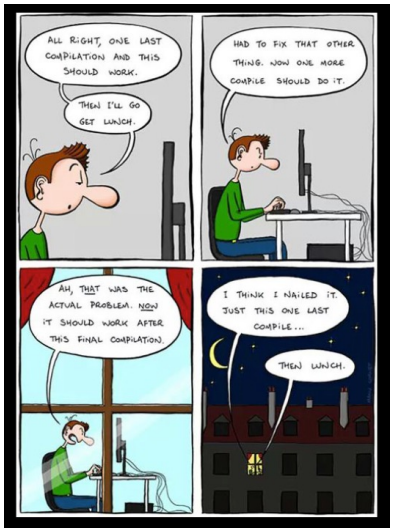
UPPSALA UNIVERSITET

C-language

Just don't forget about your lunch…

This might also be true?

Student in HPP class



12

Operators (basic)

- Binary arithmetic (two operands): + - * / %
- Unary arithmetic (single operand): - ++ -- (prefix and postfix)
- Assignment: = += -= *= /=

```
a+=1 <--> a=a+1
          a++
```

```
a=5; c=5;
b=++a; // b=6
d=c++; // d=5
```

- Relational: < <= > >= != ==
- Logical: && || ! (and, or, not)
- Mathematical functions in math.h (#include <math.h>)

13

Constructs: the if statement

- Variants:
  - if (**c**) s
  - if (**c**) s1 else s2
  - **c** is any expression and s, s1 and s2 are statements.
- **c** is false if it evaluates to 0, 0.0 or NULL, otherwise true.
- Multiple statements can be organised together within curly brackets { }.
- Indentation no meaning, except more readable code.

14

UPPSALA UNIVERSITET

Constructs: the if statement

Example Leap year: A year is a leap year if it is divisable with 4, a century is a leap year if it is divisable with 400.

```
#include <stdio.h>
int main() {
    int year;
    printf("Give a year: ");
    scanf("%d",&year);
    if (year%4==0)
        if (year%100==0)
            if (year%400==0)
                printf("Year %d is a leap year\n",year);
            else
                printf("Year %d is not a leap year\n",year);
        else
            printf("Year %d is a leap year\n",year);
    else
        printf("Year %d is not a leap year\n",year);
    return 0;
}
```

Need to declare variables with type

scanf reads from standard input and needs the & to return a value

15

UPPSALA UNIVERSITET

Constructs: the if statement

Alternative short solution:

```
#include <stdio.h>
int main() {
    int year;
    printf("Give a year: ");
    scanf("%d",&year);
    if ((year%4==0) && ((year%100!=0) || (year%400==0)))
        printf("Year %d is a leap year\n",year);
    else
    {
        printf("Year %d is not a leap year\n",year);
        printf("Why bother asking, you should know\n");
    }
    return 0;
}
```

Note: if year%4 not 0 the other statements will not be checked, think about the order for efficiency!

(Note the brackets on the else-statement – Indentation has no meaning in C.)

16