Introduction to operating systems

Module 0

Lecture

Operating systems 2020

1DT003, 1DT044 and 1DT096

2020-01-20 karl.marklund@it.uu.se Uppsala university

Karl Marklund

Course responsible lecturer for OS (1DT044) and OSPP (1DT096)

Lecturer for the OS part of DSP (1DT003)

Contact

Primarily use Piazza for questions where the answer may be of interest to other students

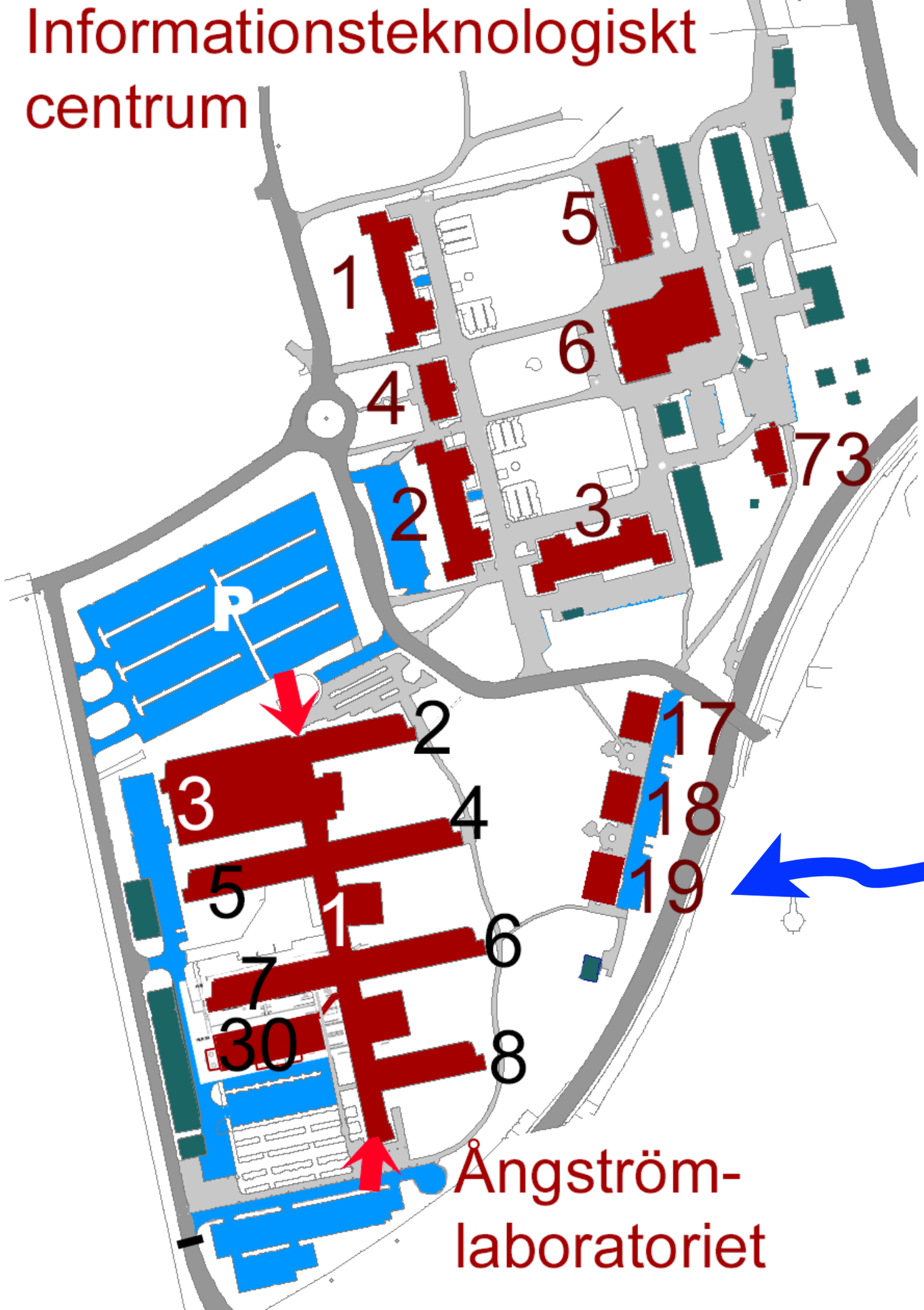
Only contact me directly if Piazza is not suitable

karl.marklund@it.uu.se

Room P19210b

Du kan kontakta mig på **svenska**

You can contact me in **English**



Karl Marklund

Room P19210b (second floor)

Piazza

If you haven't done so yet, make sure to register in Piazza

`piazza.com/uu.se/spring2020/dsposospp`

Announcements from the teaching staff

Post questions

Discuss questions with students and teaching staff

Tag all questions with **course** tag, **module** number and other appropriate tags

Slides

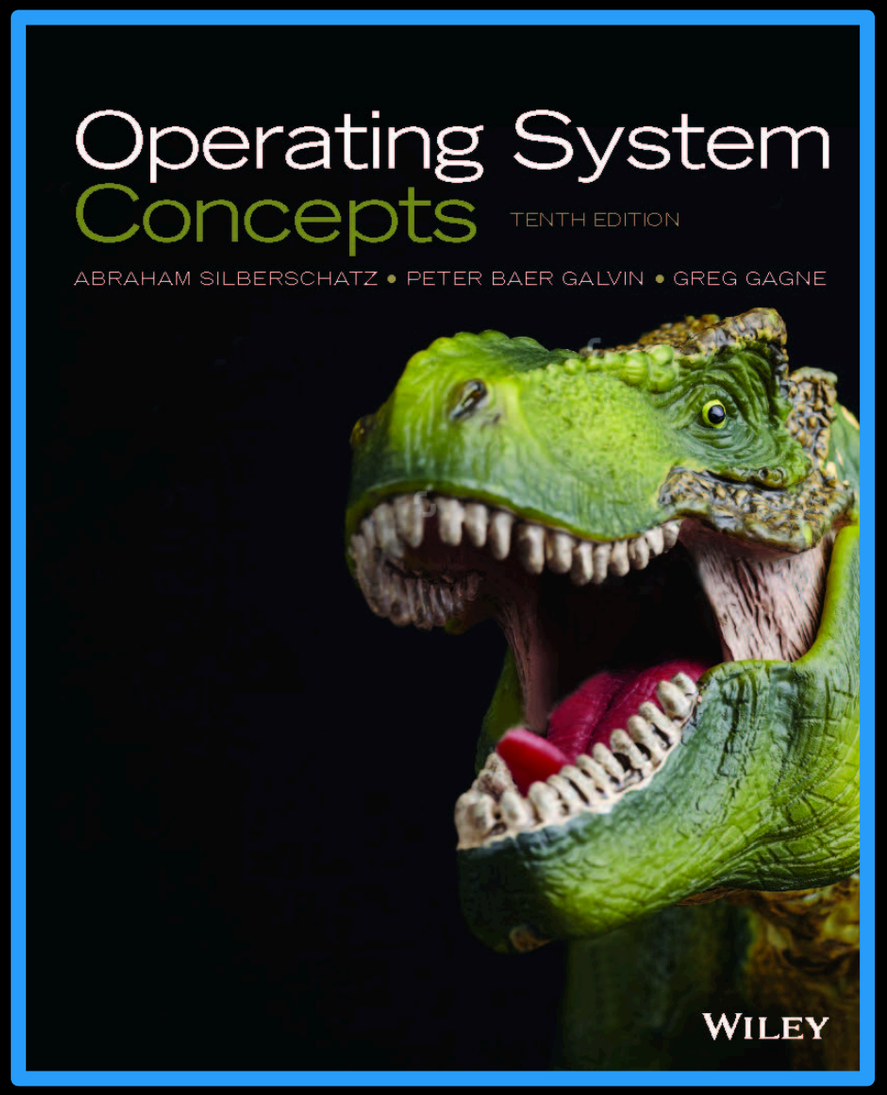
Slides for all lectures will be added to the following GitHub repositories right before the lecture starts

`http://bit.do/os-ospp-slides-2020`

`http://bit.do/dsp-slides-2020`

No text book required

There is no required text book for the OS content of the courses. If you would like to have a text book, the following book is highly recommended.



Operating System Concepts

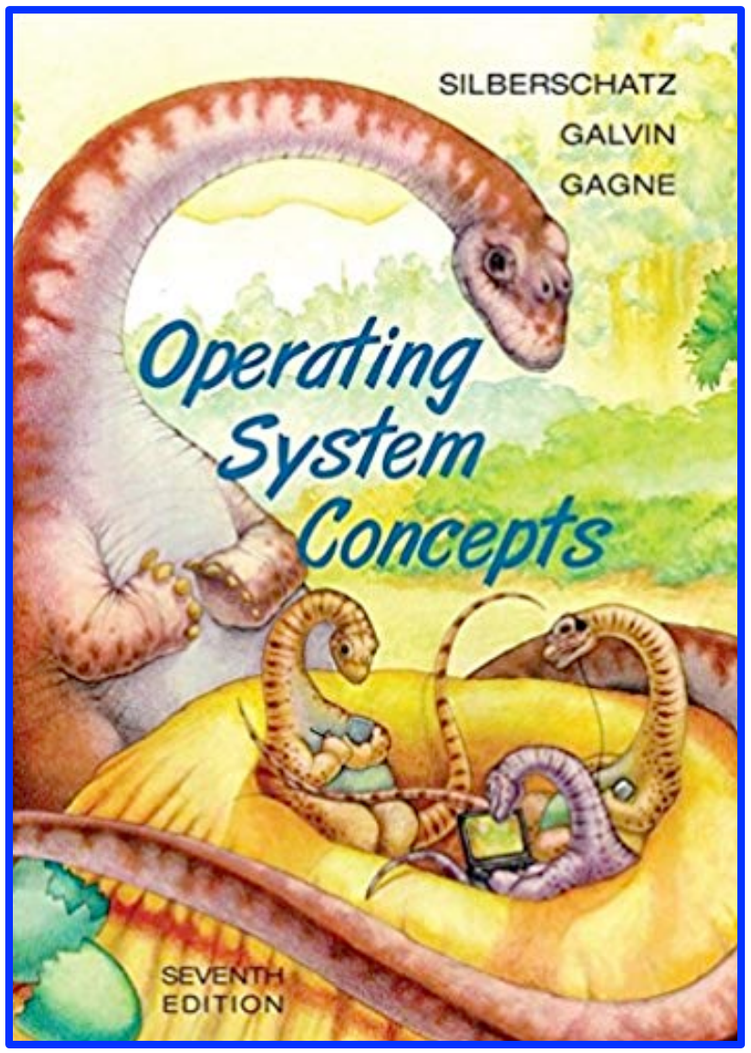
10th edition

Avi Silberschatz

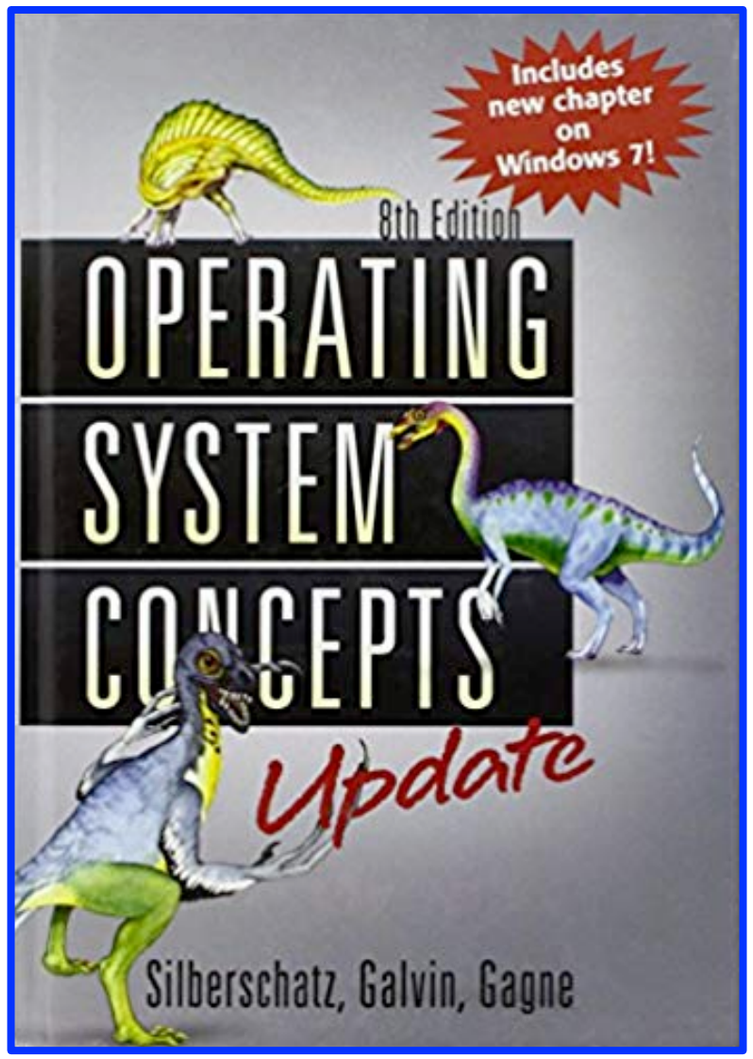
Peter Baer Galvin

Greg Gagne

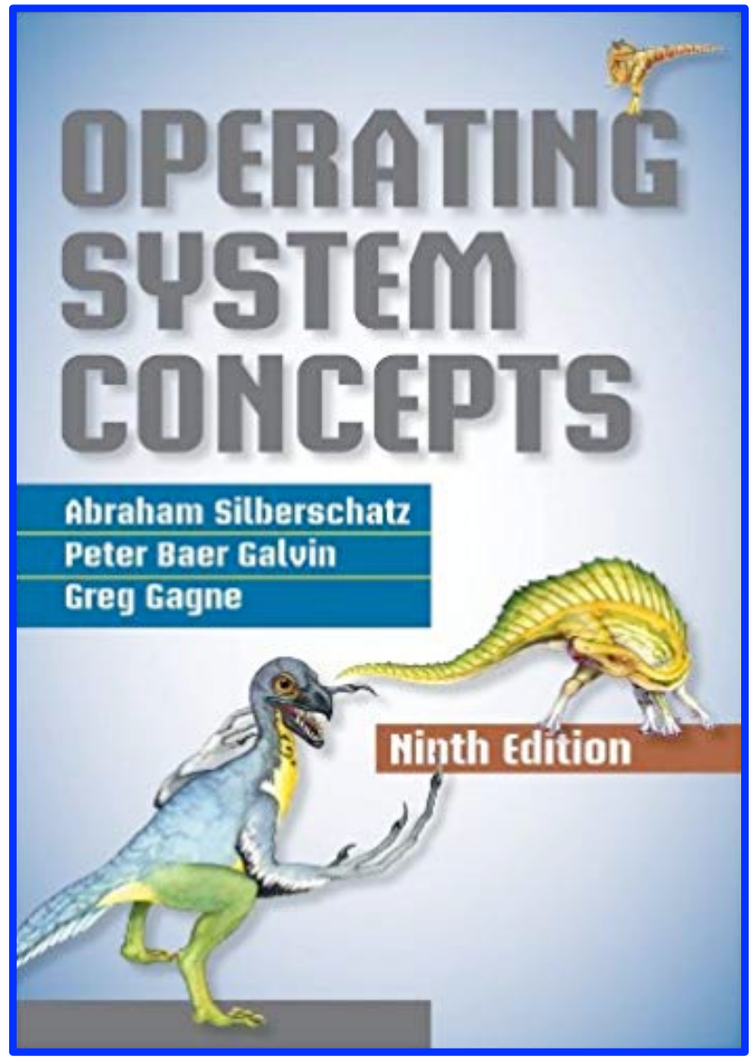
Older editions will do just as fine as the latest 10th edition.



7th edition



8th edition



9th edition

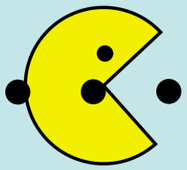
Operating Systems

- What is an operating system?
- Why is this interesting?
- Why should we learn about this?

User 1 User 2 User 3 User N




Compiler Spreadsheet Text Editor Pacman

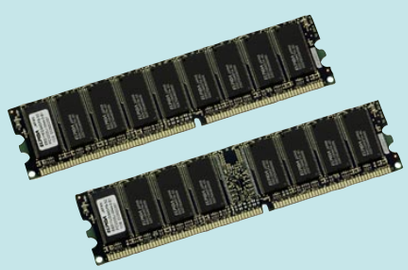


System and Application Programs

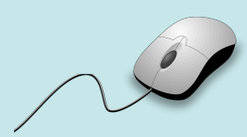
Operating System

Controls the hardware and coordinates its use among the various application programs for the various users.

Computer Hardware





Learning about **operating systems** is both **useful** and **fun**

You leverage what you learned in other courses

- ★ Computer architecture
- ★ Assembly programming
- ★ C programming

System and Application Programs

Operating System

You will learn how an operating system provides an **environment for programs to execute** (on a single core CPU) seemingly *”at the same time”* sharing the CPU without the programs knowing about each other or knowing the details of the actual hardware.

Computer Hardware

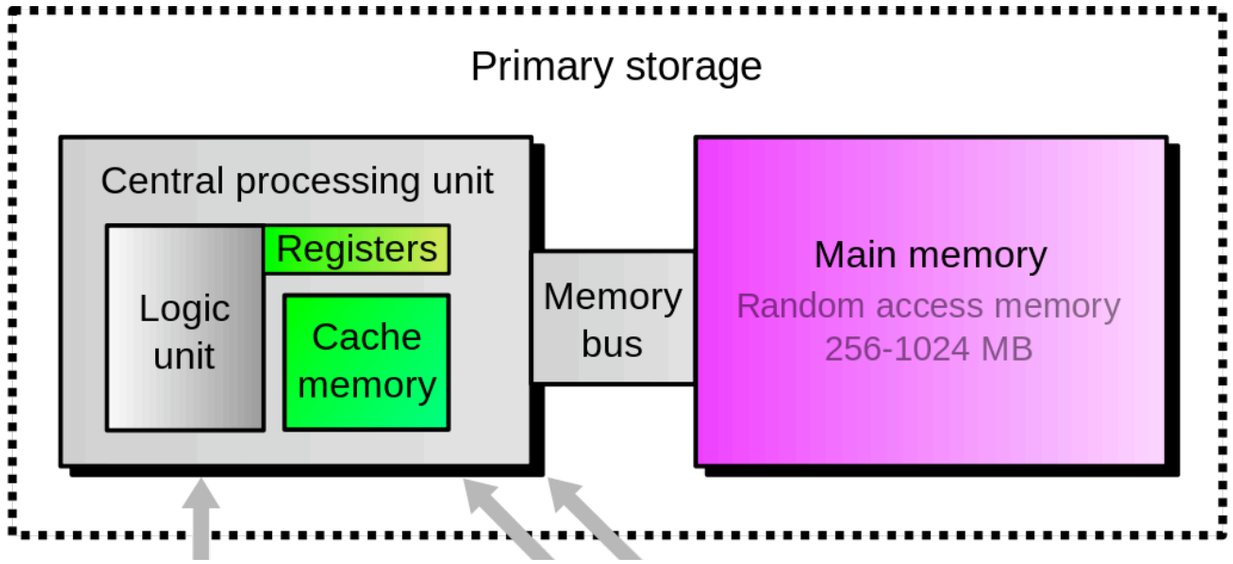
After this course you will have a better understanding of what is happening ”under the hood” and thereby **become** a much **better engineer** and/or **programmer**.

After completing this course you should have **enough knowledge** to **program** a small and **simple operating system** all by yourself if you put in some effort.

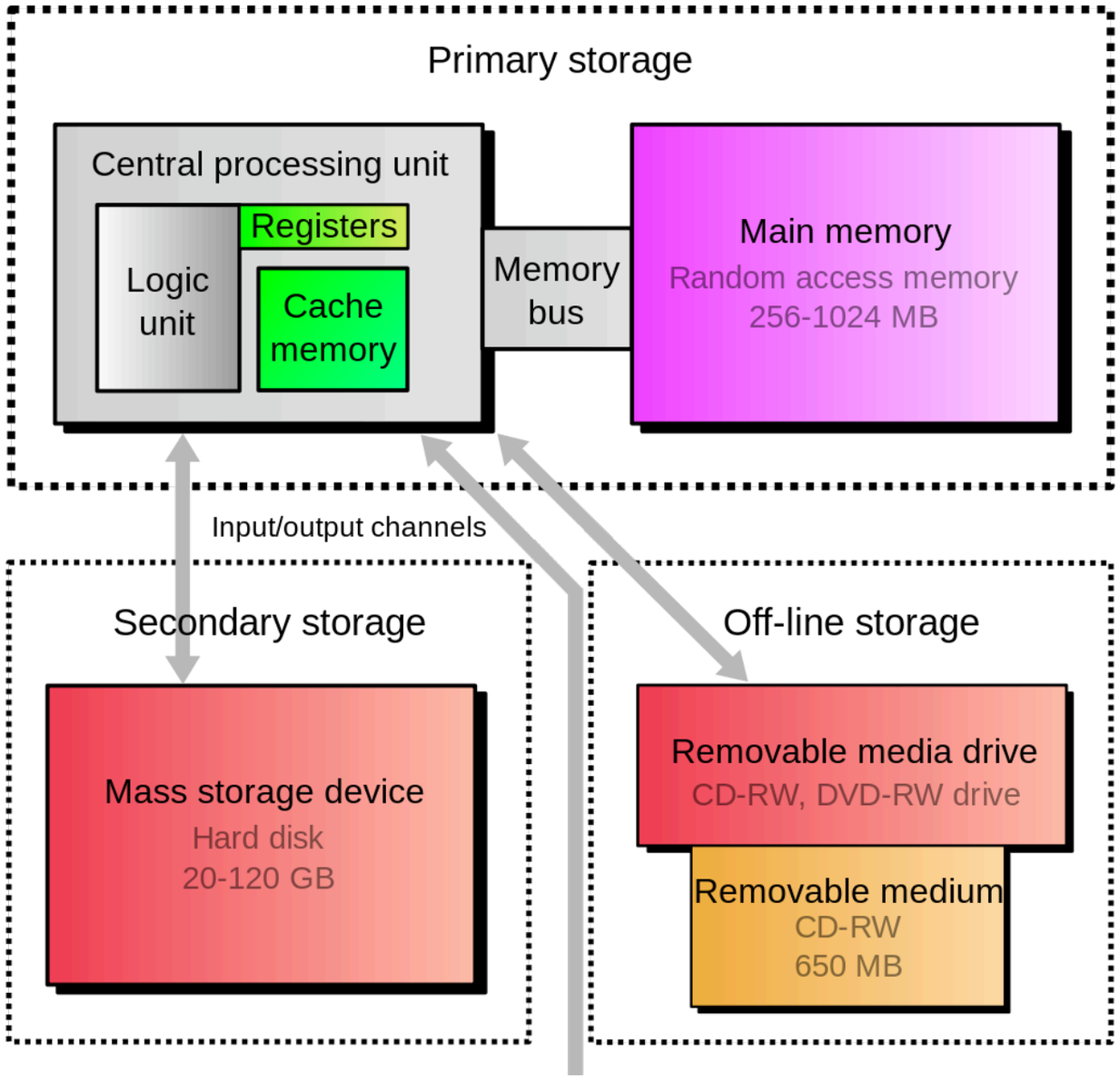
Computer storage

Hierarchy of computer data storage

- Primary storage
- Secondary storage
- Tertiary storage
- Off-line storage



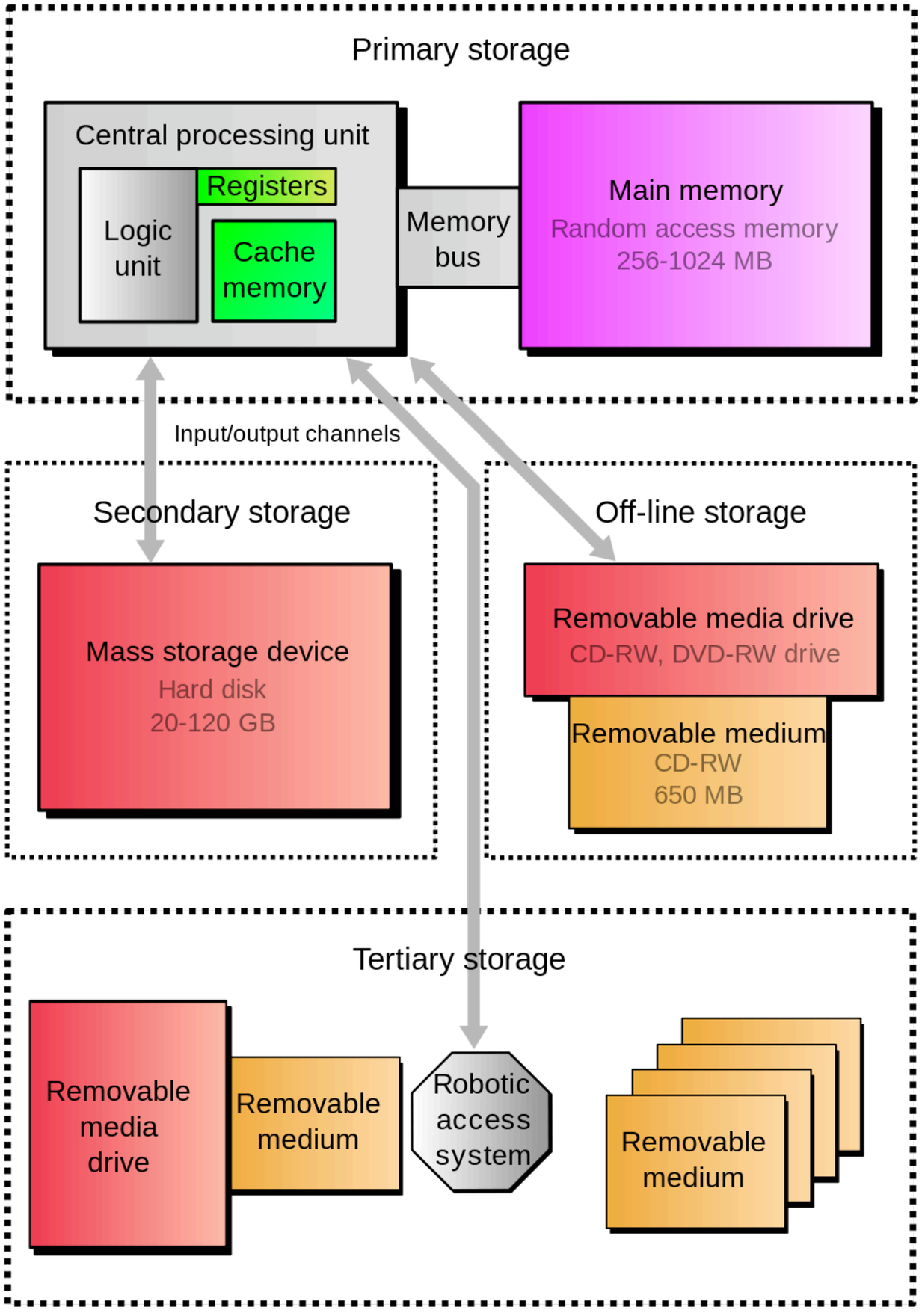
Directly accessible to the CPU



Directly accessible to the CPU

Not directly accessible to the CPU

Source https://en.wikipedia.org/wiki/Computer_data_storage#Secondary_storage



Directly accessible to the CPU

Not directly accessible to the CPU

Primarily used for archiving rarely accessed information.

Typically involves a robotic mechanism which will mount (insert) and dismount removable mass storage media into a storage device. Data often copied to secondary storage before use.

Source https://en.wikipedia.org/wiki/Computer_data_storage#Secondary_storage

Input/Output

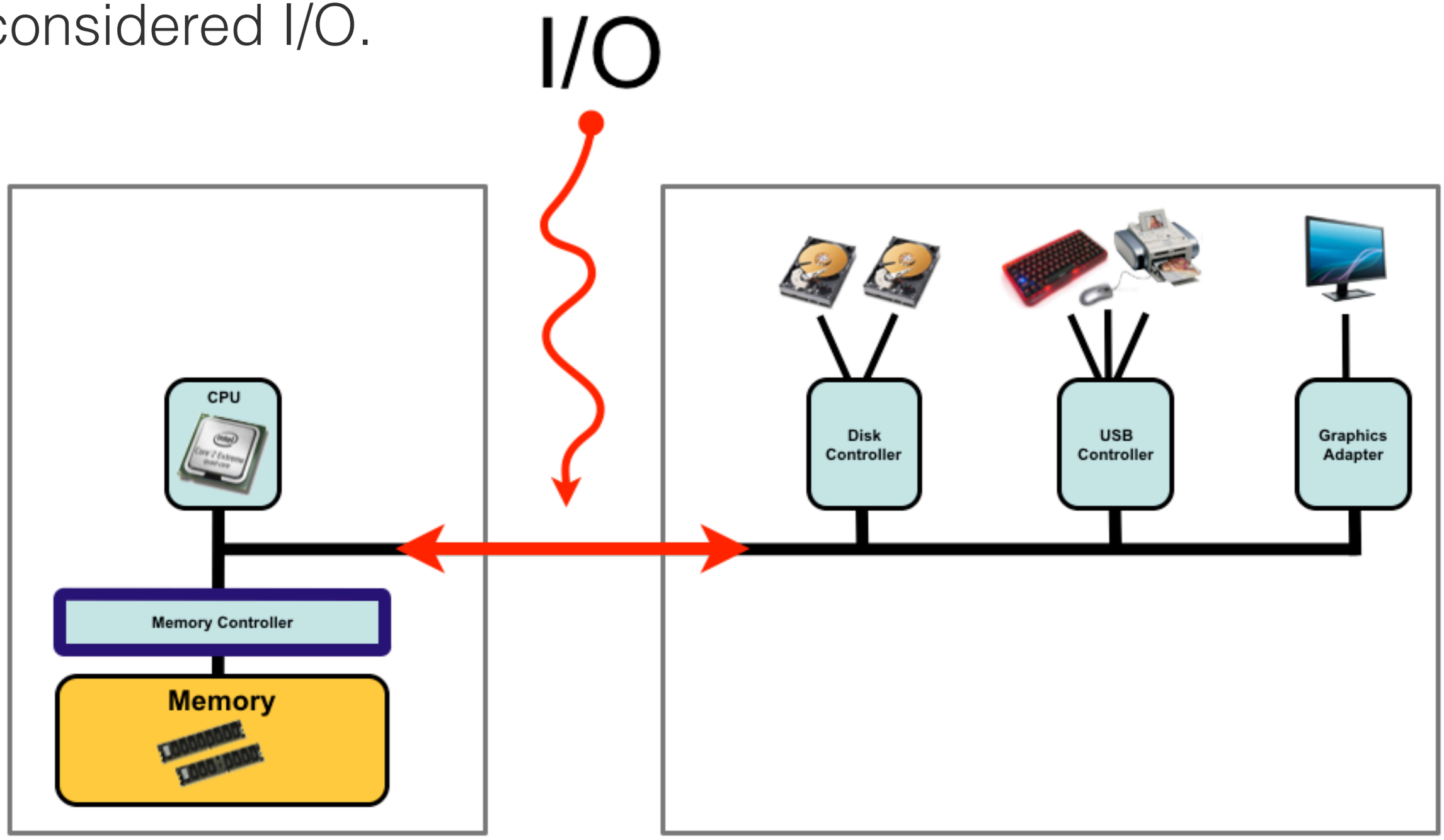
Input/Output (I/O)

In computing, input/output or I/O (or, informally, io or IO) is the communication between an information processing system, such as a computer, and the outside world, possibly a human or another information processing system.

Inputs are the signals or data received by the system and outputs are the signals or data sent from it.

Source https://en.wikipedia.org/wiki/Input/output 2018-01-23

For our purposes, any transfer of information between the CPU/Memory and any of the external devices is considered I/O.

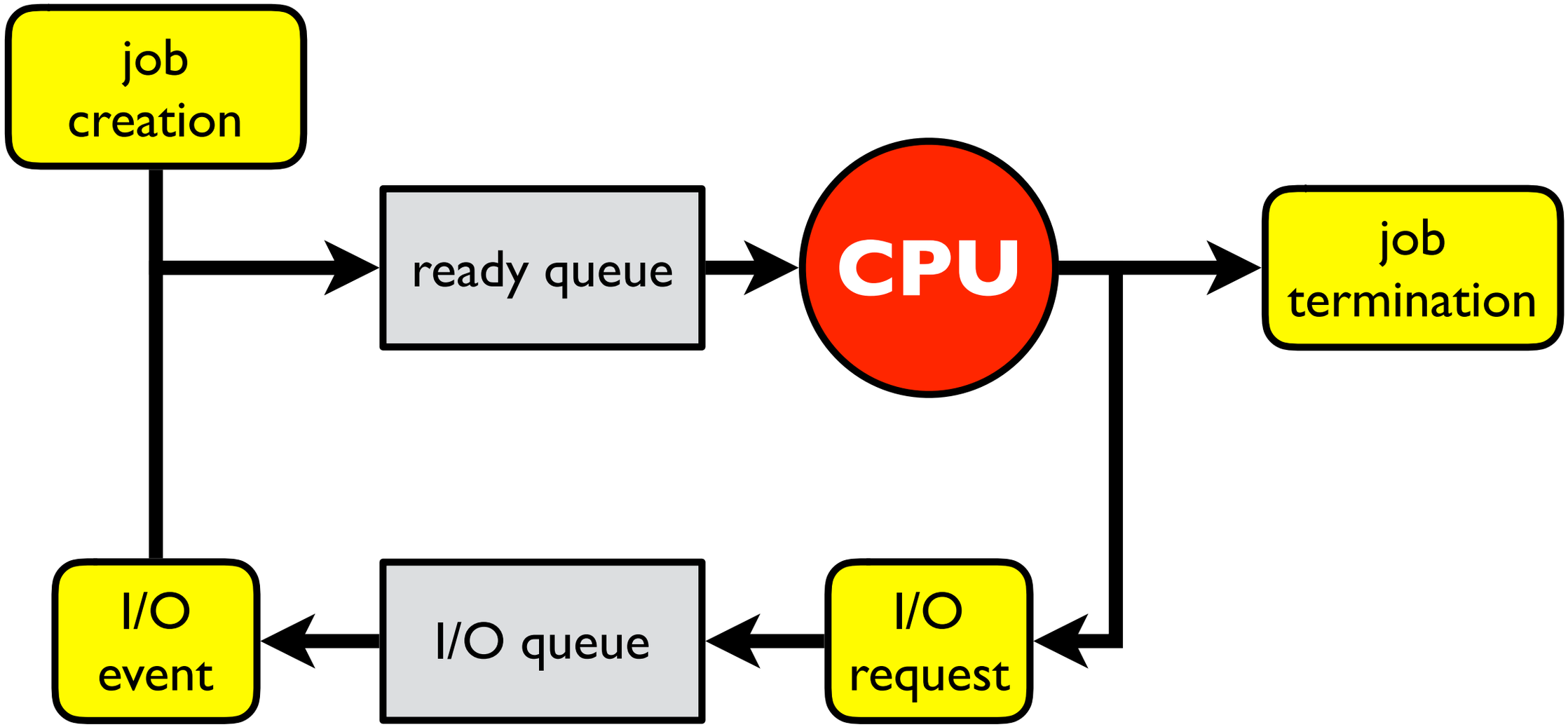


Compared to the speed of the CPU, I/O operations are very, very, **slow**.

Multiprogramming and Multitasking

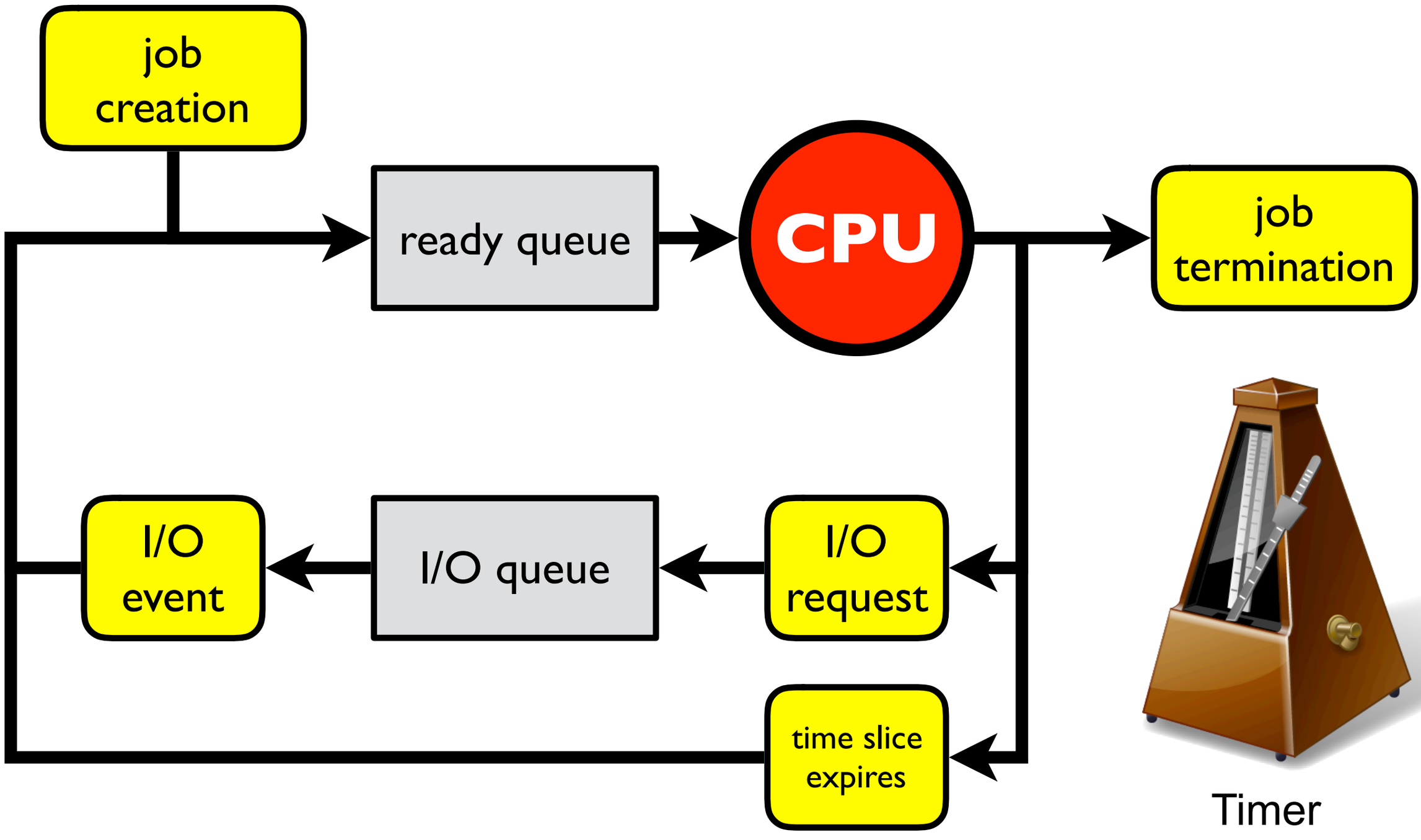
Multiprogramming

A schematic view of multiprogramming



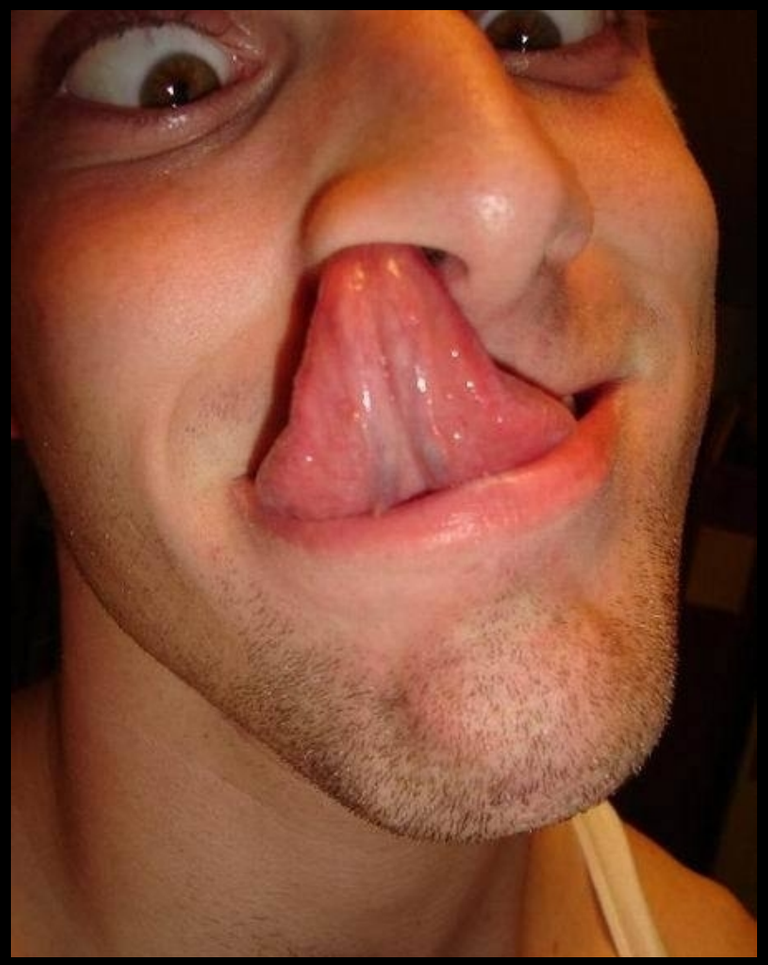
Multitasking

A schematic view of multitasking



Timer

What are you really good at?



How did you become good at this?

We learn in different ways

As a rough estimate the following conclusions can be made regarding how much you remember after a learning activity.

20 % after **listening**

30 % after **reading**

70 % after **talking**

90 % after **doing**

This tells us that listening to lectures and reading on your own may not be the most efficient way to learn ...

The cognitive domain

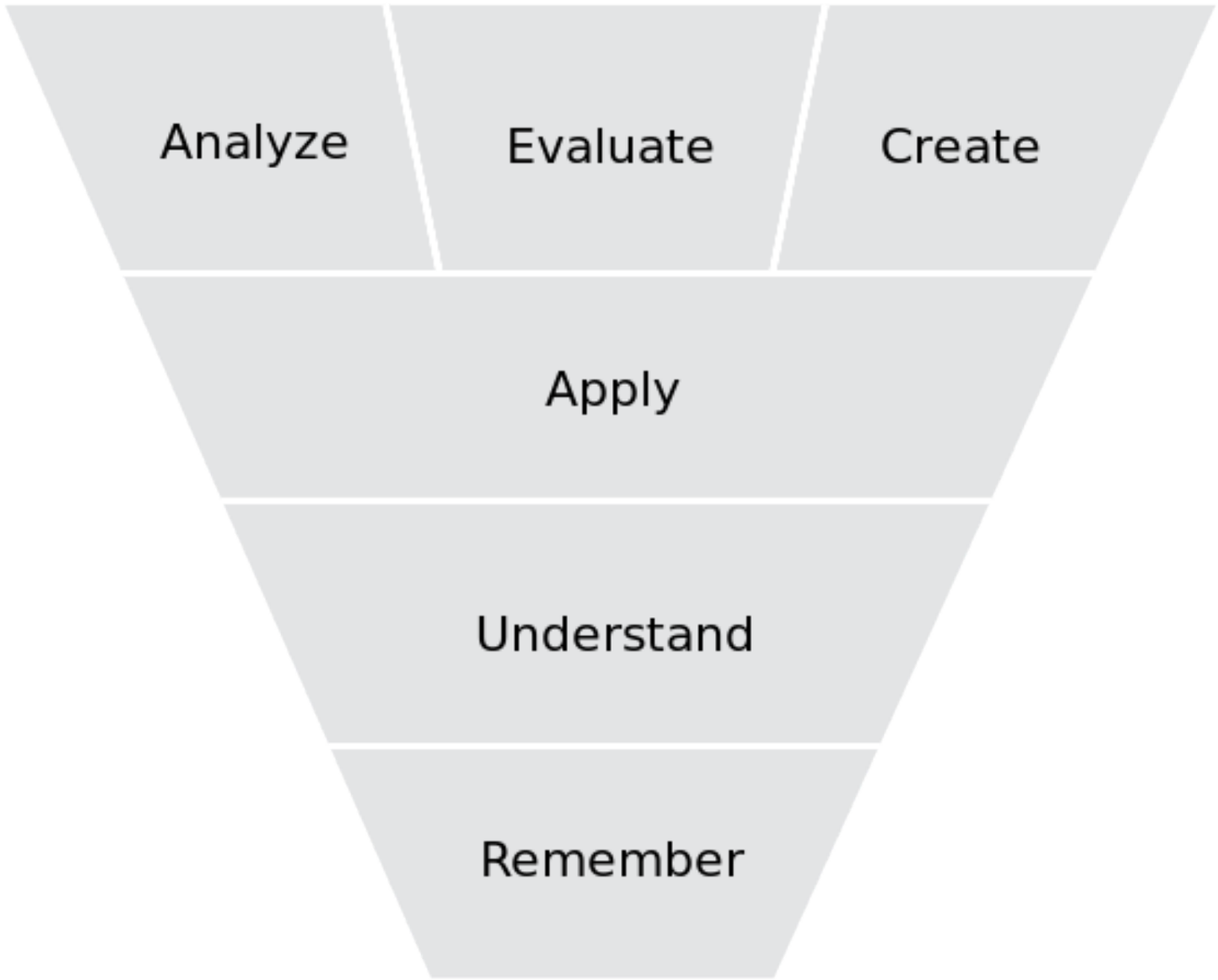


Image: http://commons.wikimedia.org/wiki/File:BloomsCognitiveDomain.svg 2016-01-17

Thinking and learning

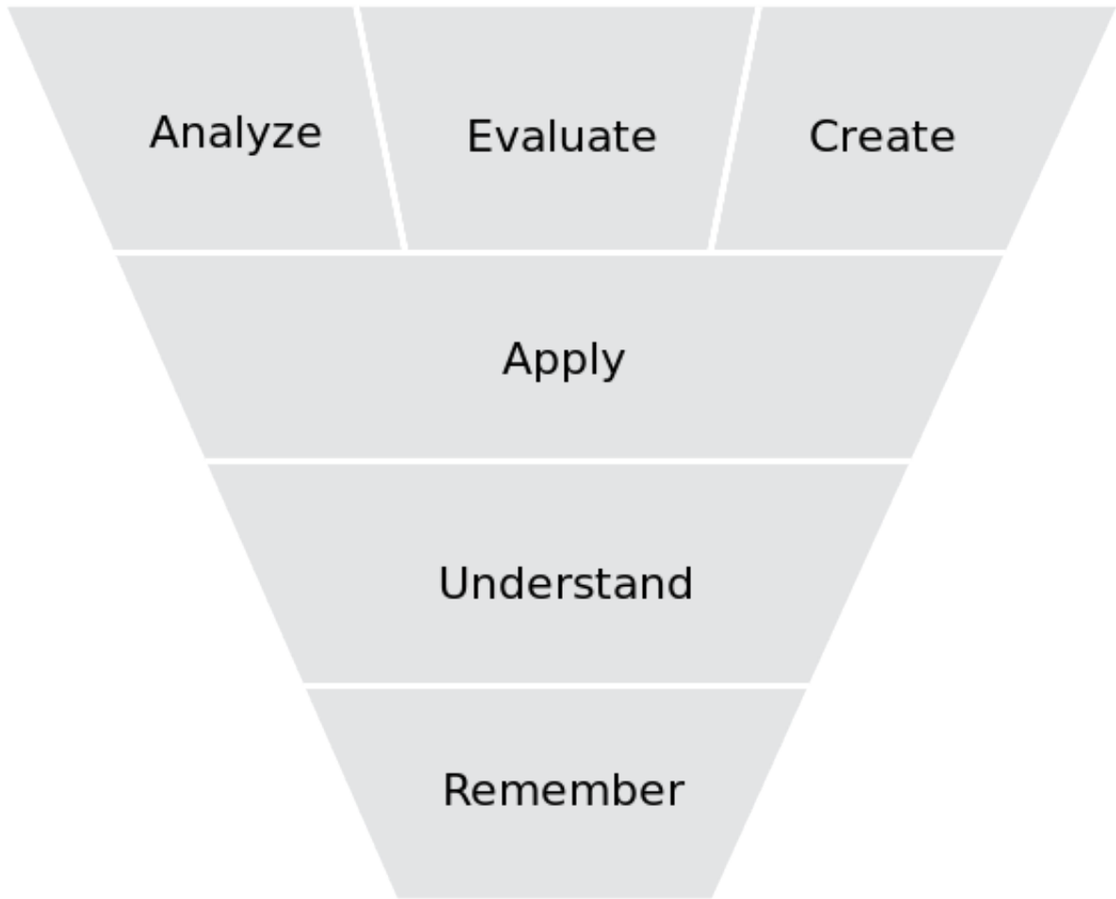
Seems like a good idea to combine these two models of learning and thinking.

20 % after **listening**

30 % after **reading**

70 % after **talking**

90 % after **doing**



Learning and examination activities

Teaching and examination is divided into four groups of activities:

- **learning**
- **examination**
- **higher grade**
- **retake**

- Participation in all **learning activities** is highly recommended but **not mandatory**.
- Participation is **mandatory** for all **examination** activities.
- To pass the course you **must pass all examination activities**.

Learning

Type	Abbreviation	Activity
Learning	L	Lecture
Learning	W	Workshop
Learning	T	Tutoring
Learning	Tr	Retake tutoring

Examination

Type	Abbreviation	Activity	Grading scale
Examination	S	Seminar	Fail/Pass
Examination	C	Code grading	Fail, 3
Higher grade	Ch	Higher grade code grading	Fail, 3, 4, 5
Examination	P	Oral presentation of a case study by the study group	Fail/Pass
Examination	E	Final written exam	Fail, 3, 4, 5

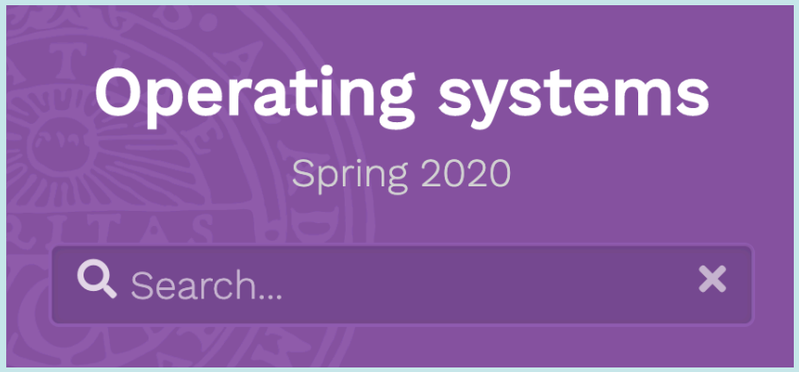
Retake

Type	Abbreviation	Activity	Grading scale
Retake	Sr	Retake seminar	Fail/Pass
Retake	Cr	Retake code grading	Fail, 3
Retake	Pr	Retake case study presentation	Fail/Pass
Retake	Er	Retake written exam	Fail, 3, 4, 5

Course homepages

More information about each activity can be found on the **course home pages**.

Course homepages



OS (1DT044) and OSPP (1DT096)

Link in the Studentportal

https://www.it.uu.se/education/course/homepage/os/vt20

DSP (1DT003)

Link in the Studentportal

https://www.it.uu.se/education/course/homepage/dsp/vt20

Modules

Modules			OS related content (DSP might have additional content)
DSP	**OS**	**OSPP**	
0	0	0	Course overview and preparations.
1	1	1	Fundamental concepts.
2	2	2	The process concept and inter process communication. File descriptors, standard streams, and I/O redirection.
3	3	3	CPU Scheduling.
4	4	4	Threads, synchronization and deadlock.
5	5	5	Memory management, files and file systems.
6	-	-	Cryptography and security (DSP only)
7	6	6	Case study
8	7	7	Written exam and exam preparations.
-	-	8	Erlang and other concurrent programming models.
9	-	9	Group project.

Course timelines

http://bit.do/dsp-os-ospp-timelines-2020

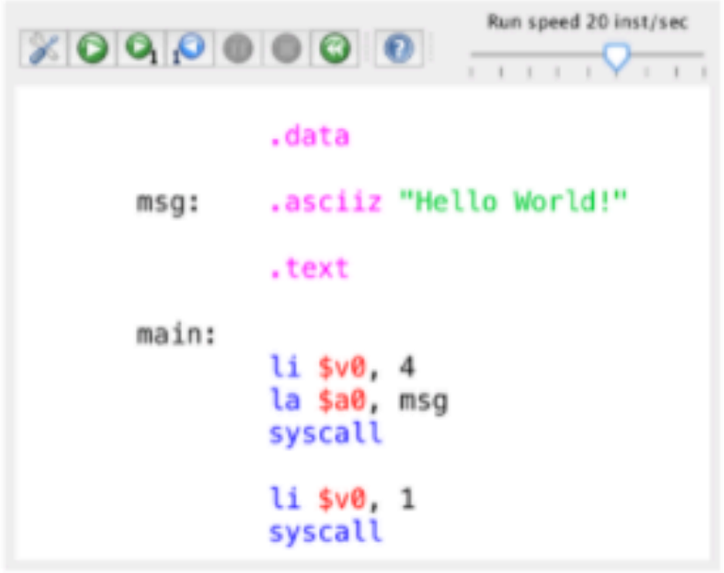
Period 3				Modul 0			Modul 1		
W	D	Date	Bivillkor	OS	OSPP	DSP	OS	OSPP	DSP
4	Mo	2020-01-20		L (KM + LÅ)					
	Mo	2020-01-20			Tm				
	Tu	2020-01-21		Tm		Tm	L (KM)		
	We	2020-01-22						W	
	Th	2020-01-23				T	W	T	W
	Fr	2020-01-24					T		T
5	Mo	2020-01-27		Tc		H		T	L (LÅ)
	Tu	2020-01-28			Tc		T	S	T
	We	2020-01-29					S	C	S
	Th	2020-01-30					C		C
	Fr	2020-01-31					Tr	Tr	
6	Mo	2020-02-03							
	Tu	2020-02-04							T + Tr
	We	2020-02-05					Sr		
	Th	2020-02-06						Sr	Sr+Cr+Ch
	Fr	2020-02-07					Cr + Ch		H
	Mo	2020-02-10						Cr + Ch	

Module 0

Module 0

OS and OSPP

Course overview and preparations.




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

int main (void) {
  printf("Hello, World!\n");
  exit(0);
}
```

To prepare for the tutorials and programming assignments you should make sure to go through the material in this section.

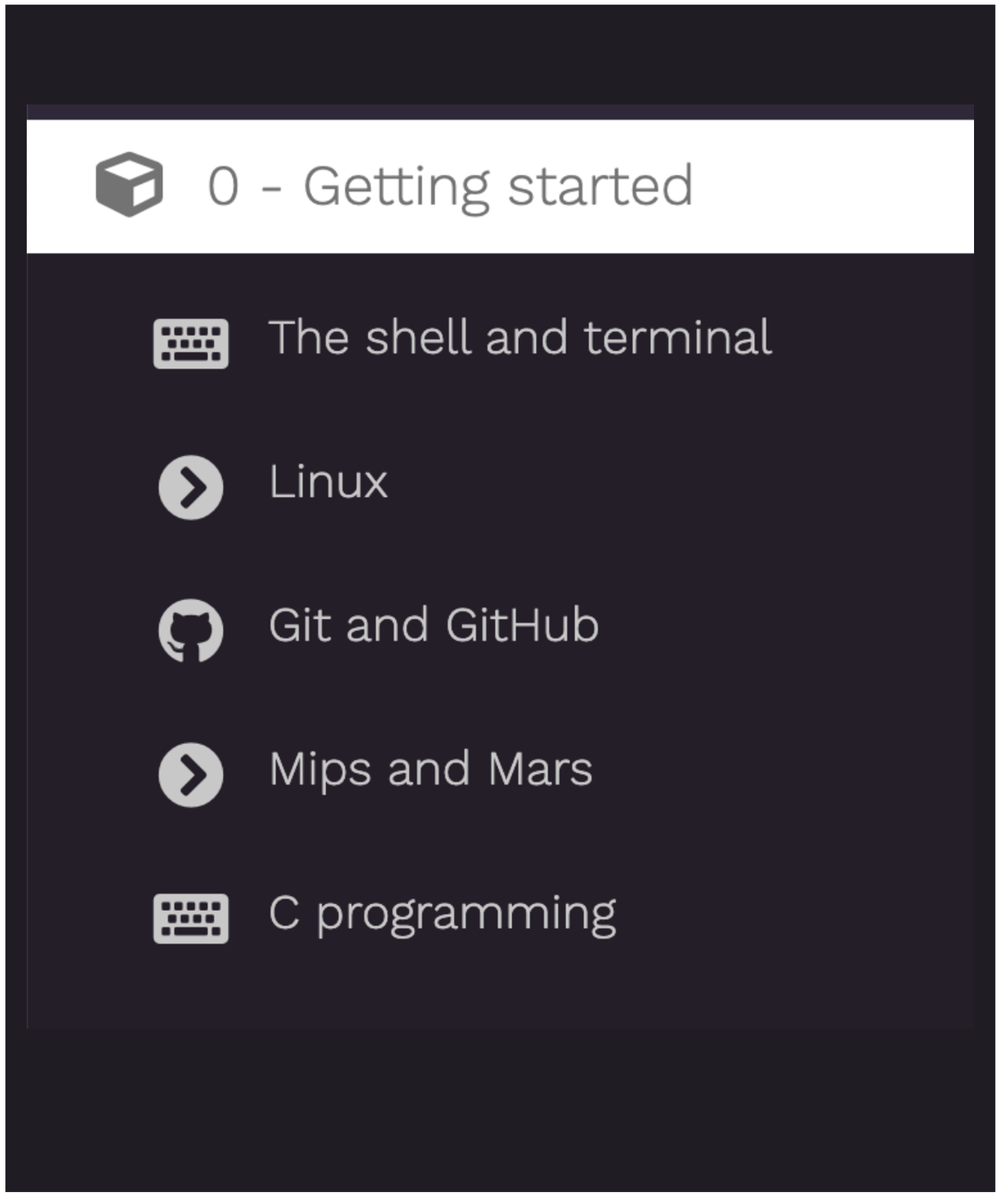
Module 0

DSP

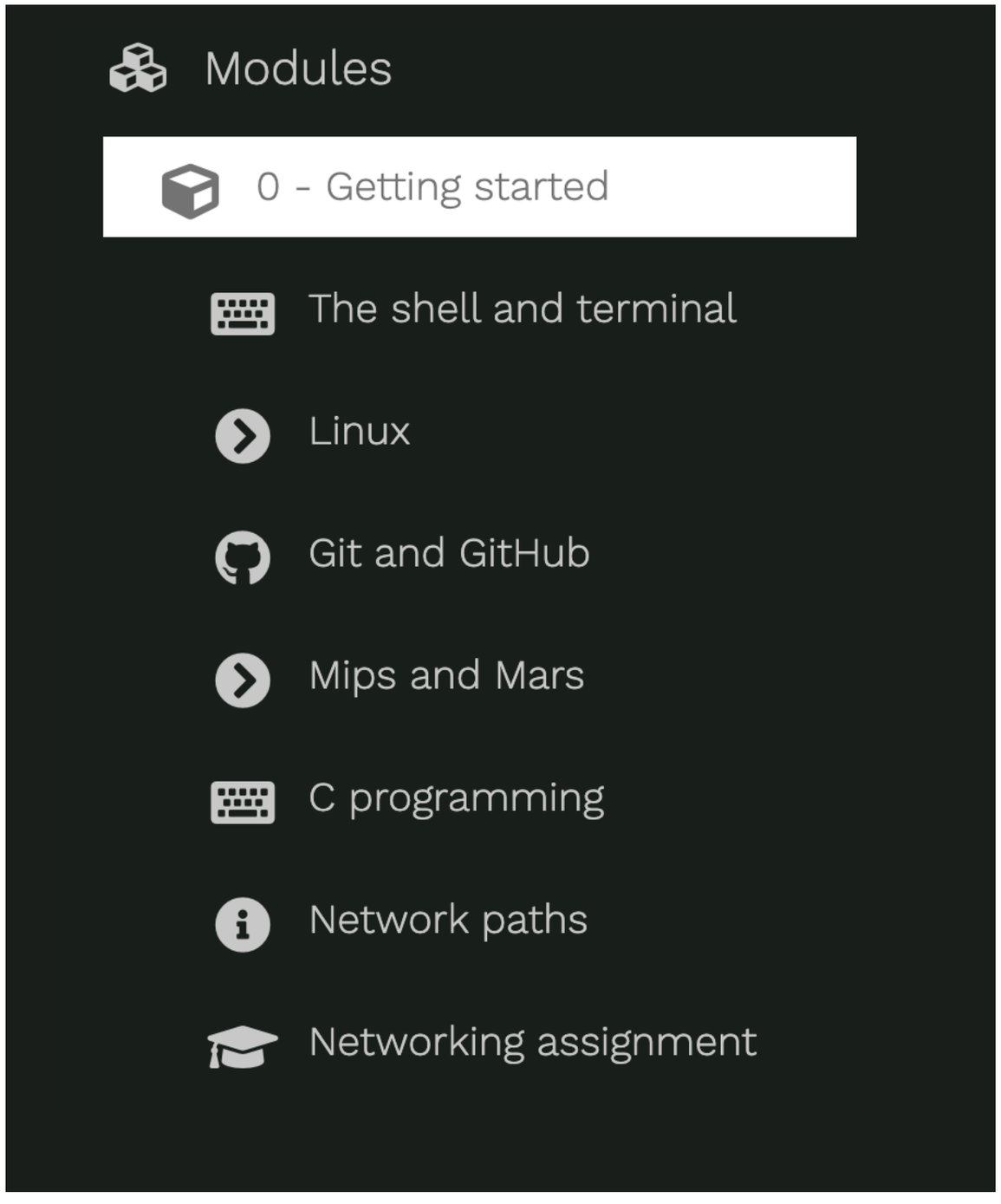
Course overview and preparations, including a network diagnostics assignment.

To prepare for the tutorials and programming assignments you should make sure to go through the material in this section. At the end of the of the module, there is an assignment related to simple network diagnostics.

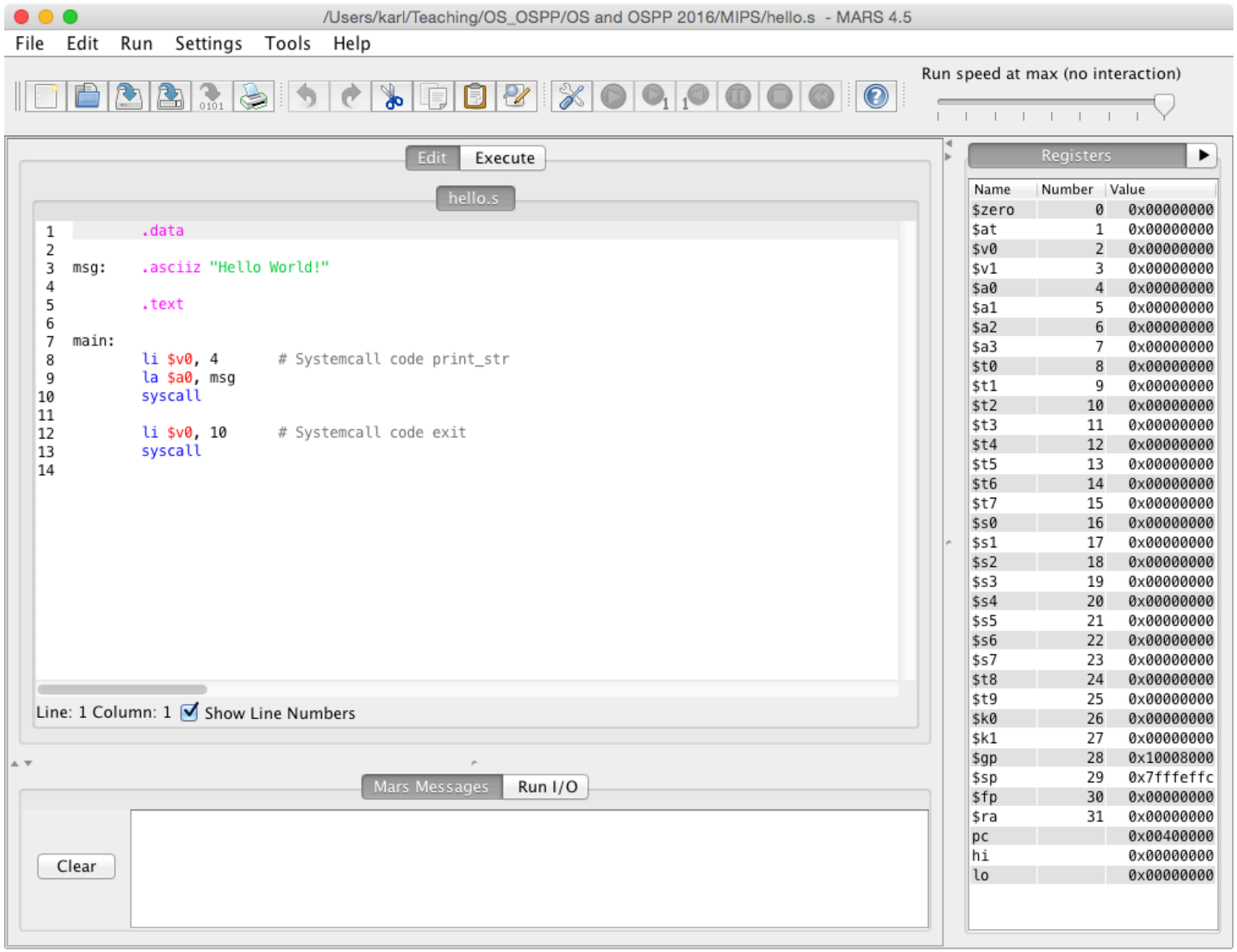
OS/OSPP


0 - Getting started
The shell and terminal
Linux
Git and GitHub
Mips and Mars
C programming

DSP


Modules
0 - Getting started
The shell and terminal
Linux
Git and GitHub
Mips and Mars
C programming
Network paths
Networking assignment

Mips and Mars



Mips and Mars

In order to study how the operating system interacts with the hardware, **Mips** assembly will be used.

To edit and execute Mips assembly programs we will use **Mars** (Mips Assembler and Runtime Simulator). Mars is available on the department Linux System.

Mars will run on any system (including Windows) as long as you have **Java installed**. If you prefer, you may **download** and install Mars on your private computer.

Mips memory layout

Clone repository

Introduction to Mars

Mips assembly examples

C programming

To study operating system concepts we will use the C programming language.

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


int main (void) {
   printf("Hello, World!\n");
   exit(0);
}
```

C programming

Important concepts

Learning resources

Programming exercise

Important concepts

To study operating system concepts we will use the C programming language.

To be able to understand the tutorials and solve the programming assignments you will need to be familiar with the following C programming concepts.

Important concepts (1)

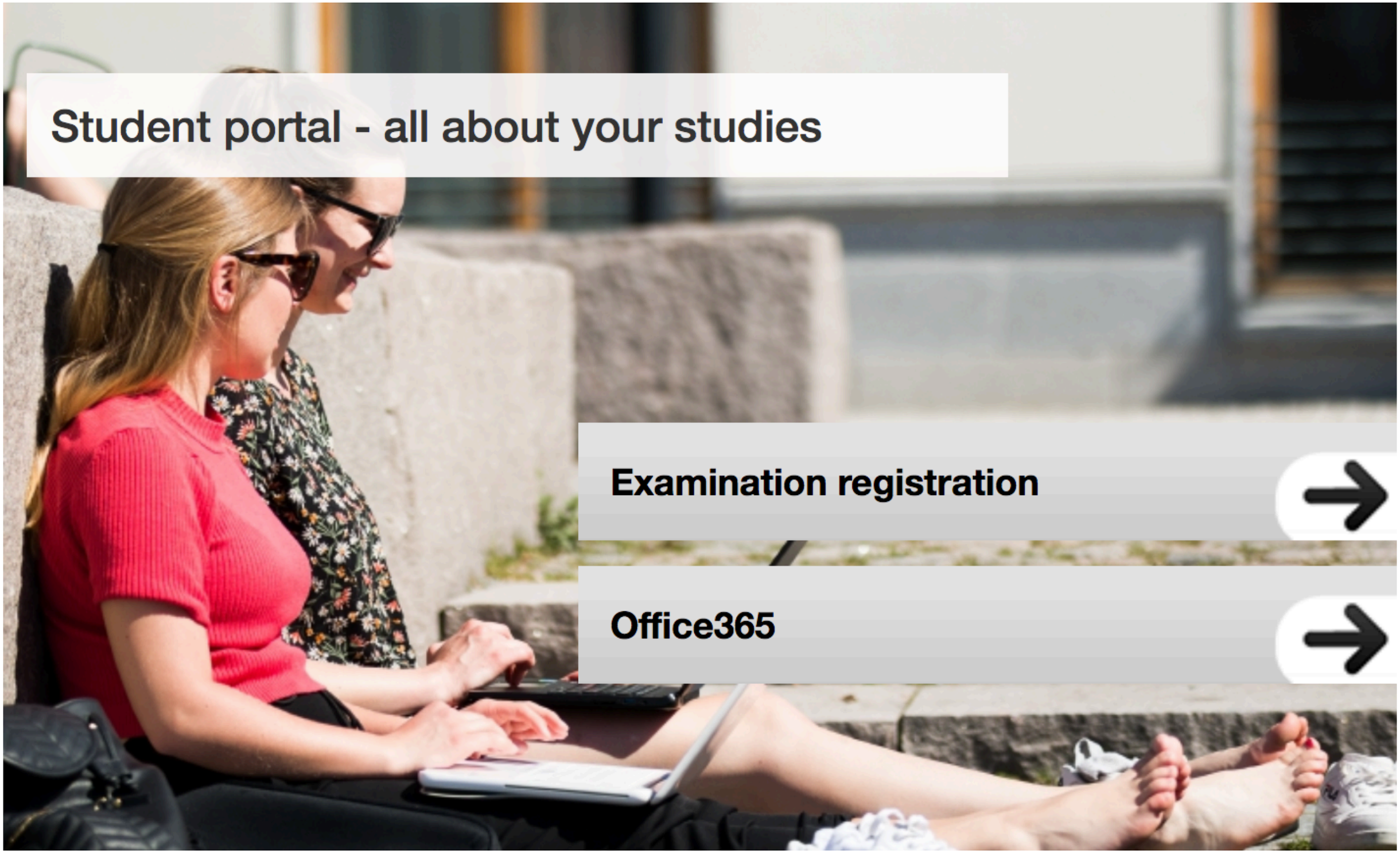
- Basic data types such as `int` and `char`.
- Arrays.
- Strings (array of `char`).
- Basic use of the `#define` directive.
- Basic use of `printf()` to print text to the terminal.
- The `if-then-else` control structure.
- The `switch` control structure.
- The `while` and `for` loops.

Important concepts (2)

- Functions and function calls.
- Pointers.
- Structures.
- Pointers to structures.
- Call by reference.
- Dynamic memory allocation (malloc and free).
- Header files.
- Separate compilation.

Formalities

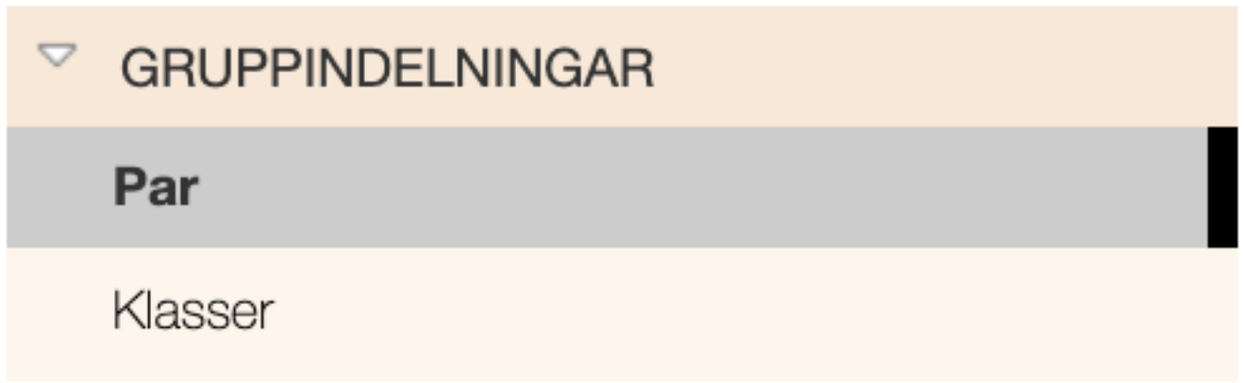
www.studentportalen.uu.se



Pairs and study groups

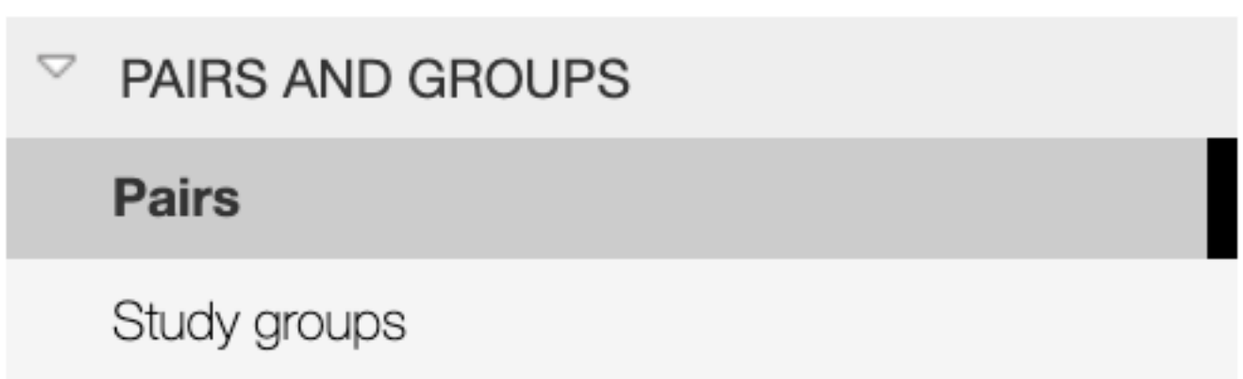
Sign up for a coding pair in the **Student portal.**

DSP (1DT003)



Deadline: 2020-01-20 18:01

OS (1DT044) and OSPP (1DT096)



Deadline: 2020-01-21 10:00

Pairs and study groups

For **programming assignments** you will work in **pairs** of two students.

- You may choose to create a pair with another student by **signing up** for a pair - please pick the available pair with the lowest number.
- If you want to be assigned to a random pair by the teaching staff - sign up to the group **randomized**.
- If you **dropped** the course, **retake** the course but don't need to belong to appear etc - sign up to the group best describing your situation.
- **Deadlines**
  - 2020-01-20 18:01 (DSP)
  - 2020-01-21 10:00 (OS and OSPP)
- During **workshops, seminars** and the **case study** you will work in **study groups** of six students. The study groups will be formed by the teaching staff by randomly grouping 3 pairs together.

www.studentportalen.uu.se

HOW DO I REGISTER?

Information about when, where and how the student is to register is found on the course page in Student Portal.

As a new student, you must:

1. **Activate your student account**
2. **Log in to the Student Portal**
   Here you see information on what courses you are admitted to and how to register.
3. **Register for your course**
   If you do not register, you may lose your place!
4. **Get Campus card**

Tip! Information concerning how to register can be found by searching for the name or application code for the course via **Search** in the Student Portal.

Are you registered?

Admitted students

Please web register in the student portal. Your registration will be transferred automatically to Ladok. The web registration will close one week after course start.

Students who are unable to web-register, should contact the Student Office (IT-kansliet): **it-kansli@it.uu.se**

Information for students admitted with conditions

Please web register in the student portal. Your registration will be transferred automatically to Ladok.

The student counsellors will handle all conditions in the students' admissions.

Non Admitted student

If you are on the waiting list we will contact you if there are seats available. Please contact us for help with registration. **it-kansli@it.uu.se**

Exceptions:

- **Master students** should contact their programme counsellor (Liselott Dominicus): **studievagledare@it.uu.se**.
- **Exchange students** should contact Ulrika Jaresund, who is the exchange student coordinator at the IT Dept: **ulrika.jaresund@it.uu.se**.
- **Students with an older registration**, who want to re-register, should contact the Student Office (IT-kansliet): **it-kansli@it.uu.se**. You can only re-register if the course is not full.
- **Students who are placed on the waiting-list**, should wait for a possible answer (if any) via e-mail from the Student Office (IT-kansliet).

You are not able to take the exam, nor have any other results reported to Uppdok if you aren't admitted and registered for the course.

Sign up for final written exam

- You must sign up for the final written exam in the Student Portal.

- The exam signing-up system will not open until 2 weeks after the course start.

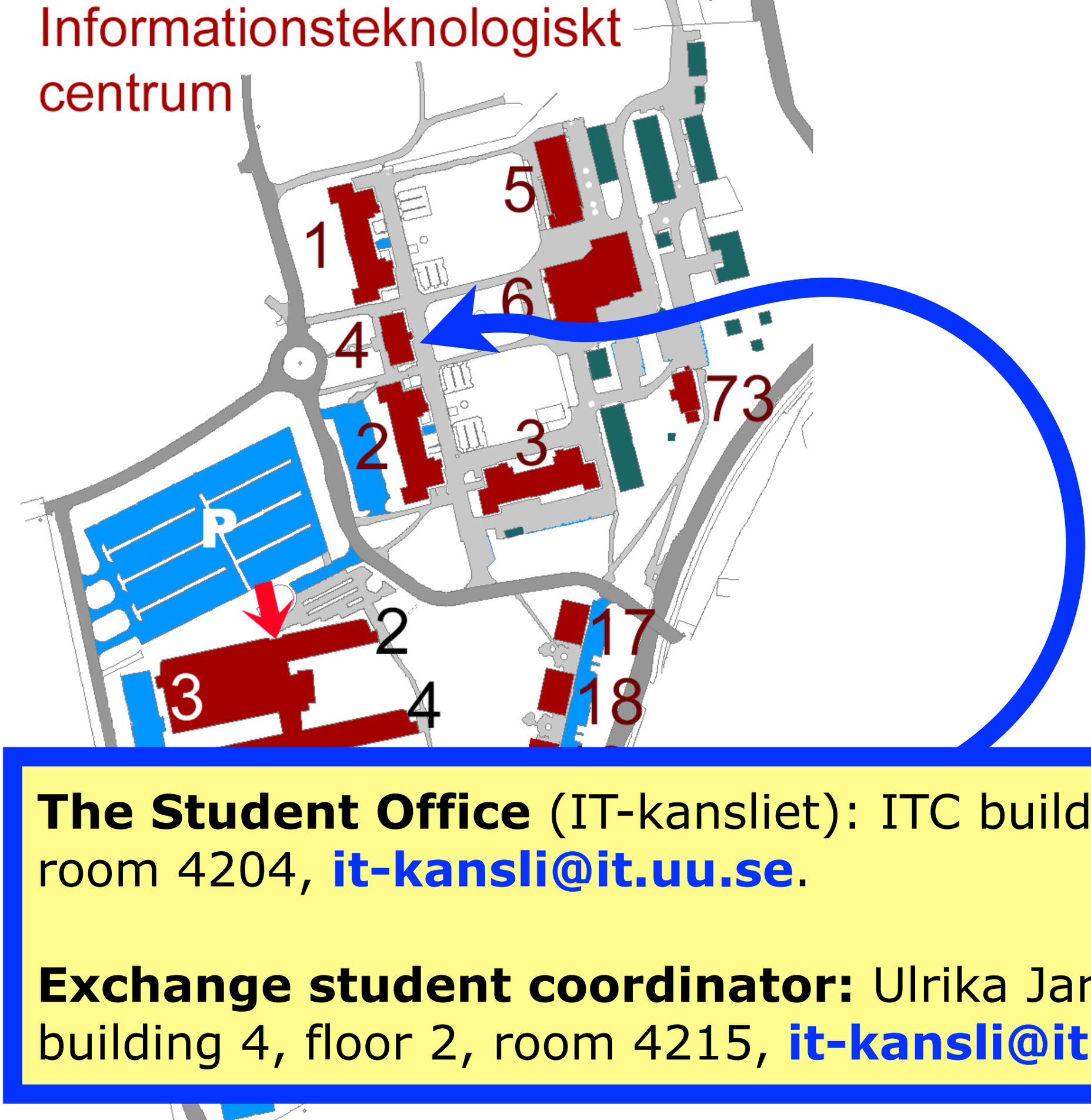
- The exam signing-up system in the Student Portal closes 12 days before the exam date.

Drop the course?

- Students who quit a course, must inform the Student Office (IT-Kansliet): it-kansli@it.uu.se

- If less than 3 weeks have passed since the course started, the course registration will be removed.

- After 3 weeks a "course intermission" will be reported to UPPDOK instead.



The Student Office (IT-kansliet): ITC building 4, floor 2, room 4204, **it-kansli@it.uu.se**.

Exchange student coordinator: Ulrika Jaresund, ITC building 4, floor 2, room 4215, **it-kansli@it.uu.se**.