Systems Programming and Computer Architecture (252-0061-00)

Exercise Session 01
Data Lab

Systems@ETH Zürich

Goal

Get familiar with bit level representations,
C and Linux

Outline

- Setting up your work environment
- Introduction to Linux
- Preview of the assignment
- Version Control (git)

Setting up your work environment

Setting up Linux

Getting started

Systems@ETH zürich

You will need a Linux compatible environment to solve the exercises.
If you don't have a computer running Linux, either:

- Use the lab machines, they are running Linux (dual boot)
- **Use the Windows Subsystem for Linux (only for Windows devices)**
- Use a Docker container (especially Apple M ARM chips)
- Install Linux in a virtual machine
- Last resort: Remote access a lab machine via ssh (maximus machines)

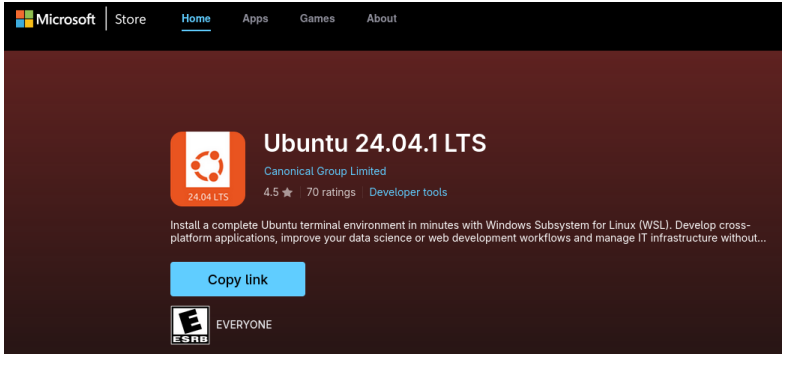
You can also setup your laptop for dual boot if you like or use Live Disks

WSL

The Windows Subsystem for Linux lets you run a GNU/Linux environment directly on Windows without the overhead of a traditional VM or dual boot setup

WSL Setup Part 1

- Install Ubuntu 24.04 LTS (Microsoft Store)

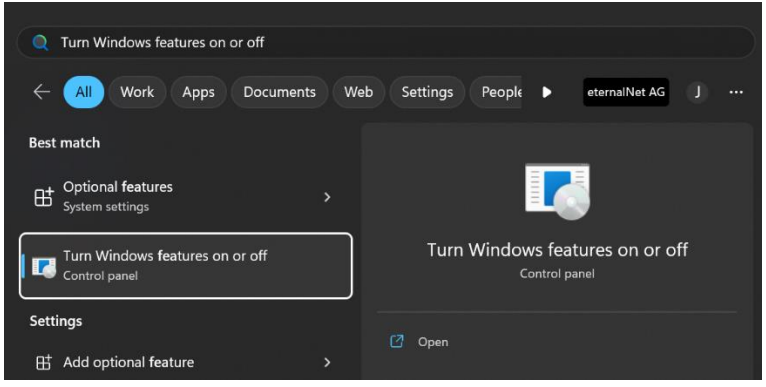


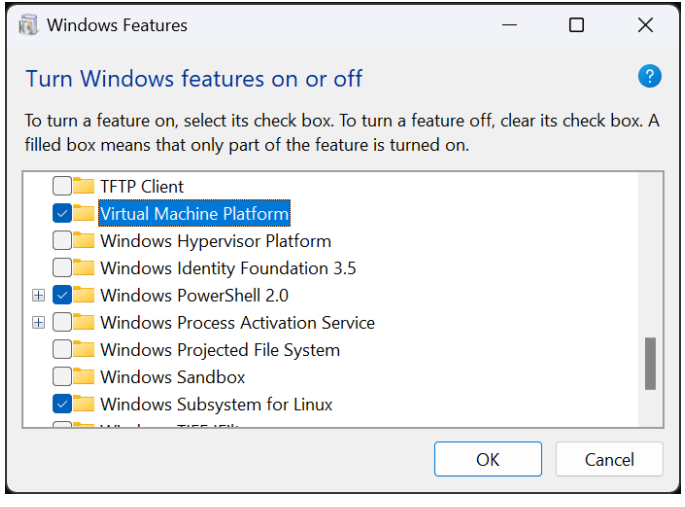
WSL

WSL Setup Part 2

- Make sure the Windows feature “Virtual Machine Platform” is enabled

- If this is not the case enable it by marking the checkbox and restart your device when asked



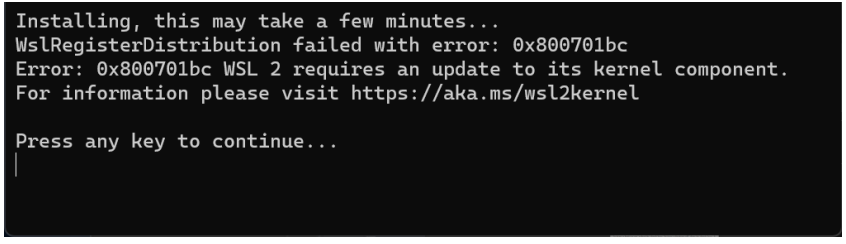


WSL

Systems@ETH zürich

WSL Setup Part 3

- Open Ubuntu 24.04
  (the one you installed in Part 1)



- You might encounter an the error message above. In this case install the kernel update. You can download it using this link:
  https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi

- After the installation enter the following command:
  wsl --set-default-version 2

WSL

Systems@ETH zürich

WSL Setup Part 4

- You should now be able to successfully start Ubuntu 24.04 and enter Linux commands (described in the following slides)

- To open the current folder with VS Code enter the command:
  code .

- If you want to access your Windows files you can enter
  cd /mnt/c (c is the Windows drive letter)
  Only use this if you really have to, since it reduces performance!

Docker container

Systems@ETH zürich

This should work for everyone, especially new ARM MACs.

- Install Docker
- Look at the Moodle 'Additional material' secion for the docker container file

Alternative Solution: Virtual machine

Systems@ETH zürich

1. Download VirtualBox https://www.virtualbox.org/
2. Install VirtualBox on your machine
3. Obtain a copy of Ubuntu **24.04 LTS** http://www.ubuntu.com/
4. Create a new machine and install Ubuntu on it. https://docs.oracle.com/cd/E26217_01/E26796/html/qs-create-vm.html

Last Resort

Using ssh: maximus.inf.ethz.ch

Systems@ETH zürich

Every student of D-INFK can log in to *maximus.inf.ethz.ch*, which has the same Linux setup as the student labs.

https://www.isg.inf.ethz.ch/Main/HelpRemoteAccessSSH

We can access it using the **secure shell protocol**

- SSH creates a secure connection from one device to another (often over the terminal) which allows one to execute commands on the other device
- Use this only as a last resort, as the maximus system is small and not capable of handling many students at once. More info at the end of the slides.

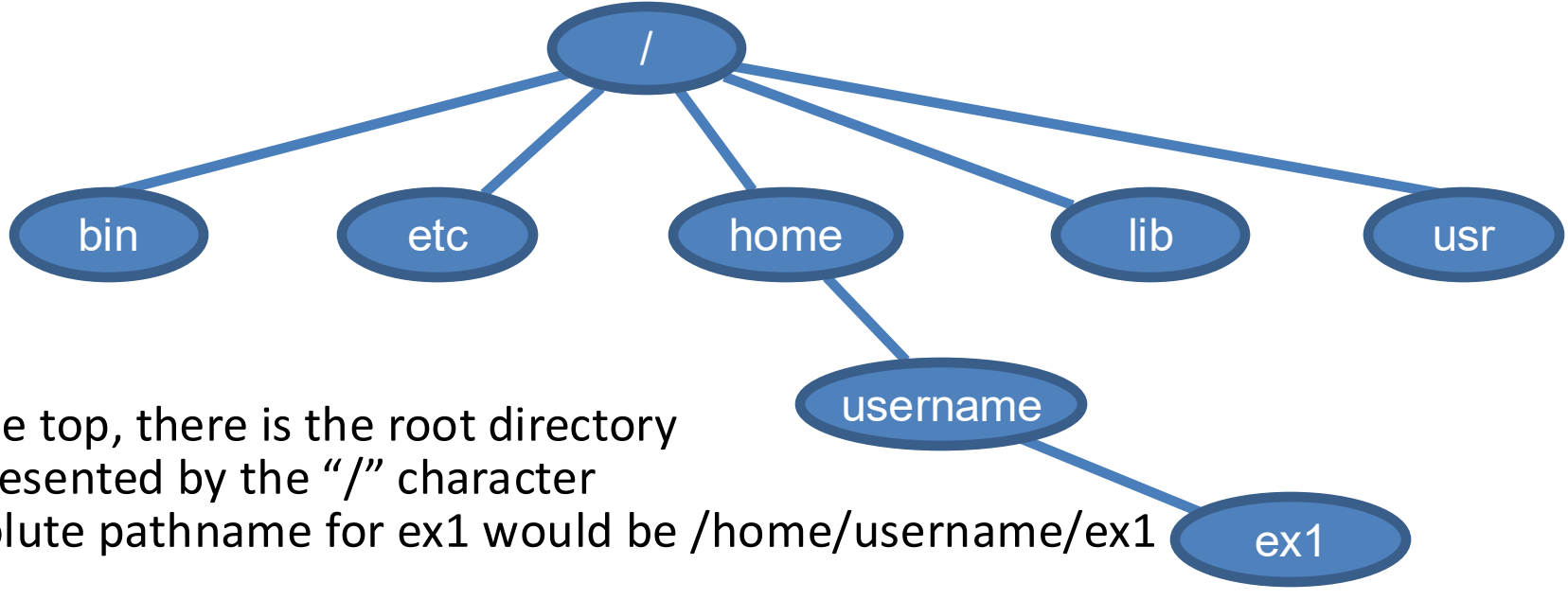
Introduction to Linux

Ubuntu 24.04 LTS

File System

Systems@ETH zürich

- UNIX organizes user data, programs, etc. into structures called files.
- Files are placed in directories.
- Directories are organized into a hierarchical structure.



- At the top, there is the root directory
- Represented by the “/” character
- Absolute pathname for ex1 would be /home/username/ex1

Browsing the Filesystem

Systems@ETH zürich

- **whoami**: prints the login name of the current user

- **pwd**: prints the working directory

- **ls**: lists files and directories
  - Has more options such as –F, -a, -l, -all.

- **cd**: changes the current working directory to the given pathname
- e.g.: cd /home/username/ex1

- **“.”** is the current directory and **“..”** stands for the parent directory, both can be used with cd
- **“~”** stands for your home directory

Browsing the Filesystem

Systems@ETH zürich

- **mkdir**: creates a directory
  - mkdir /home/username/ex1/newfolder
- **rmdir**: removes a directory
  - will only remove empty directories
- **cp**: copies files/folders from one location to another
  - cp /etc/hosts /home/username
- **mv**: move/rename existing files/folders
  - mv /home/username/hosts /home/username/ex1/newfolder
- **rm**: removes files/folders
  - rm /home/username/ex1/newfolder/hosts

Processes

Systems@ETH zürich

- **ps**: see the processes associated with the current shell
  - ps -ef to get a full listing of all processes in the system

- **top**: display the processes using the most CPU time
  - Quit with q

- **kill**: terminates a process
  - Used as ‘kill <ProcessID>’.
  - -9 option to force kill

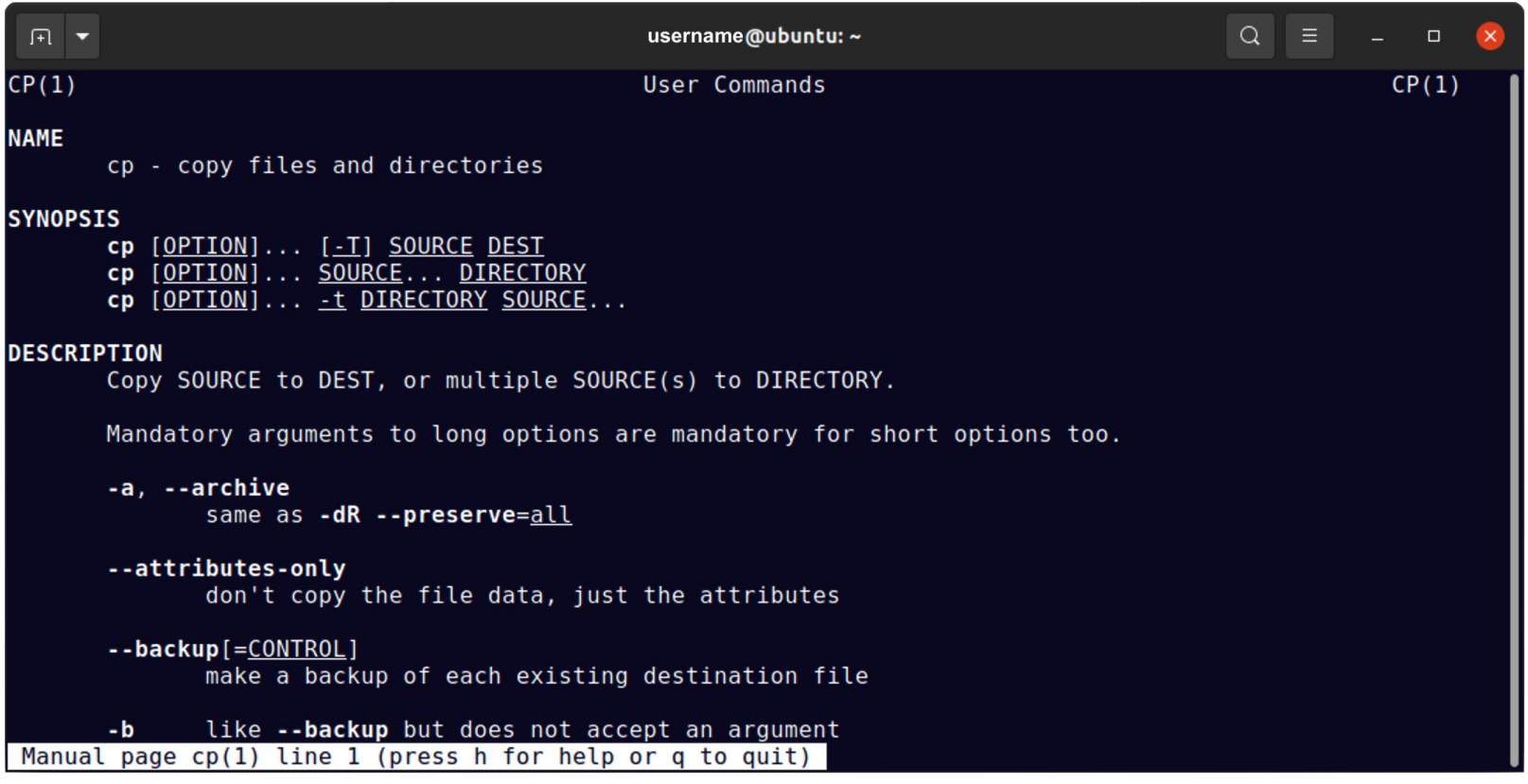
Miscellaneous

Systems@ETH zürich

- **nano, gedit, emacs, vi/vim**: useful text editors for writing your programs and editing files.
- **cat, more, less**: useful to view files
- **grep**: useful for searching text files
- **gcc/gdb:** compilers and debuggers

Lost? Try “man”.

man cp

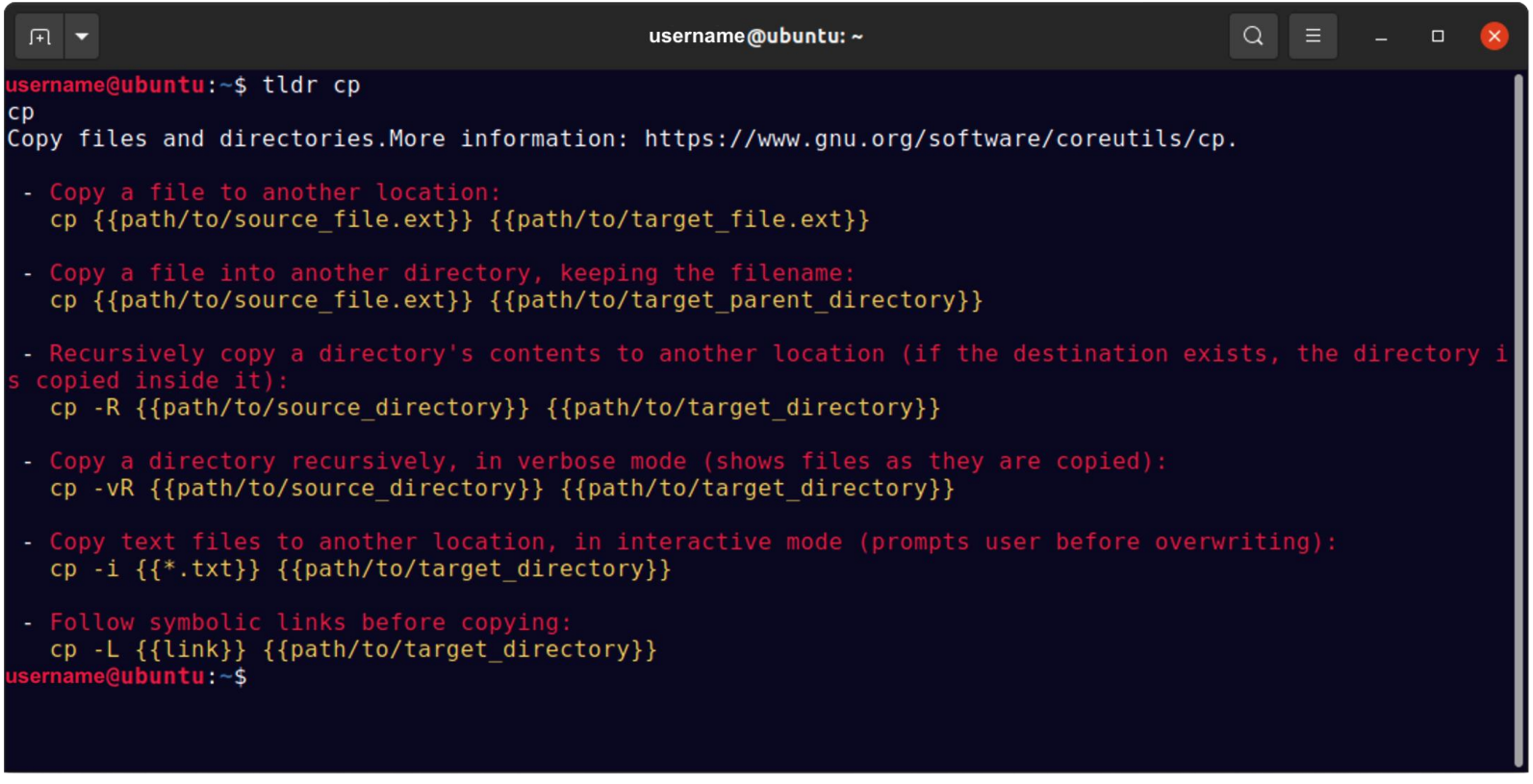


Still lost? Try “tldr”.

Systems@ETH zürich

Can be installed with sudo apt install tldr

tldr cp



More tutorials online

Systems@ETH zürich

- http://people.ischool.berkeley.edu/~kevin/unix-tutorial/toc.html
- http://www.ee.surrey.ac.uk/Teaching/Unix/
- http://www.unixtutorial.org/commands/
- … just Google/ChatGPT for more!
- A lot to take in, but it will become second nature over time :)

Preview of Assignment 1

The Data Lab

Pre-requisites

Systems@ETH zürich

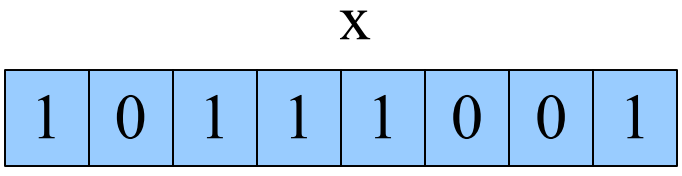
- You will need a working Linux environment
  - If you just installed Ubuntu on a VM, you still need to install some tools (gcc, etc.)

```
$ sudo apt update
$ sudo apt install build-essential
$ sudo apt install flex bison
```

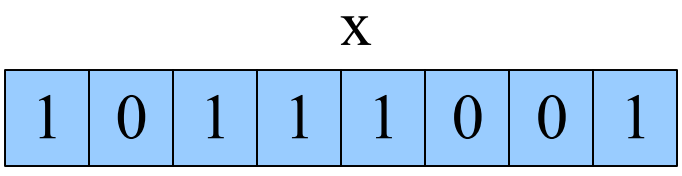
- Download the assignment sheet and follow the instructions carefully.
- All you need to change is in **bits.c**

Introduction Bit-Operators in C

- Memory is organized as an array of bits
- Smallest addressable memory unit: byte
- The type of a variable determines its value
- e.g.: integers are represented with two's complement



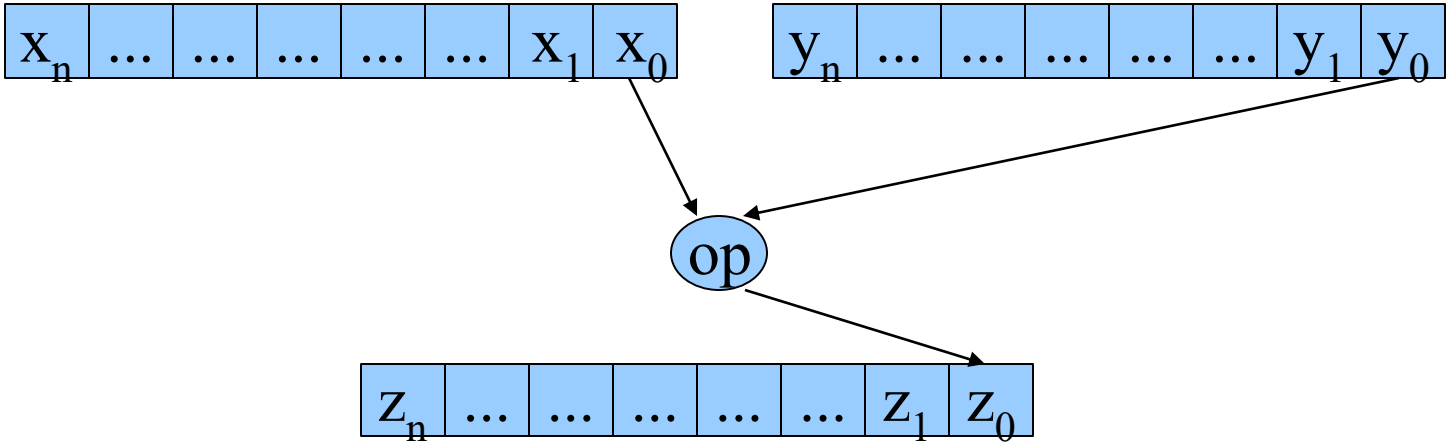
signed char x = -71



unsigned char x = 185

Introduction Bit-Operators in C

- Bitwise operations are performed on every bit of the two operands individually
- Can be applied to any “integral” datatype
- $Z = X \text{ op } Y \quad \rightarrow \quad Z_i = X_i \text{ op } Y_i$



Logical vs Bitwise Operators

Systems@ETH zürich

- **Logical operators** evaluate the truth or falsity of **an expression**
  - The result is either **true or false**
  - In C: 0 is false, **anything else** is true
  - Logical AND: && Logical OR: || Logical NOT: !

- **Bit operators** perform the operation on **each bit**
- The result can be an **arbitrary** value
  - Bit-wise AND: & Bit-wise OR: | Bit-wise NOT: ~

Bit Masks

- Used to set/delete/test single bits
  - **Delete** and test bits with AND
  - **Set** bits with OR
  - **Flip** bits with XOR
- Example: x is either '0' or '1'

```
  xxxxxxxx      xxxxxxxx      01101001
& 01010101    | 01010101    ^ 01010101
  --------      --------      --------
  0x0x0x0x      x1x1x1x1      00111100
```

Bit Masks

- Test if i-th bit is 1
  - result = (input & (1<<i))
- Flip i-th bit
  - result = (input ^ (1<<i))
- Set i-th bit
  - result = (input | (1<<i))

Shift Operators

- Right Shift *"Division by a power of two"*
  - Logical (Java: >>>): fill left-end with 0's, used with unsigned types
  - Arithmetic (Java: >>): fill left-end with MSB, used with signed types
    WARNING: not all compilers do arithmetic shift with signed types, thus shift with signed types considered to be **UNDEFINED.**

- Left Shift *"Multiplication by a power of two"*

```
x = 0b0011;      // x = 3
z = x << 2;      // z = 0b001100 = 12 = 3 * 2^2
```

Your Turn! Do the homework

Systems@ETH zürich

- Complete function skeletons in **bits.c**
- **Restrictions**
  - **No loops, conditions or jumps**
  - **Use the following operators only: ! ~ & ^ | + << >>**
  - **Constants must not be longer than 8 bits**

- **Contest**: “Beat the professor”

- **Goal**: Use as few operations as possible

Example

- Return the min. value *Tmin* of a signed integer

Example

- Return the min. value *Tmin* of a signed integer
- Tmin is 0x80000000
- Idea: shift 1 31 positions to the left

```
int Tmin() {
    return (1 << 31);
}
```

- Note: return (0x80000000); is not legal, since constants must not be longer than 1 byte!

Systems@ETH zürich

Version Control using git

How to submit your solution

Preparation

- You will need to install git and ssh:

  $ sudo apt install git openssh-client

- You will need to generate and put your SSH key to gitlab and clone your repo.
  (Instructions also in assignment1).

Tell git about you

```
$ git config --global user.name "Jane Doe"
$ git config --global user.email "jdoe@student.ethz.ch"
```

Generate an SSH key pair

Systems@ETH zürich

- If you haven't used ssh before, generate a new key
  $ ssh-keygen

- Confirm defaults with enter three times (or use a passphrase). Then display your public key
  $ cat .ssh/id_rsa.pub
  ssh-rsa AAAAB3NzaC1yc2EAAAADAQ…

- Copy the key (in the terminal, copy/paste with ctrl-c/ctrl-v doesn't work. Select the text and use right-click, copy)

Upload SSH key to gitlab

Systems@ETH zürich

- Open https://gitlab.inf.ethz.ch/-/user_settings/ssh_keys
- Login with your nethz credentials
- Paste your key and save

Checkout your repository

(replace the placeholder NETHZ below with your NETHZ)

Systems@ETH zürich

- Clone your repository

  $ git clone git@gitlab.inf.ethz.ch:course-spca2025/spca2025-NETHZ-hand-in.git

- This will create a folder “spca2025-NETHZ-hand-in”

  $ cd spca2025-NETHZ-hand-in

Submitting your solution

Systems@ETH zürich

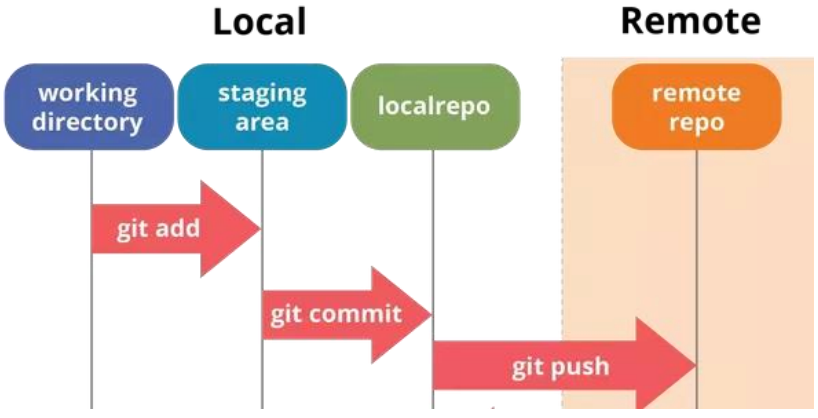
- You need to copy the file bits.c into your git repository

- Make a new directory and copy your solution into it
  $ mkdir assignment1
  $ cp bits.c assignment1

- Add, commit and push
  $ git add bits.c
  $ git commit –m “assignment1”
  $ git push

Add, commit, push?

- add
  Add to staging area
- commit
  turn staging area into a commit
- push
  push commit(s) to the server



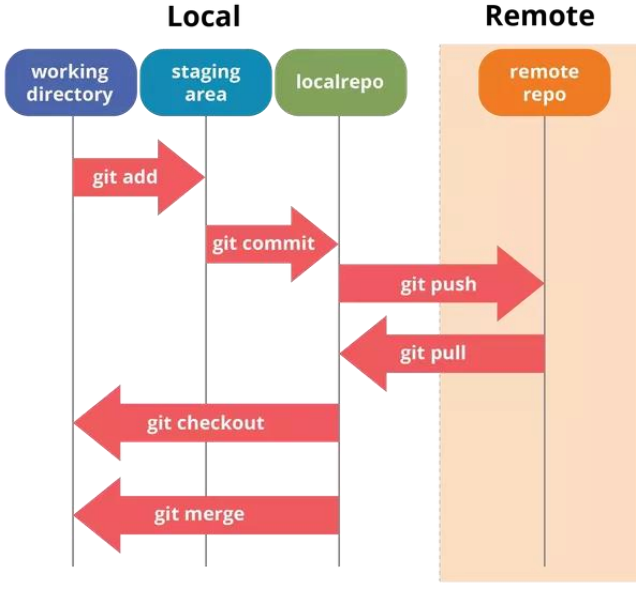
- Commits = savegame
- add and commit do not do any network access

What if push fails?

Should not happen in this assignment

Systems@ETH Zürich

- Probably the server has a more recent version than you (somebody else pushed a newer commit)
- To get new commits from the server
  $ git pull
- ***If*** there are no conflicts, you're done!
  $ git push



Submitting your solution

- You can repeat these steps to update your solution

- Check your score (only from ETH network)
  -> http://spca.ethz.ch

Last but not least: Code Expert

Systems@ETH zürich

Enroll in your TA session with this link:

https://expert.ethz.ch/enroll/AS25/spca

Appendix:

Systems@ETH zürich

If everything is going wrong and you can't come to CAB to use a lab machine, you can use a remote lab machine. The following slides explain how to easily use them.

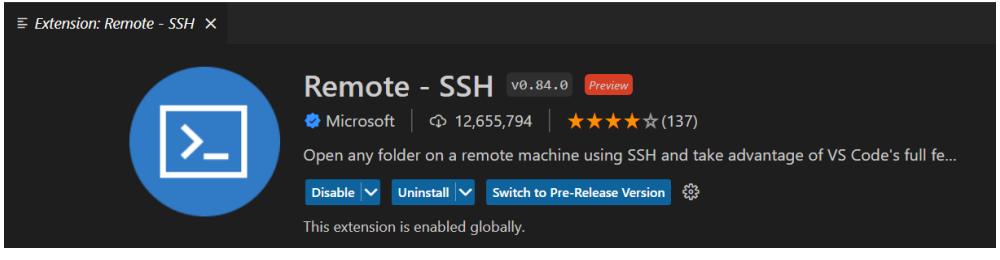
Using ssh: maximus.inf.ethz.ch

Systems@ETH zürich

VS Code Remote Setup Part 1

Install VS Code

- Install Remote SSH Extension (in VS Code)

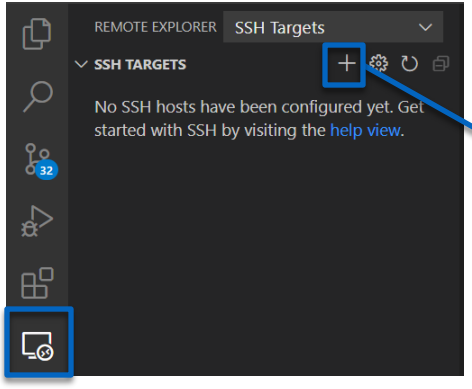


Using ssh: maximus.inf.ethz.ch

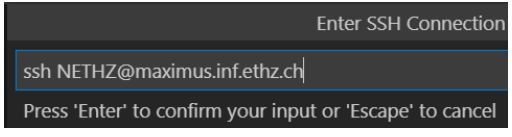
Systems@ETH zürich

VS Code Remote Setup Part 2

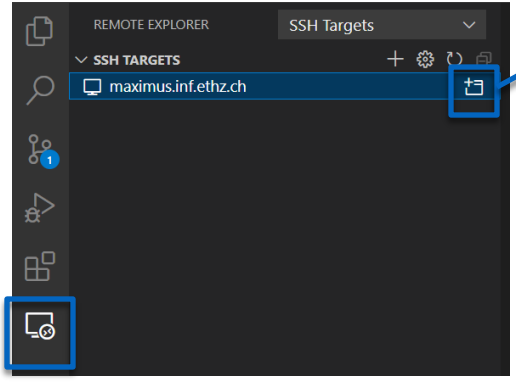
Add a new SSH Target



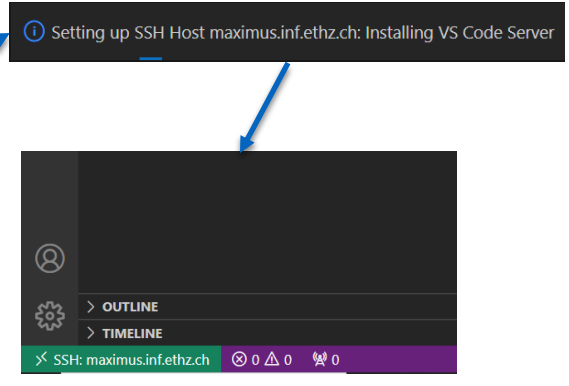
Replace NETHZ



- Connect to SSH Target



Login and Wait



Using ssh: config files and ssh keys

Systems@ETH zürich

- To reduce password prompting you can setup your ssh config file with a ssh key [optional but highly recommended]

Step 1: Key generation

- Open terminal/powershell
- Enter $ ssh-keygen into the terminal (without the dollar sign) and follow the prompts to generate your key
  - check if key pair is in ~/.ssh directory, else move it there
  - On Windows: ~/ corresponds to C:\Users\YOUR_USERNAME

Using a config file and ssh keys

Systems@ETH zürich

Step 2: move key to maximus

- Unix/MacOS:

  $ ssh-copy-id -i ~/.ssh/nameofkey.pub NETHZ@maximus.inf.ethz.ch
- Windows:
  - cat nameofkey.pub -> copy output
  - Connect to maximus and paste into ~/.ssh/authorized_keys
- Test if key was added successfully by running

$ ssh –i ~/.ssh/nameofkey NETHZ@maximus.inf.ethz.ch

Step 3: config file

- Open your config file under ~/.ssh/config
- Add the following lines:

```
1  Host maximus
2          HostName maximus.inf.ethz.ch
3          User NETHZ
4          IdentityFile ~/.ssh/id_rsa_nameofkey
```

Using a config file and ssh keys

Systems@ETH zürich

Step 4: SSH-ing into maximus

- If everything worked, you should now be able to ssh into maximus without having to enter your username and password every time
- Less time spent doing repetitive tasks -> more time for fun things (like SPCA) :D