Systems@ETH Zürich

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

Exercise Session 01
Data Lab

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

You will need a Linux compatible environment to solve the exercises, you can:

- Use the lab machines, they are running Linux (dualboot)
- Install Linux in a virtual machine

You can also setup your Laptop for dualboot if you like or use Live Disks

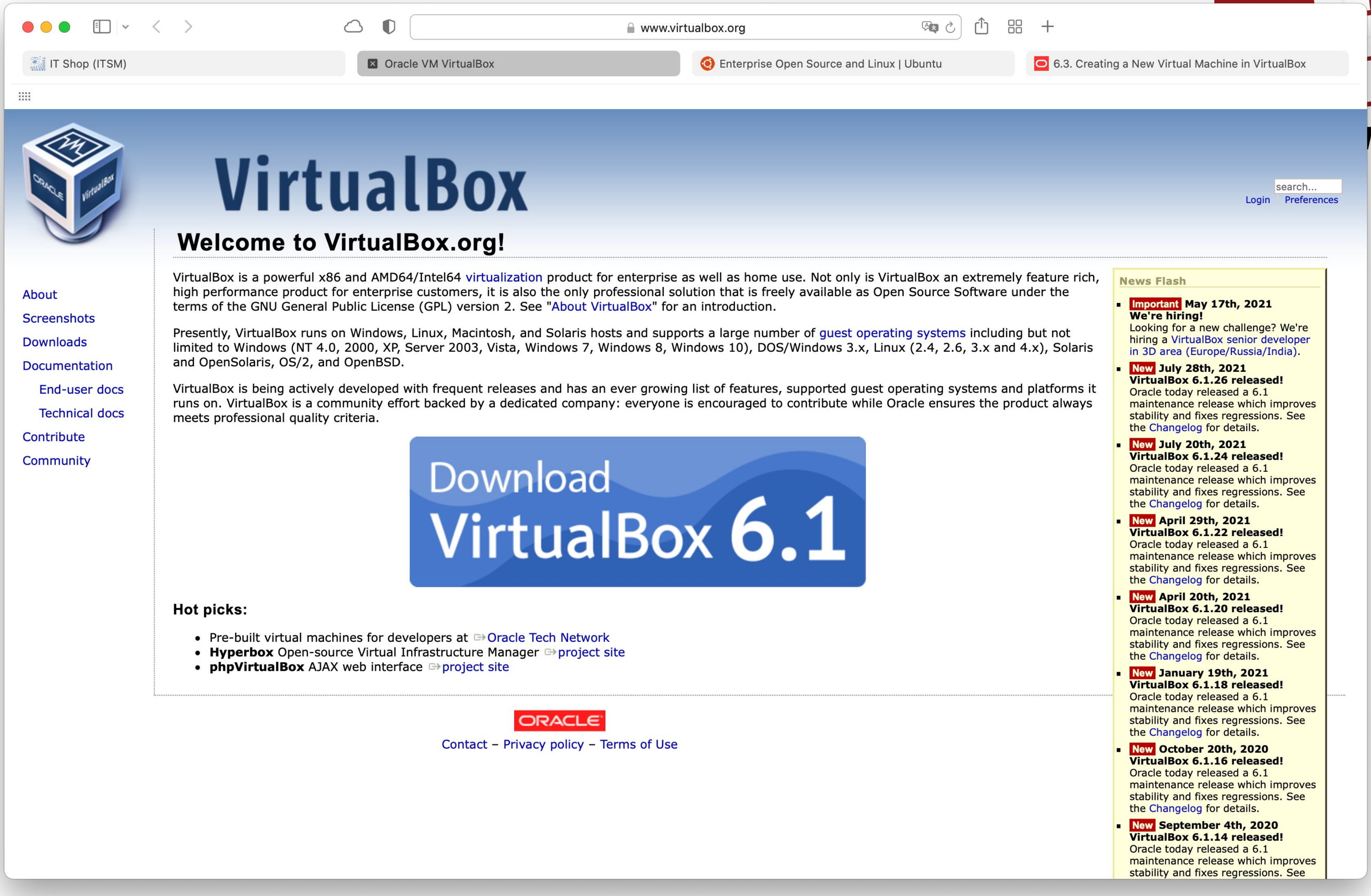
Setting up the Virtual machine

1. Download VirtualBox https://www.virtualbox.org/

2. Install VirtualBox on your machine

3. Obtain a copy of Ubuntu **22.04 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


www.virtualbox.org
IT Shop (ITSM)
Oracle VM VirtualBox
Enterprise Open Source and Linux | Ubuntu
6.3. Creating a New Virtual Machine in VirtualBox
search...
Login
Preferences

VirtualBox

Welcome to VirtualBox.org!

- About
- Screenshots
- Downloads
- Documentation
  - End-user docs
  - Technical docs
- Contribute
- Community

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "About VirtualBox" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

Download
VirtualBox 6.1

Hot picks:

- Pre-built virtual machines for developers at Oracle Tech Network
- **Hyperbox** Open-source Virtual Infrastructure Manager project site
- **phpVirtualBox** AJAX web interface project site

ORACLE

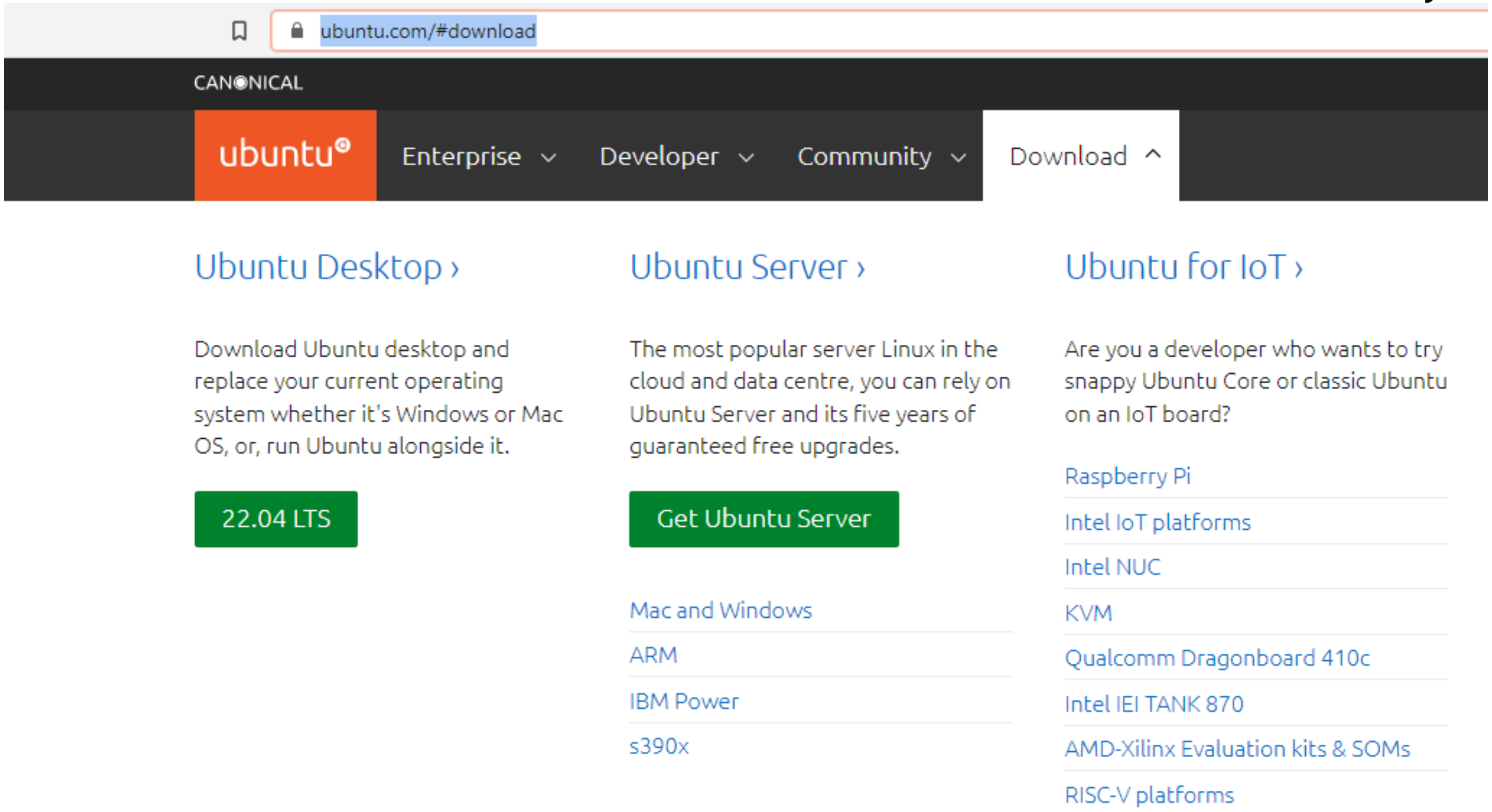
Contact – Privacy policy – Terms of Use

News Flash

- Important **May 17th, 2021**
  We're hiring!
  Looking for a new challenge? We're hiring a VirtualBox senior developer in 3D area (Europe/Russia/India).
- New **July 28th, 2021**
  VirtualBox 6.1.26 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **July 20th, 2021**
  VirtualBox 6.1.24 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **April 29th, 2021**
  VirtualBox 6.1.22 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **April 20th, 2021**
  VirtualBox 6.1.20 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **January 19th, 2021**
  VirtualBox 6.1.18 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **October 20th, 2020**
  VirtualBox 6.1.16 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **September 4th, 2020**
  VirtualBox 6.1.14 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See

Setting up the Virtual machine

1. Download VirtualBox https://www.virtualbox.org/
2. Install VirtualBox on your machine
3. Obtain a copy of Ubuntu **22.04 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



Ubuntu Desktop ›

Download Ubuntu desktop and replace your current operating system whether it's Windows or Mac OS, or, run Ubuntu alongside it.

Ubuntu Server ›

The most popular server Linux in the cloud and data centre, you can rely on Ubuntu Server and its five years of guaranteed free upgrades.

Mac and Windows

ARM

IBM Power

s390x

Ubuntu for IoT ›

Are you a developer who wants to try snappy Ubuntu Core or classic Ubuntu on an IoT board?

Raspberry Pi

Intel IoT platforms

Intel NUC

KVM

Qualcomm Dragonboard 410c

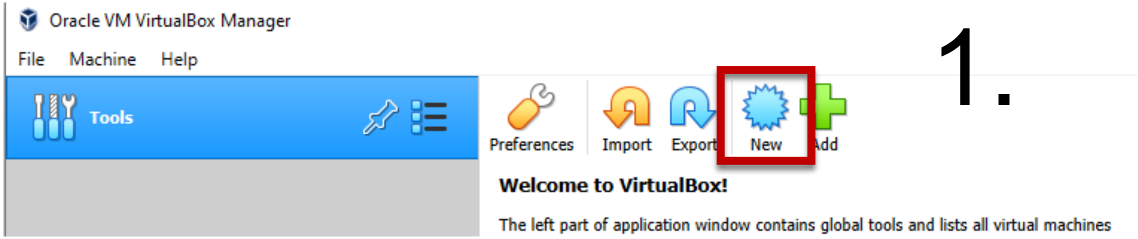
Intel IEI TANK 870

AMD-Xilinx Evaluation kits & SOMs

RISC-V platforms

Create a new VM

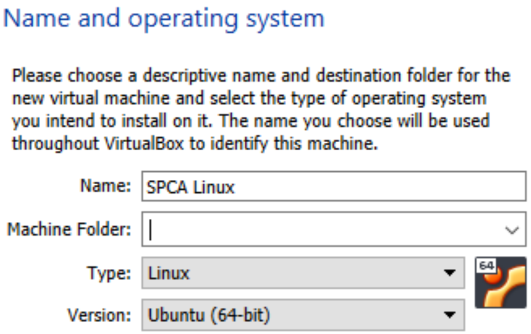
1.



Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines

2.



Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

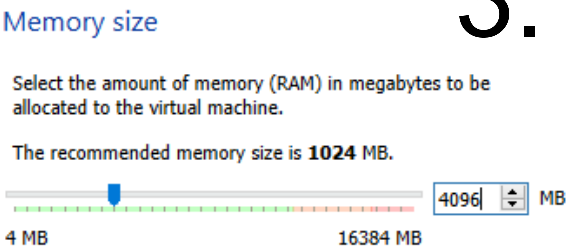
Name: SPCA Linux

Machine Folder:

Type: Linux

Version: Ubuntu (64-bit)

3.



Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

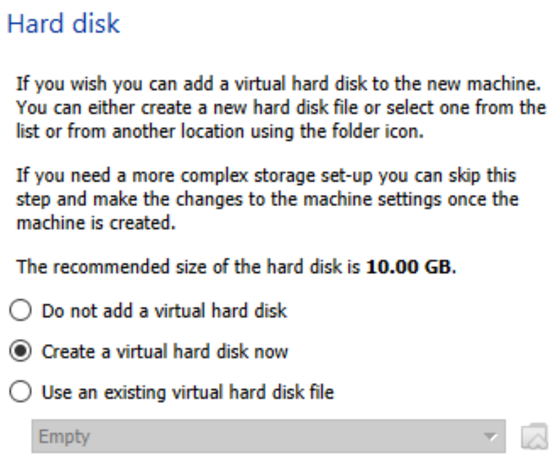
The recommended memory size is **1024** MB.

4096 MB

amount depends on how much you have available

Create a virtual hard disk

1.



Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

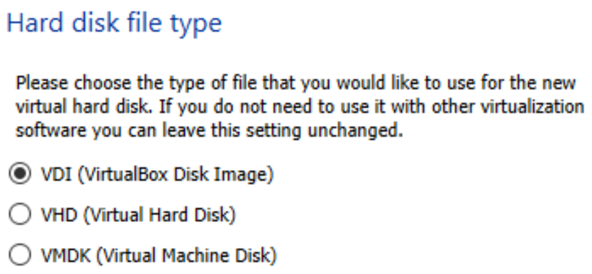
The recommended size of the hard disk is **10.00 GB**.

○ Do not add a virtual hard disk

● Create a virtual hard disk now

○ Use an existing virtual hard disk file

2.



Hard disk file type

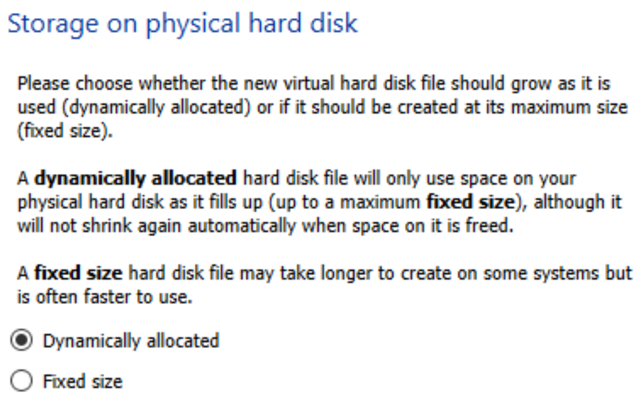
Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

● VDI (VirtualBox Disk Image)

○ VHD (Virtual Hard Disk)

○ VMDK (Virtual Machine Disk)

3.



Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

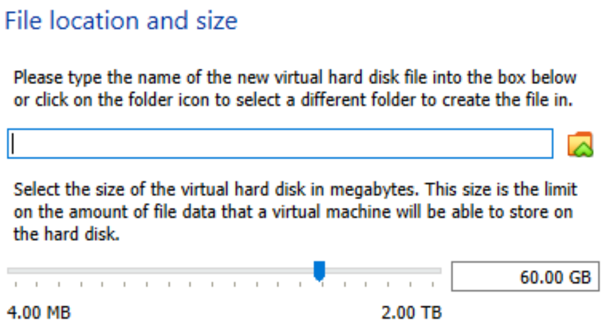
A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

● Dynamically allocated

○ Fixed size

4.



File location and size

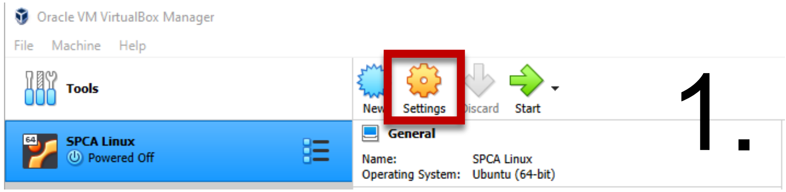
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

60.00 GB

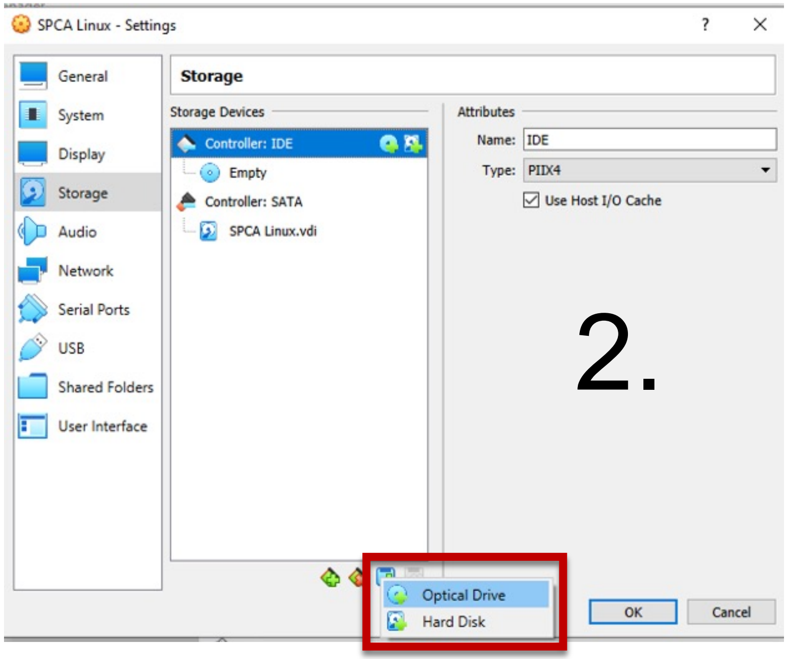
pick a location + size

Create a new VM: Setting the boot media



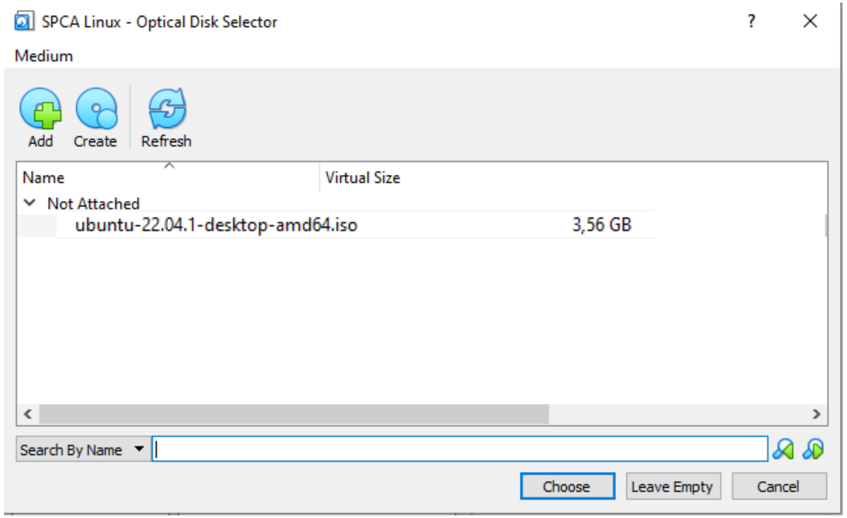
Name: SPCA Linux
Operating System: Ubuntu (64-bit)

1.



Name: IDE
Type: PIIX4
☑ Use Host I/O Cache

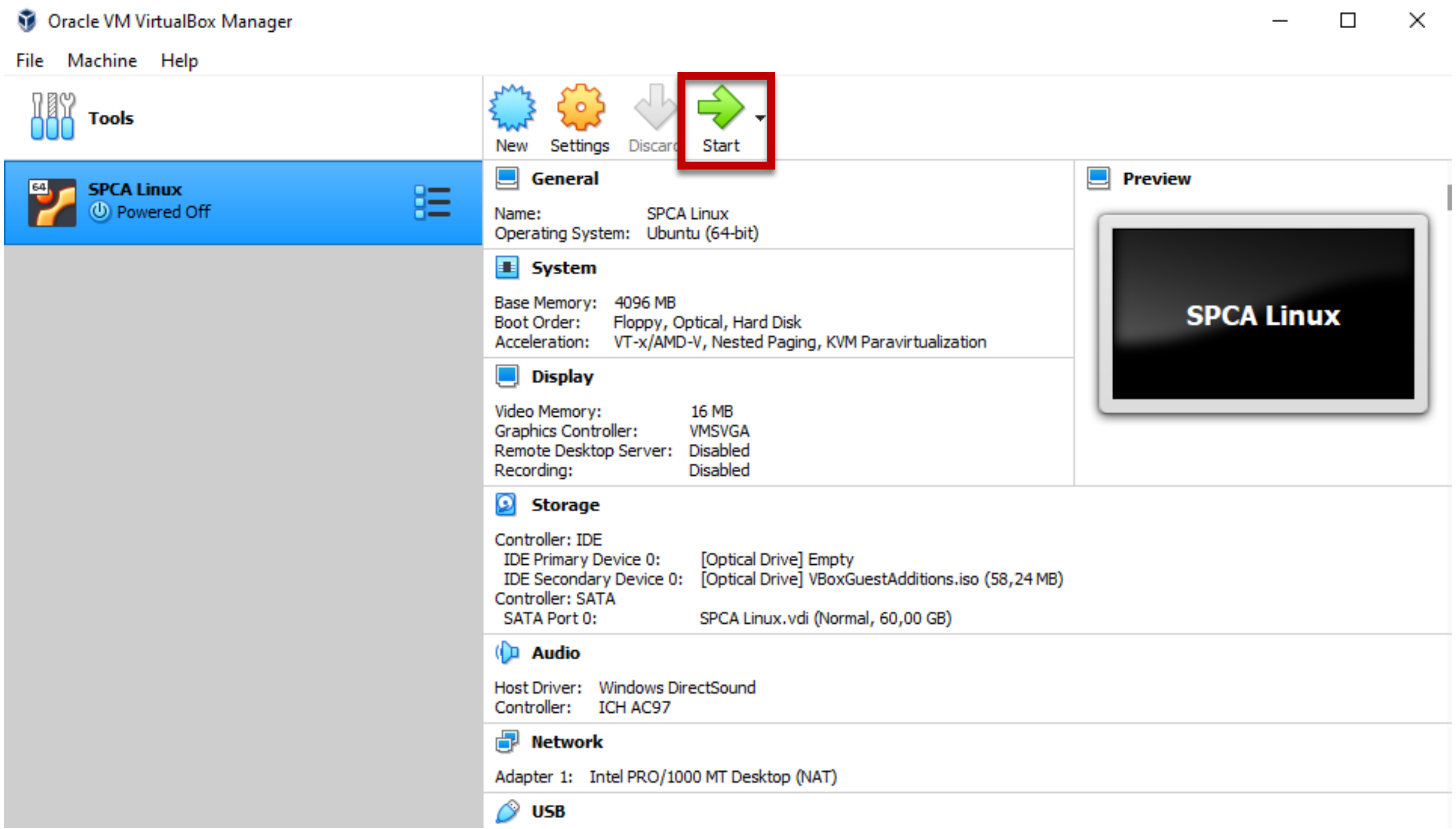
2.



Name	Virtual Size
Not Attached	
ubuntu-22.04.1-desktop-amd64.iso	3,56 GB

Select the downloaded Ubuntu ISO

Start the VM



General

Name: SPCA Linux
Operating System: Ubuntu (64-bit)

System

Base Memory: 4096 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Primary Device 0: [Optical Drive] Empty
IDE Secondary Device 0: [Optical Drive] VBoxGuestAdditions.iso (58,24 MB)
Controller: SATA
SATA Port 0: SPCA Linux.vdi (Normal, 60,00 GB)

Audio

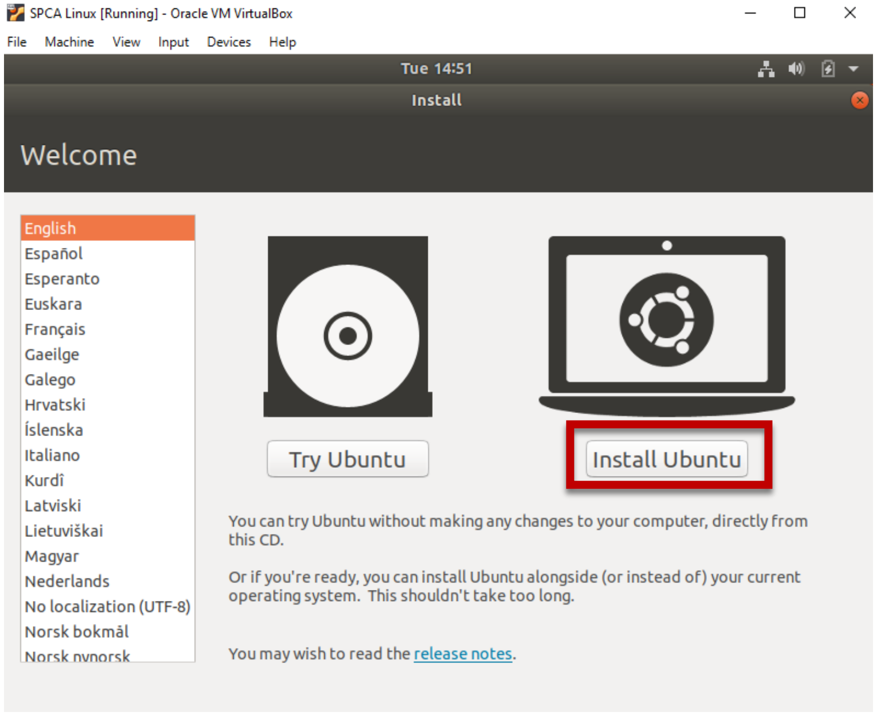
Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

Follow the Ubuntu installation wizard



Welcome

English
Español
Esperanto
Euskara
Français
Gaeilge
Galego
Hrvatski
Íslenska
Italiano
Kurdî
Latviski
Lietuviškai
Magyar
Nederlands
No localization (UTF-8)
Norsk bokmål
Norsk nynorsk

You can try Ubuntu without making any changes to your computer, directly from this CD.

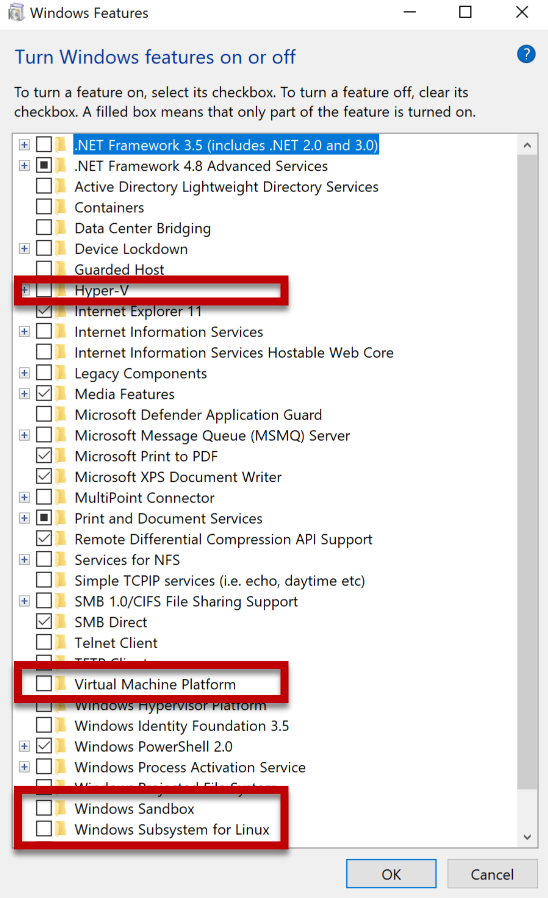
Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.

Troubleshooting: VM Setup

- If you have trouble installing Ubuntu 22.04 in Virtual Box, turn off the Windows Features:
  - Hyper-V
  - Virtual Machine Platform
  - Windows Sandbox
  - Windows Subsystem for Linux

https://stackoverflow.com/a/63229718



Windows Features

Turn Windows features on or off

To turn a feature on, select its checkbox. To turn a feature off, clear its checkbox. A filled box means that only part of the feature is turned on.

- ☐ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- ■ .NET Framework 4.8 Advanced Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Containers
- ☐ Data Center Bridging
- ☐ Device Lockdown
- ☐ Guarded Host
- ☐ Hyper-V
- ☑ Internet Explorer 11
- ☐ Internet Information Services
- ☐ Internet Information Services Hostable Web Core
- ☐ Legacy Components
- ☑ Media Features
- ☐ Microsoft Defender Application Guard
- ☐ Microsoft Message Queue (MSMQ) Server
- ☑ Microsoft Print to PDF
- ☑ Microsoft XPS Document Writer
- ☐ MultiPoint Connector
- ■ Print and Document Services
- ☑ Remote Differential Compression API Support
- ☐ Services for NFS
- ☐ Simple TCPIP services (i.e. echo, daytime etc)
- ☐ SMB 1.0/CIFS File Sharing Support
- ☑ SMB Direct
- ☐ Telnet Client
- ☐ TFTP Client
- ☐ Virtual Machine Platform
- ☐ Windows Hypervisor Platform
- ☐ Windows Identity Foundation 3.5
- ☑ Windows PowerShell 2.0
- ☐ Windows Process Activation Service
- ☐ Windows Projected File System
- ☐ Windows Sandbox
- ☐ Windows Subsystem for Linux

Optimal VirtualBox Settings

1. Set your graphics controller to VBoxSVGA and 3D acceleration off for automatic resolution scaling with decent performance

2. Give your at least VM 4GB of RAM and 32MB of graphics memory, if possible

3. If your computer allows for it, give the VM two CPU cores.
4. For people using laptops: VMs use a lot of performance. Try to either be plugged in or, on Windows, set your energy options to the performance setting for a better experience.

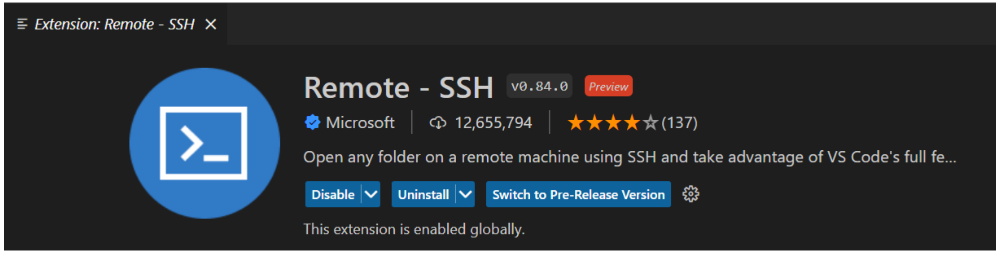
Alternative Solution: maximus.inf.ethz.ch

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

VS Code Remote Setup Part 1 [Optional]

- Install VSCode
- Install Remote SSH Extension (in VS Code)



Extension: Remote - SSH

Remote - SSH v0.84.0 Preview

Microsoft | 12,655,794 | ★★★★☆ (137)

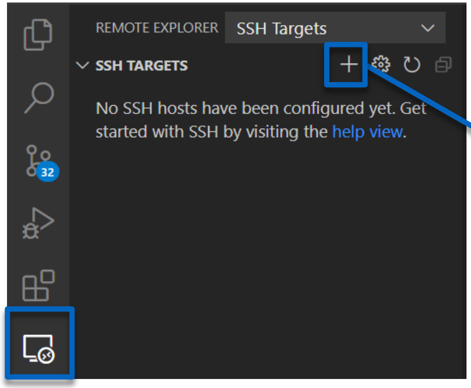
Open any folder on a remote machine using SSH and take advantage of VS Code's full fe...

This extension is enabled globally.

Alternative Solution: maximus.inf.ethz.ch

VS Code Remote Setup Part 2 [Optional]

- Add a new SSH Target

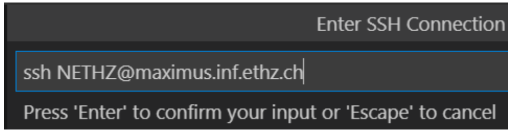


REMOTE EXPLORER SSH Targets

SSH TARGETS

No SSH hosts have been configured yet. Get started with SSH by visiting the help view.

Replace NETHZ

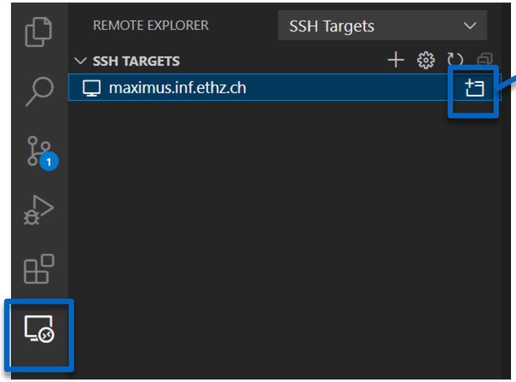


Enter SSH Connection

ssh NETHZ@maximus.inf.ethz.ch

Press 'Enter' to confirm your input or 'Escape' to cancel

- Connect to SSH Target

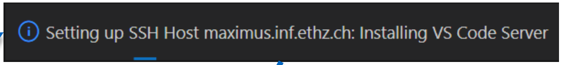


REMOTE EXPLORER SSH Targets

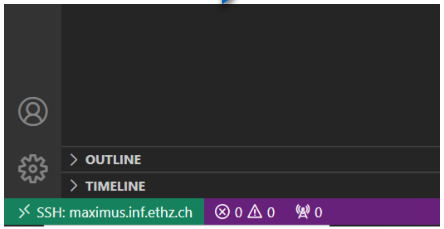
SSH TARGETS

maximus.inf.ethz.ch

Login and Wait



Setting up SSH Host maximus.inf.ethz.ch: Installing VS Code Server



OUTLINE

TIMELINE

SSH: maximus.inf.ethz.ch ⊗ 0 ⚠ 0 0

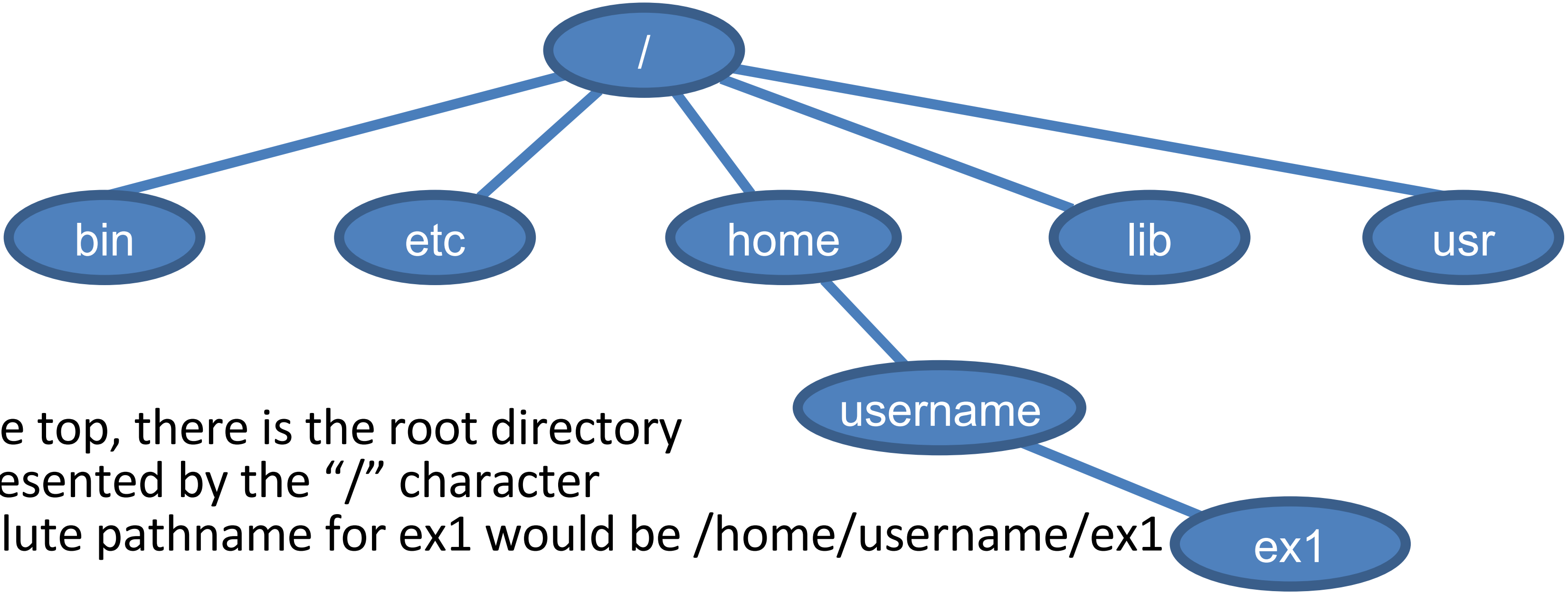
Systems@ETH Zürich

Introduction to Linux

Ubuntu 22.04 LTS

File System

- 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 Files

- **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
- cd /home/username/ex1

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

Browsing the Files

- **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

- **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

Misc

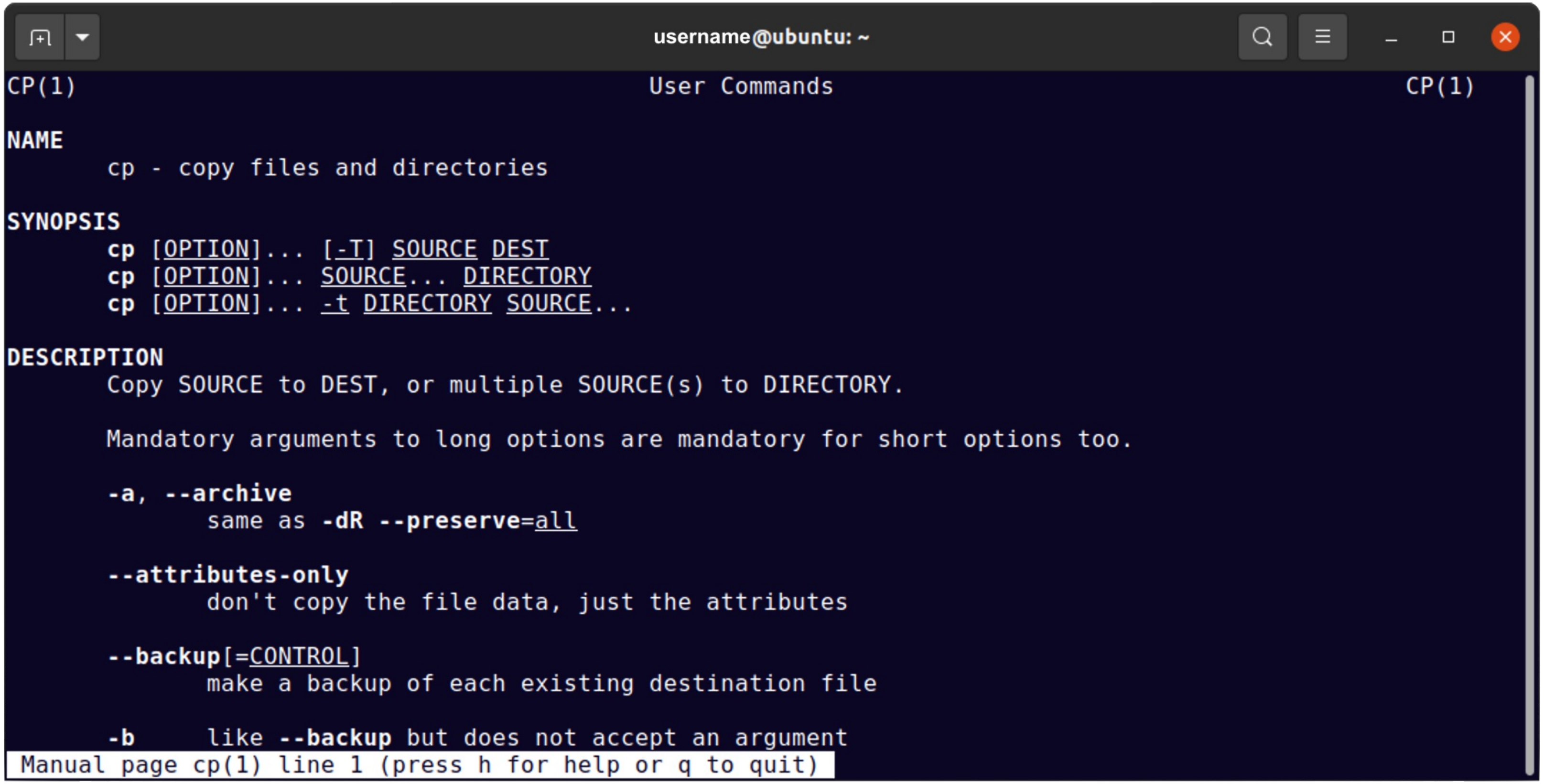
- **gedit, emacs, vi/vim**: useful text editors for writing your programs and editing files.

- **cat, more, less**: useful to view files
- **grep**: useful for searching text files

- **gcc/gdb:** compilers and debuggers

Lost? Try "man".

man cp

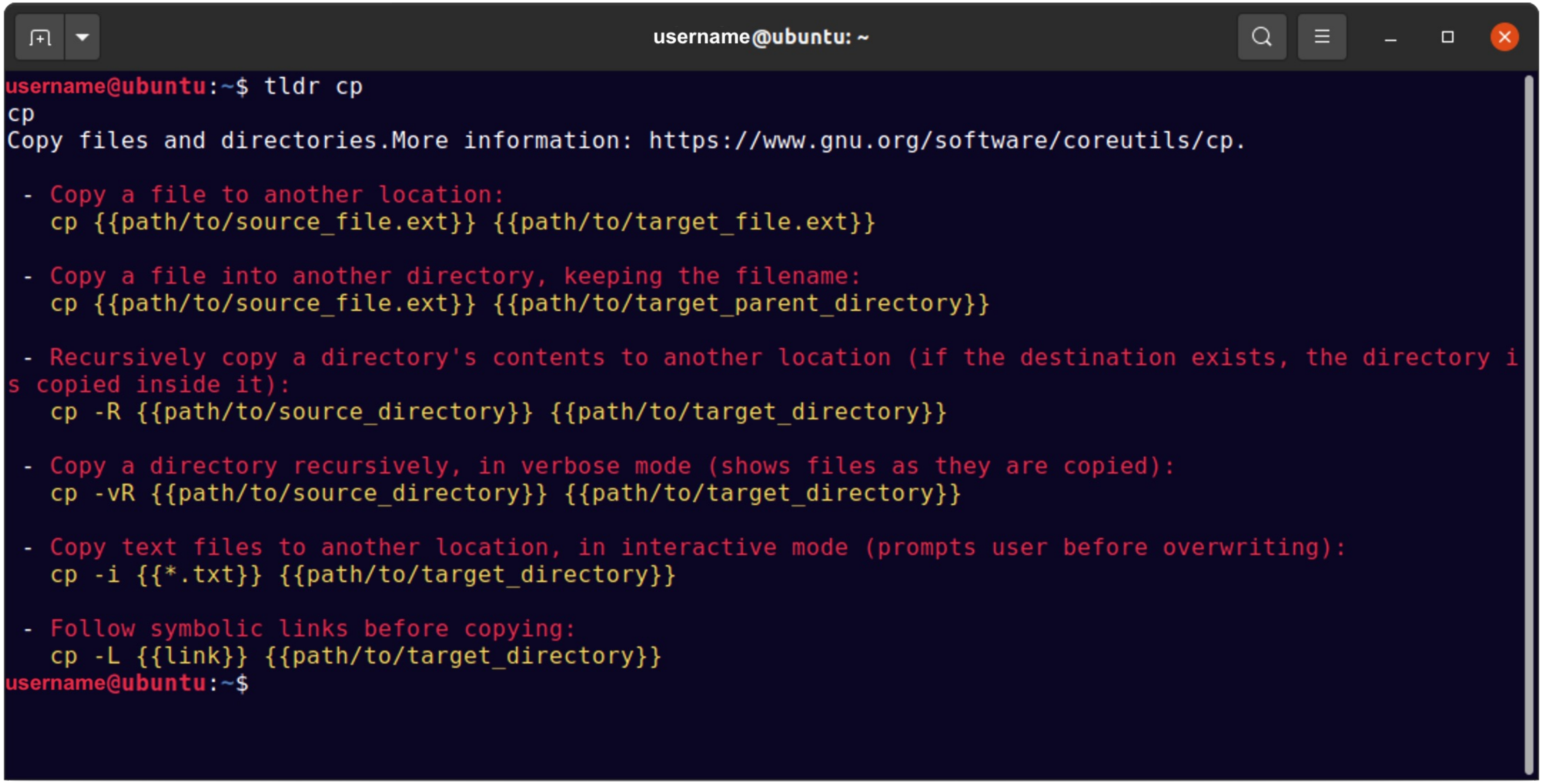


```
CP(1)                            User Commands                           CP(1)

NAME
       cp - copy files and directories

SYNOPSIS
       cp [OPTION]... [-T] SOURCE DEST
       cp [OPTION]... SOURCE... DIRECTORY
       cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
       Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --archive
              same as -dR --preserve=all

       --attributes-only
              don't copy the file data, just the attributes

       --backup[=CONTROL]
              make a backup of each existing destination file

       -b     like --backup but does not accept an argument
 Manual page cp(1) line 1 (press h for help or q to quit)
```

Still lost? Try “tldr”.

Can be installed with `sudo apt install tldr`

tldr cp



```
username@ubuntu:~$ tldr cp
cp
Copy files and directories.More information: https://www.gnu.org/software/coreutils/cp.

 - Copy a file to another location:
   cp {{path/to/source_file.ext}} {{path/to/target_file.ext}}

 - Copy a file into another directory, keeping the filename:
   cp {{path/to/source_file.ext}} {{path/to/target_parent_directory}}

 - Recursively copy a directory's contents to another location (if the destination exists, the directory i
s copied inside it):
   cp -R {{path/to/source_directory}} {{path/to/target_directory}}

 - Copy a directory recursively, in verbose mode (shows files as they are copied):
   cp -vR {{path/to/source_directory}} {{path/to/target_directory}}

 - Copy text files to another location, in interactive mode (prompts user before overwriting):
   cp -i {{*.txt}} {{path/to/target_directory}}

 - Follow symbolic links before copying:
   cp -L {{link}} {{path/to/target_directory}}
username@ubuntu:~$
```

More tutorials online

- 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 for more!

Preview of Assignment 1

The Data Lab

Pre-requisites

- 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 it's value
- e.g.: integers are represented with two's complement

x

1	0	1	1	1	0	0	1

signed char x = -71

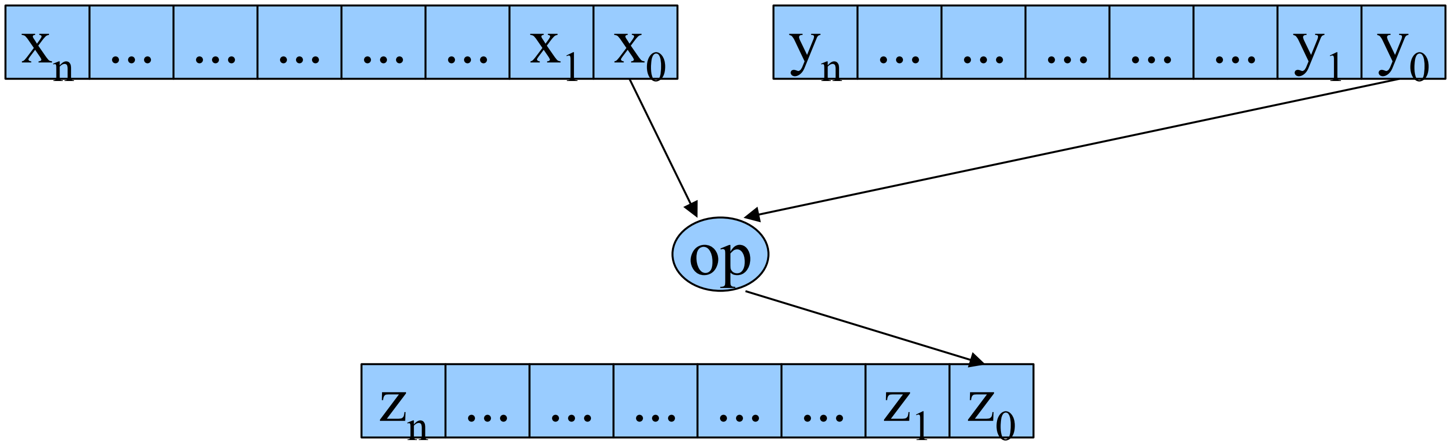
x

1	0	1	1	1	0	0	1

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 op Y → $Z_i = X_i \text{ op } Y_i$



Logical vs Bitwise Operators

- **Logical operators** evaluate the truth or falsity of **an expression**
  - The result is either **true or false**
  - 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: fill left-end with 0’s, used with unsigned types
  - Arithmetic: 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=0011;          //x=3
z=x << 2;        //z = 001100 = 12 = 3*2^2
```

Your Turn! Do the homework

- 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!

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

- 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

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

Checkout your repository

(replace the placeholder NETHZ below with your NETHZ)

- Clone your repository

```
$ git clone git@gitlab.inf.ethz.ch:course-spca2022/spca22-
NETHZ.git
```

- This will create a folder "`spca22-NETHZ`"

```
$ cd spca22-NETHZ
```

Submitting your solution

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

- Make a new directory and copy your solution into it

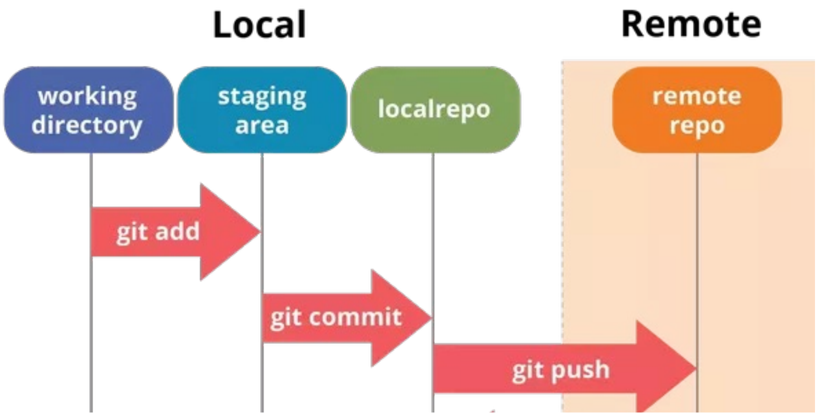
```
$ mkdir assignment1
$ cp bits.c assignment1
```

- 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



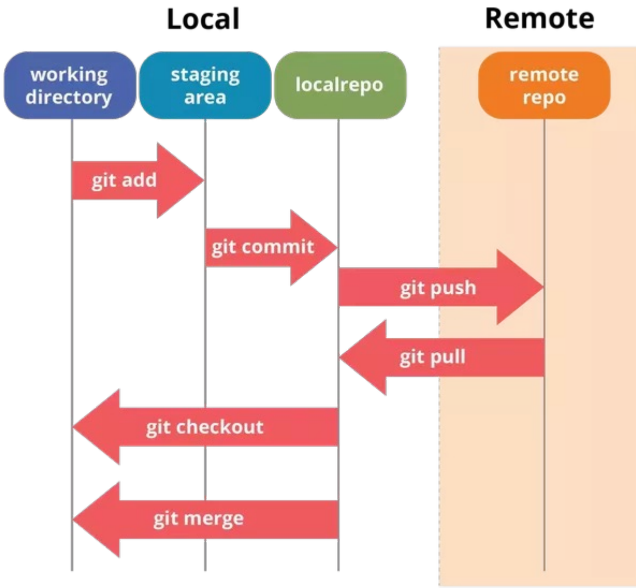
- Commits = savegame
- add and commit do not do any network access

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`