



Computing Package - Day 1 of 4

Ali Skaf, Sebastian Wozniewski, Chritian Scheulen, Ishan Pokharel, Marcel Niemeyer, Sreelakshmi Sindhu, Steffen Korn, Wael Alkakh

II. Physikalisches Institut, Universität Göttingen

March 8, 2022

1. CERN Resources
2. Introduction to Linux
3. Introduction to Emacs
4. C++ tutorial, OOP, Debugging
5. Hands-on Exercises

Day 1

6. Python tutorial
7. Git tutorial

Day 2

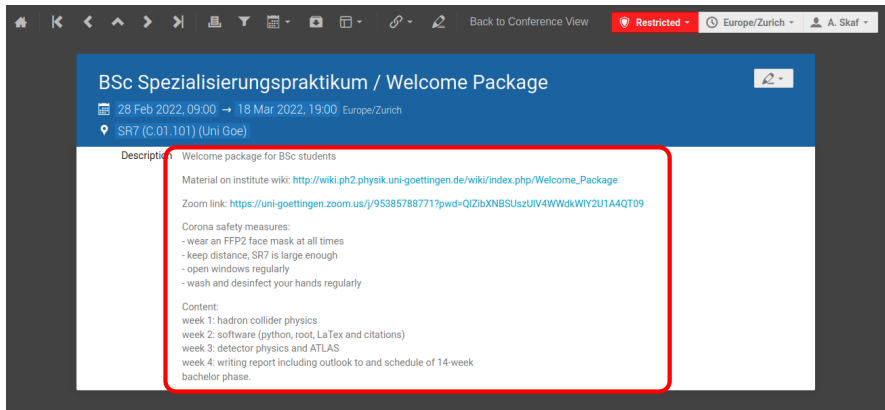
8. Root tutorial

Day 3

9. ML tutorial

Day 4

- Will be used for anything organisational
- The page of this welcome package: Indico



The screenshot shows an Indico event page. At the top, there is a navigation bar with icons for home, back, forward, search, and other functions. The event title is "BSc Spezialisierungspraktikum / Welcome Package". Below the title, the dates are "28 Feb 2022, 09:00 → 18 Mar 2022, 19:00" and the location is "Europe/Zurich". The event is marked as "Restricted" and the user "A. Skaf" is logged in. The description of the event is as follows:

Description: Welcome package for BSc students

Material on institute wiki: http://wiki.ph2.physik.uni-goettingen.de/wiki/index.php/Welcome_Package

Zoom link: <https://uni-goettingen.zoom.us/j/95385788771?pwd=QIZibXNBSUzUjV4WWdkWlY2U1A4QT09>

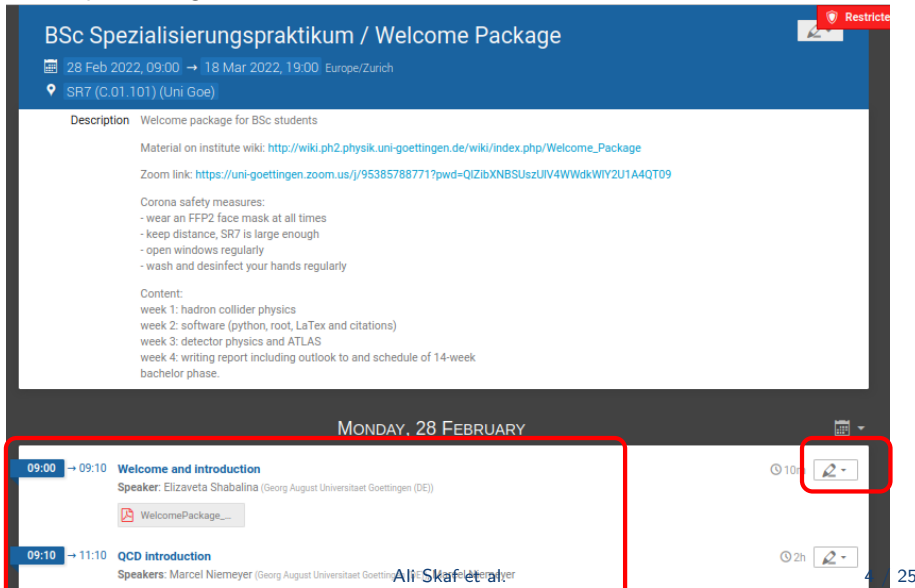
Corona safety measures:

- wear an FFP2 face mask at all times
- keep distance, SR7 is large enough
- open windows regularly
- wash and disinfect your hands regularly

Content:

- week 1: hadron collider physics
- week 2: software (python, root, LaTeX and citations)
- week 3: detector physics and ATLAS
- week 4: writing report including outlook to and schedule of 14-week bachelor phase.

- Important agenda view



BSc Spezialisierungspraktikum / Welcome Package

28 Feb 2022, 09:00 → 18 Mar 2022, 19:00 Europe/Zurich

SR7 (C.01.101) (Uni Goe)

Description Welcome package for BSc students

Material on institute wiki: http://wiki.ph2.physik.uni-goettingen.de/wiki/index.php/Welcome_Package

Zoom link: <https://uni-goettingen.zoom.us/j/95385788771?pwd=QlZibXNBSUszUUV4WWdkWlY2U1A4QT09>

Corona safety measures:

- wear an FFP2 face mask at all times
- keep distance, SR7 is large enough
- open windows regularly
- wash and disinfect your hands regularly

Content:

- week 1: hadron collider physics
- week 2: software (python, root, LaTeX and citations)
- week 3: detector physics and ATLAS
- week 4: writing report including outlook to and schedule of 14-week bachelor phase.

MONDAY, 28 FEBRUARY

09:00 → 09:10 Welcome and introduction

Speaker: Elizaveta Shabalina (Georg August Universität Göttingen (DE))

WelcomePackage_...

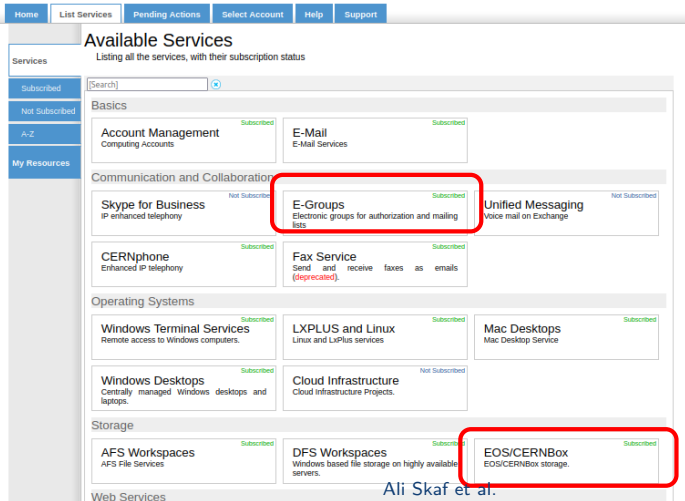
09:10 → 11:10 QCD introduction

Speakers: Marcel Niemeyer (Georg August Universität Göttingen (DE)), Ali Skaf et al.

- Click on „List Services“ to get an overview of your subscriptions
- The page of this welcome package: Available here

CERN Resources Portal

Manage your CERN Resources, lifecycle, settings, etc.



The screenshot displays the CERN Resources Portal interface. At the top, a navigation bar includes links for Home, List Services, Pending Actions, Select Account, Help, and Support. Below this, the 'Available Services' section is titled 'Listing all the services, with their subscription status'. A search bar is present. The services are categorized into several groups: Basics, Communication and Collaboration, Operating Systems, and Storage. Each service card indicates its subscription status with a green 'Subscribed' label or a blue 'Not Subscribed' label. Two services are highlighted with red rectangles: 'E-Groups' under Communication and Collaboration, and 'EOS/CERNBox' under Storage. Both are marked as 'Subscribed'.

Category	Service	Subscription Status
Basics	Account Management Computing Accounts	Subscribed
	E-Mail E-Mail Services	Subscribed
Communication and Collaboration	Skype for Business IP enhanced telephony	Not Subscribed
	E-Groups Electronic groups for authorization and mailing lists	Subscribed
	Unified Messaging Voice mail on Exchange	Not Subscribed
	CERNphone Enhanced IP telephony	Subscribed
Operating Systems	Fax Service Send and receive faxes as emails (deprecated)	Subscribed
	Windows Terminal Services Remote access to Windows computers.	Subscribed
	LXPLUS and Linux Linux and LxPlus services	Subscribed
	Mac Desktops Mac Desktop Service	Subscribed
Storage	Windows Desktops Centrally managed Windows desktops and laptops.	Subscribed
	Cloud Infrastructure Cloud Infrastructure Projects.	Not Subscribed
	AFS Workspaces AFS File Services	Subscribed
	DFS Workspaces Windows based file storage on highly available servers.	Subscribed
Storage	EOS/CERNBox EOS/CERNBox storage.	Subscribed

- You can use CERNBox to store/share files.

CERN Resources Portal

Manage your CERN Resources, lifecycle, settings, etc.

[Home](#) [List Services](#) [Pending Actions](#) [Select Account](#) [Help](#) [Support](#)

Service Information

EOS/CERNBox

EOS/CERNBox storage.

Storage and web space for personal use, projects and experiments.

Personal space

CERNBox provides personal data storage space to anyone having a CERN account. Data can be:

- Accessed from any web browser or file explorer.
- Shared between individuals or groups of collaborators.
- Synchronised across a series of devices (smartphone, tablet, laptop, desktop).
- Accessed via the SHARED Filesystem on lxplus or via EOS.

CERNBox uses EOSUSER (an instance of EOS, the CERN disk storage system) as the storage backend.

To activate your CERNBox personal storage space (1 TB, up to 1 million files, hosted in the CERN Computer Center): login to <https://cernbox.cern.ch> using your CERN account and password).

Projects and experiments

EOS spaces for physics data are managed by the experiments themselves. All other EOS space is managed by CERN-IT. Please consult [KB0003151](#) to request for space.

Help and support

To report a problem, contact [CERNBox Service](#).

Related sites

- [Account Management](#)
- [Service Portal \(Get Help\)](#)

- Choose your e-groups of interest.
- Some e-group subscriptions require approval.

European Laboratory for Particle Physics

Report on [e-groups](#) | Suggest new [e-groups](#)

Quick Search
 e-group name | begins with | Search
☐ All e-groups ☒ ALICE ☐ ATLAS ☐ CMS ☐ FASER ☐ LCG ☐ LHCB ☐ LHCb ☐ MEDAL ☐ SND@LHC ☐ TOTEM
☐ Only groups I own or manage | ☐ Only groups I am on | Page Size:

AB SKAF | Group Memberships: 148 | [Logout](#)

[Create new static group](#) [Create new dynamic group](#) [Show groups for one member](#) [Manage groups for one member](#) [Manage user roles](#)

E-groups

Go to: 1-30

Name	Type	Topic	Description	Status	Owner	Actions	Archive
200-experiments-default-access	Static		Default read access group for machine 360 personas	Active	thomas.william.battle@cern.ch		
ab-skaf	Dynamic		People with active Ab-skaf account	Active	win.van'tremm@cern.ch		
ab-skaf-external-access	Static	AIS-ODI	Users access to E-Groups Web Services	Active	Rafaela.Tibb@cern.ch		
ab-skaf-users	Dynamic		Persons status is USER UPAS or USAS	Active	Azma.Cook@cern.ch		
ab-skaf	Dynamic		contains all MPAs	Active	Gabriele.Tiande@cern.ch		
ab-skaf-at-cern	Static		All people with status USER UPAS or UPAS	Active	Azma.Cook@cern.ch		
ab-skaf	Dynamic		ap-user	Active	Catherine.Hegobu@cern.ch		
ab-skaf-act	Static	ATLAS Information Protection	IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-1	Static		IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act	Static		IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act-1	Static		IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act-2	Static		IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act-3	Static		IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act-4	Static	ATLAS Management	ATLAS Active Members Protection Group	Active	Daria.Barberis@cern.ch		Go to archive
ab-skaf-act-act-5	Static		ATLAS active members - including additional	Active	Daria.Barberis@cern.ch		
ab-skaf-act-act-6	Static	ATLAS Information Protection	IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act-7	Static		IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act-8	Static		IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act-9	Static		IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act-10	Static		IPD testing purposes	Active	Oleg.Zernin@cern.ch		
ab-skaf-act-act-11	Static	ATLAS Information Protection	For IPD clearing of computing groups	Active	Daria.Barberis@cern.ch		
ab-skaf-act-act-12	Static		Active ATLAS physicists	Active	elzabeth.gawed@physics.ac.uk		Go to archive
ab-skaf-act-act-13	Static		General	Active	Kate.Richardson@cern.ch		Go to archive
ab-skaf-act-act-14	Dynamic	ATLAS Institutes	ATLAS participants from German institutes	Active	Annul.Quadt@cern.ch		
ab-skaf-act-act-15	Static	ATLAS TriggerDAQ and Online	ATLAS Inner Trigger Upgrade DAQ	Active	Paolo.Morobbi@cern.ch	Unsubscribe	

- Will be using the terminal for this introduction and other tutorials today
- Open a terminal on your computer and connect to lxplus:

```
> ssh -Y username@lxplus.cern.ch
```

- You are now connected to a computer at Cern running CentOS7
- It's also possible to connect to the local computers:

```
> ssh -Y -p 24 username@login.ph2.physik.uni-goettingen.de  
> ssh -Y pcatlasXX
```

- Will be using lxplus here and the git tutorial

- 1969: UNIX operating system developed at AT&T Bell Labs.
- 1975: two versions : the commercial Unix, the official AT&T Unix, and the free BSD Unix (Development of BSD descendants like FreeBSD, OpenBSD, NetBSD, DragonFly BSD and PC-BSD is still active today).
- 1980s:
 - IBM created AIX, Sun SunOS (later Solaris), HP HP-UX and about a dozen other companies did the same.
 - root of Linux with Richard Stallman and the GNU project. His goal was to make an operating system that was freely available to everyone
- 1992: Linus Torvalds, a Finnish student, built first Linux version
- 2015: Linux kernel version 4.0 was released; with source code of several 100k lines thanks to contributions of thousands of developers paid by hundreds of commercial companies.

- **Red Hat:** Red Hat is a billion dollar commercial Linux company (Red Hat Enterprise Linux (RHEL) and Fedora. Free source, but paid support!
- **Ubuntu:** released in 2004 by Canonical, with nice graphical desktop, can be used as Windows.
- **Debian:** no company, but a well organized developer community
- **SuSE and Mandriva:** widely used
- **CentOS, Oracle Enterprise Linux and Scientific Linux:** (based on Red Hat)
- **Linux Mint, Edubuntu:** and many other *buntu named distributions are based on Ubuntu

A set of standard ways to handle and use the system, found in most linux distributions

- Bash: The GNU shell
- GCC: The GNU C Compiler
- GDB: The GNU Debugger
- Coreutils: a set of basic UNIX-style utilities, such as ls, cat and chmod
- Findutils: to search and find files
- Fontutils: to convert fonts or make new fonts
- Gimp: GNU Image Manipulation Program
- Gnome: the GNU desktop environment
- Emacs, Vim: very powerful editors
- Ghostscript - Ghostview: interpreter & graphical frontend for PostScript files.
- GNU SQL: relational database system

No dedicated tutorial, please try some of these commands as we go:

<code>uname -a</code>	# Display Linux system information
<code>uname -r</code>	# Display kernel release information
<code>uptime</code>	# Show how long the system has been running + load
<code>hostname</code>	# Show system host name
<code>hostname -I</code>	# Display the IP addresses of the host
<code>last reboot</code>	# Show system reboot history
<code>date</code>	# Show the current date and time
<code>cal</code>	# Show this month's calendar
<code>w</code>	# Display who is online
<code>whoami</code>	# Who you are logged in as

No dedicated tutorial, please try some of these commands as we go:

ls -al	# List a ll files in a long l isting (detailed) format
pwd	# Display the p resent w orking d irectory
mkdir	# m ake a directory
cd	# c hange d irectory
cd ~	# Move to your home directory
cd ..	# Move one directory up
rm	# R emove file permanently!
rm -r	# Remove a directory and its content r ecursively
rm -f	# F orce remove a file
cp -r file dest	# C opy file (or directory) to its destination file or directory

No dedicated tutorial, please try some of these commands as we go:

<code>mv file1 file2</code>	<code>#</code> M ove or rename file1 to file2 (also directories)
<code>ln -s file linkname</code>	<code>#</code> Create a s ymbolic l ink to linkname
<code>touch</code>	<code>#</code> Create an empty file or update modification times
<code>cat</code>	<code>#</code> Display context of file
<code>less</code>	<code>#</code> Browse a file in the terminal (q to quit, / to search)
<code>head</code>	<code>#</code> Display first 10 lines of a file
<code>tail</code>	<code>#</code> Display last 10 lines of a file
<code>grep "string" file</code>	<code>#</code> Look through a file for "string"
<code>sed</code>	<code>#</code> S teamline E ditor to modify files w/o opening them
<code>find</code>	<code>#</code> find files in the system

- In general: use the "--help" flag to get usage information
- You can use "man commandName" to get detailed info on that command

- Permissions of a file can be seen with "ls -l file"
- Generally, three different permissions:
 - r: reading the file
 - w: write/modify the file
 - x: execute the file (e.g. shell-scripts, see later)
- Can be changed using "chmod"

User	Group	Other	
rwX	rwX	rwX	chmod 777 filename
rwX	rwX	r-X	chmod 775 filename
rwX	r-X	r-X	chmod 755 filename
rw-	rw-	r--	chmod 664 filename
rw-	r--	r--	chmod 644 filename
421	421	421	binary encoding for permissions

No dedicated tutorial, please try some of these commands as we go:

ps	# Display your currently running processes
ps -ef	# Display all processes on the system
top	# Display and manage the top processes
htop	# Interactive top
kill pid	# Kill a process with specific Process ID
killall processname	# Kill all processes with specific name
command &	# Start command in background
jobs	# Display your jobs
Ctrl + C	# Kill the job currently running in the foreground
Ctrl + Z	# Pause the job currently running in the foreground
fg %n	# Run job n in the foreground
bg %n	# Run job n in the background

No dedicated tutorial, please try some of these commands as we go:

<code>ssh host</code>	<code># Open a Secure Shell to work on host</code>
<code>ssh user@host</code>	<code># Connect as user</code>
<code>ssh -p port</code>	<code># Connect via a certain port</code>
<code>ssh -Y</code>	<code># Allow the use of graphical interfaces</code>
<code>scp file1 server:file2</code>	<code># Secure cp file1 to the server as file2</code>
<code>scp server:file1 file2</code>	<code># Copy file from server to local computer</code>
<code>logout/exit/Ctrl+D</code>	<code># End connection</code>

- Shell scripts are text files consisting of many shell commands
- Useful for longer operations and sequences of commands that you need to do very often
- Typically end on ".sh"
- Two ways of running them:
 - "source script.sh": Goes line by line
 - "./script.sh": if it includes bash coding
- Example: "~/.bashrc" or "~/.bash_profile"
 - Are run every time that you open the shell
- Used for shell configurations, aliases
- Aliases define short commands for long commands, e.g. alias short="very long command or path to shell script"

Pipelines:

- Pass the std output of one command to another
- example: "ps -ef | grep processname" to search for a specific process
- "|&" includes also the std error
- Similar: Redirection
- "command > file" redirects the stdout of command to file
- "command >> file" appends the stdout to the file

Regular Expressions:

- doesn't look for strings but for patterns
- look a bit cryptic
- No details in this tutorial, too extensive
- Used for grep and sed

Example from my .bashrc:

```
alias createLocalList='ls -l | sed -E "s/( ){2,}/ /g" | cut -d" " -f9 | sed -E  
"s@(^.)@$PWD/\1@" | sed -E "/(^$|local_list.txt)/d" | cat > local_list.txt'
```

- You might need to work remotely a lot
- Using graphical interfaces is not the best idea
- Vim, Emacs, and Nano can all be run inside the terminal
- Nano is a very basic editor, fine for small edits
- For development work, you might want to have a look at Vim or Emacs
- For now, it's not too important which one you chose, as long as it works for you

Classical learning
curves for some
common editors

Notepad

Brackets



Vim

Emacs

VIM
visible in just about
any environment.
does one thing, well.



EMACS
flexible, customizable, and
packed with useful features
known to man.



NANO
readily used by people
who do not know
what they are doing,
or psychopaths.



© 2010 CLIFF CARRAN - CARRAN-PHOTO.COM

- Emacs is a very feature-rich text editor that can be used in the terminal
- Start it by doing "emacs -nw [file]" (file is optional, no need to use one here)
- Inside of emacs, data is organised as buffers
- If you don't specify a file, you'll be opening the **scratch** buffer
- Buffers named like: **name** are virtual and do not correspond to any files
- Scratch is for taking notes and simple ELisp evaluation
- Navigate inside of Emacs by using modifier keys "Ctrl" (C) and "Alt/Meta" (M)
- Basic configuration is set in "~/.emacs" (or similar)
- Includes own package manager
- Commands are usually written like this: C-x or M-d

- Managing buffers includes file operations, starting other processes in emacs, etc.
- Commonly bound to "C-x <something>"

C-x C-f	# Open a file
C-x C-s	# Save file
C-x C-w	# Write file to [prompt]
C-x C-b	# Show a list of buffers
C-x b	# Switch between buffers [prompt]
C-x 2/3	# Split window horizontally/vertically
C-x o	# Switch between windows
C-x 0/1	# Close current window/all windows except current one
C-x 4 C-f/b	# Open file/buffer in new window
C-x k	# Kill buffer [prompt]
C-x C-c	# Close emacs

- Moving is possible with arrows and page up/down
- Also own keybindings, if needed

C-p/n	# Moving to previous/next line
M-{/}	# Moving to previous/next paragraph
C-f/b	# Moving forward/back in line (1 character)
M-f/b	# Moving forward/back in line (1 word)
C-a/e	# Move to start(Anfang?)/end of line
C-v	# Page down
M-v	# Page up
M->/<	# Move to end/beginning of file
M-g M-g	# Go to line number [prompt]
C-l	# Set current line to center/top/bottom of window (repeated use)

- By default, emacs adapts somewhat to the language used for the file (e.g. cpp or python), called major-mode
- Copying etc. works between buffers

C-s/r	# search forward/reverse [prompt]
M-%	# Find and replace [prompt]
C-M-s	# search forward (regex enabled)
C-M-%	# find and replace (regex enabled)
C-Space	# Mark text [move cursor afterwards]
C/M-w	# Cut/Copy marked text
C-y	# "yank" (paste) text
M-y	# Move backwards through copied texts
C/M-k	# Kill line/next word
C-/	# Undo
C-g C-/	# Redo
C-<number> <command>	# Do <command> <number> times

- Much more that wasn't shown here
- All commands can be called by "M-x <command name>"
- "M-x package list packages" leads you to the package manager
- Can also be used to start processes, e.g. "M-x term" or "M-x grep"
- If you want to do something with emacs, there's probably an extension for it, be it auto-complete, remote editing, a pdf viewer or a browser

