Scientific computing in high-energy physics

Lecture 1, 23.04.2020

Ante Bilandzic
(E62, Dense and Strange Hadronic Matter)

European Research Council
Established by the European Commission

Outline of today's lecture

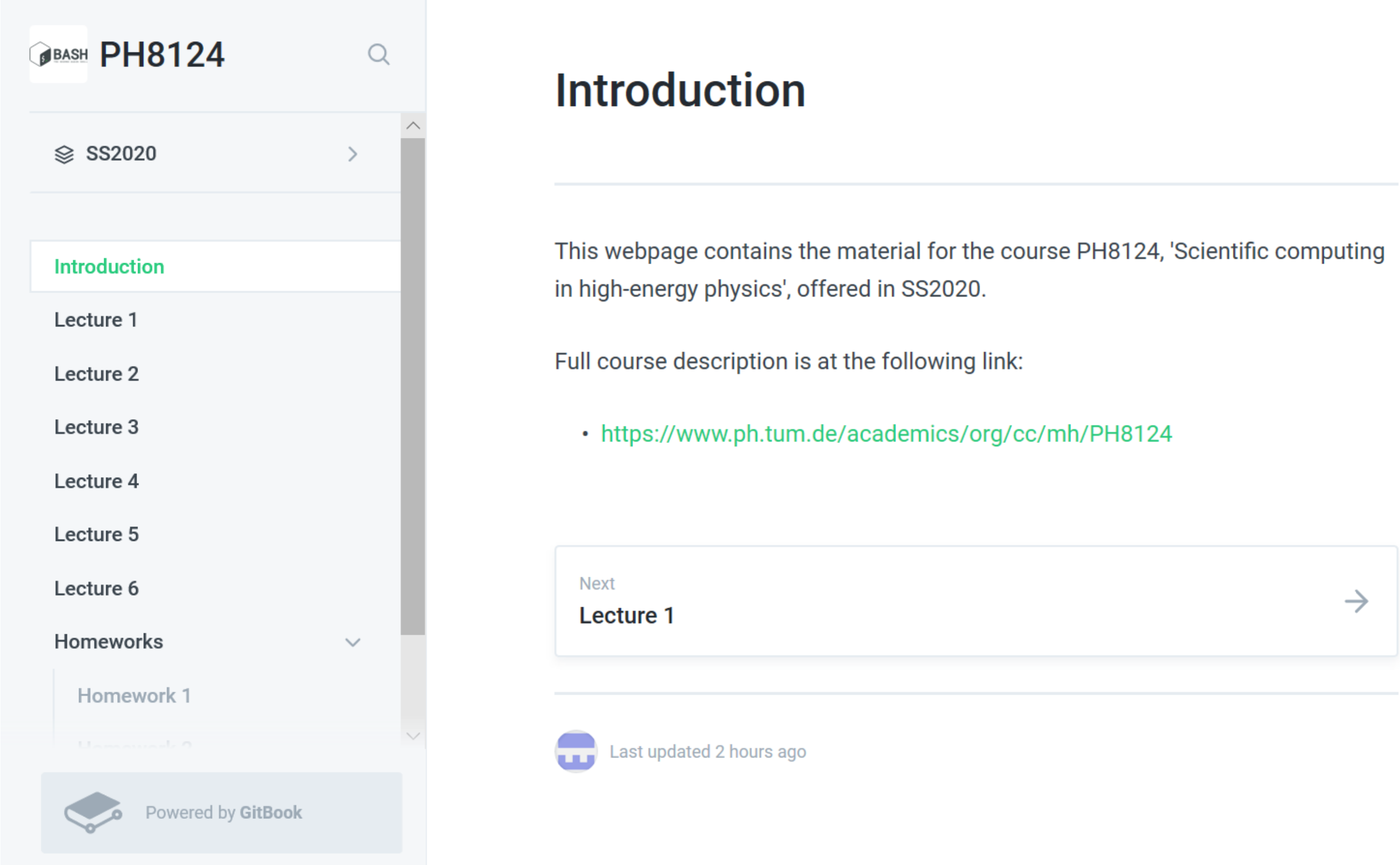
- Trivia
- Introduction
- Hands-on: Taking off

Trivia

- **PH8124:** 'Scientific computing in high-energy physics'
  - https://campus.tum.de/tumonline/wbLv.wbShowLVDetail?pStpSpNr=950465541
  - https://www.ph.tum.de/academics/org/cc/mh/PH8124/
- When & where:
  - SS 2020
  - Thursday: 14:00-16:00
  - Physics Department, E12 seminar room 2024
  - 12 contact days (last lecture is on July 23rd)
- Examination:
  - Homeworks (10 in total)
  - Final oral examination over the final project presentation
- Contact:
  - Ante Bilandzic, ante.bilandzic@tum.de, Office PH 2162

Webpage

- The lecture has a dedicated webpage:
  - Link: https://abilandz.gitbook.io/ph8124/v/SS2020/



Introduction

This webpage contains the material for the course PH8124, 'Scientific computing in high-energy physics', offered in SS2020.

Full course description is at the following link:

- https://www.ph.tum.de/academics/org/cc/mh/PH8124

Last updated 2 hours ago

Trivia

- After each lecture, executive summary of the covered material will be shared via email via official TUM online interface for this course
- The course webpage will be updated regularly
- In the same way, I will also share the homework exercises
- Recommended literature:
  - Mendel Cooper: ‘Advanced Bash-Scripting Guide’ (http://tldp.org/LDP/abs/abs-guide.pdf)
  - Cameron Newham and Bill Rosenblatt, ‘Learning the bash Shell: Unix Shell Programming (In a Nutshell (O'Reilly))’
  - ROOT User’s Guide (https://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuide.html)

Trivia

- Grading:
  - 3 ECTS points
  - Final grade = grade at final project examination - ‘1 unit’ if you have completed correctly 75% of all homeworks
    - There will be in total 10 homework exercises, one after each lecture
- Oral examination at the final project presentation:
  - List of topics for the final programming project will be offered at some point towards the end of the lecture
  - The oral exam of about 25 minutes consists of presenting:
    1. How your programme was designed/implemented?
    2. Testing the execution of your code (crash-free, bug-free, efficiency in terms of CPU usage and memory consumption)
    3. Testing the code flexibility (e.g. how you would add some new feature in the code?)

Trivia

- Preliminary list of topics to be covered:
  - **Linux:** filesystem hierarchy and file manipulation, handling processes and jobs, frequently used commands, etc.
  - **Bash:** shell environment, variables, string manipulation, built-in commands, aliases, functions, conditional statements, loops, command substitution, command chain, test constructs, piping, redirections, code blocks, subshells, process substitution, brace expansion, regular expressions, here-strings and here-documents, etc.
  - **ROOT:** using ROOT GUI, plotting, histogramming, functions, fitting, trees, file merging, etc.

Trivia

- Course classification
  - At the moment, classified as a ‘Non-physics elective course’, open both to Bachelor and Master students
    - https://www.ph.tum.de/academics/msc/physics/nonphys/
- Course evaluation
  - At some point during the lecture, you will be asked to evaluate the quality of this course: Please, do it! (reminder will be sent later)

Trivia

- Whenever there are exceptional circumstances which prevent us to meet in person, course will be offered via internet
- Software to be used for online lecturing:
  - Zoom: https://tum-conf.zoom.us/ (licensed version!)
  - Vidyo: https://www.vidyo.com/
- Please install and familiarize yourself both with Zoom and Vidyo
  - By default, we use Zoom

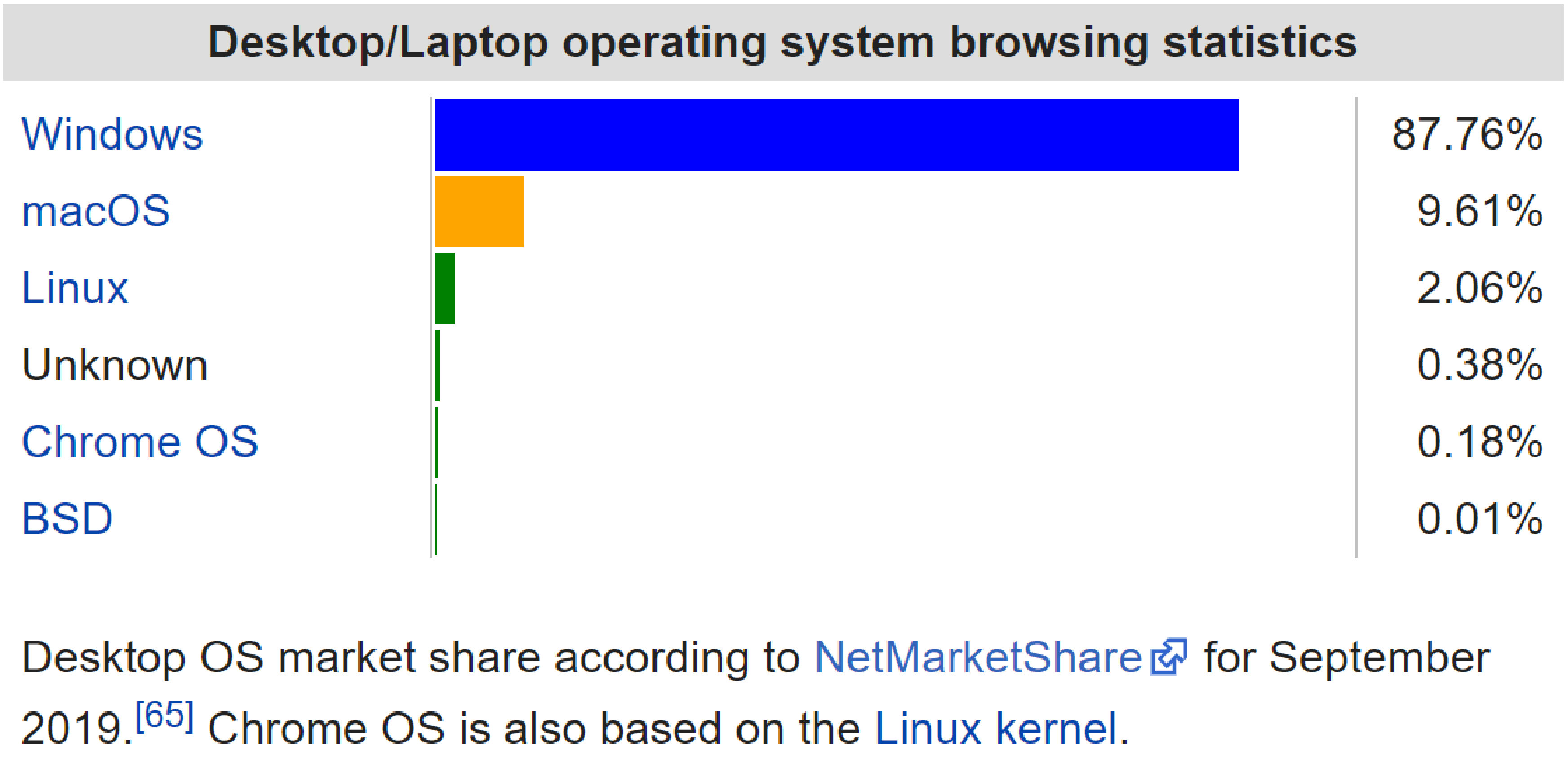
Trivia

- No lecture on:
  - May 21st – Ascension Day
  - June 11th – Corpus Christi

Introduction

Free advert (#1)

- Why Linux? Statistics for all desktop/laptop computers:

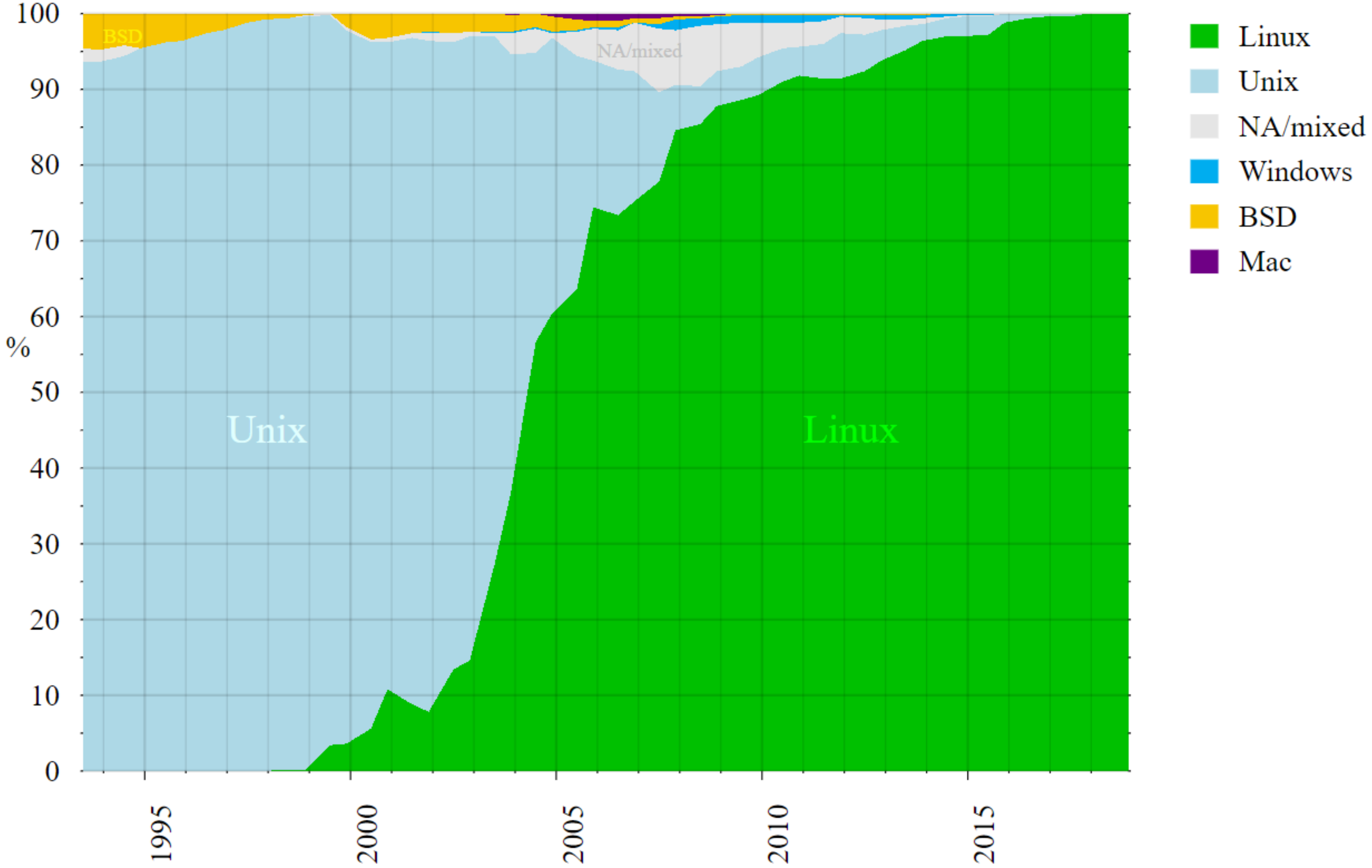


Desktop/Laptop operating system browsing statistics	
Windows	87.76%
macOS	9.61%
Linux	2.06%
Unknown	0.38%
Chrome OS	0.18%
BSD	0.01%

Desktop OS market share according to NetMarketShare for September 2019.[65] Chrome OS is also based on the Linux kernel.

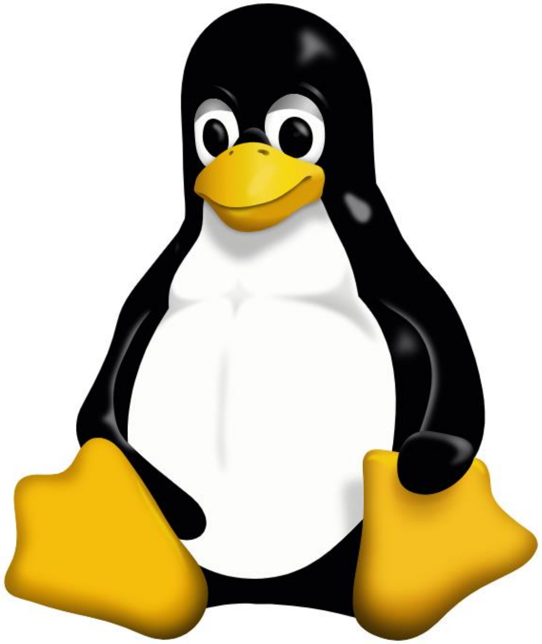
Free advert (#1)

- Why Linux? Statistics for supercomputers:



Free advert (#1)

- Why Linux?
- Linux is by far the leading operating system in computers used in scientific research (CERN, NASA, etc.)
- Developed initially by Linus Torvalds in the early 90s, and then by thousands of collaborators afterwards



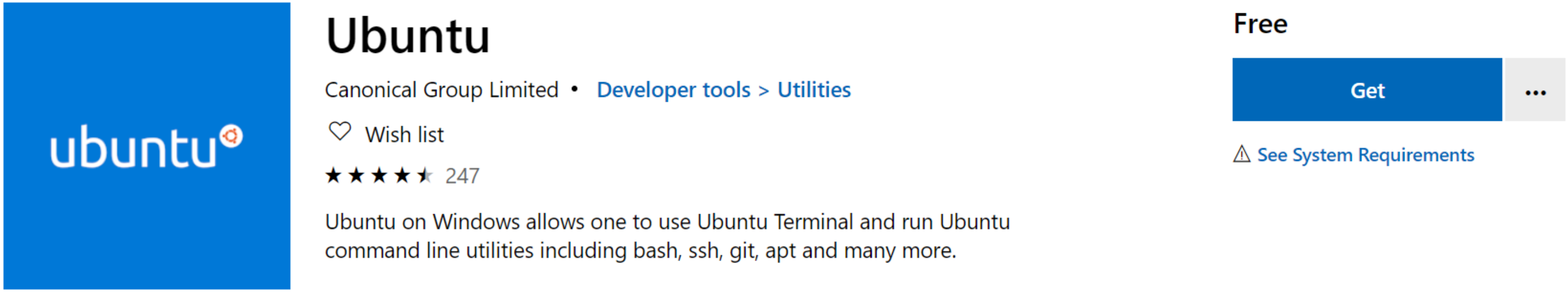
Penguin named Tux is the most commonly used logo for Linux

Free advert (#1)

- Plethora of different Linux distributions (Ubuntu, Fedora, CentOS, Scientific Linux, …)
- The material presented in this course will be demonstrated on Ubuntu, but no worries... everything applies also to any other Linux distribution (as we are covering only the core Linux functionalities!)

Getting Ubuntu for Windows users

- If you have on your laptop only Windows, you could either:
  - install Ubuntu from the following link https://www.microsoft.com/en-us/p/ubuntu/9nblggh4msv6?activetab=pivot:overviewtab



Ubuntu

Canonical Group Limited • Developer tools > Utilities

♡ Wish list

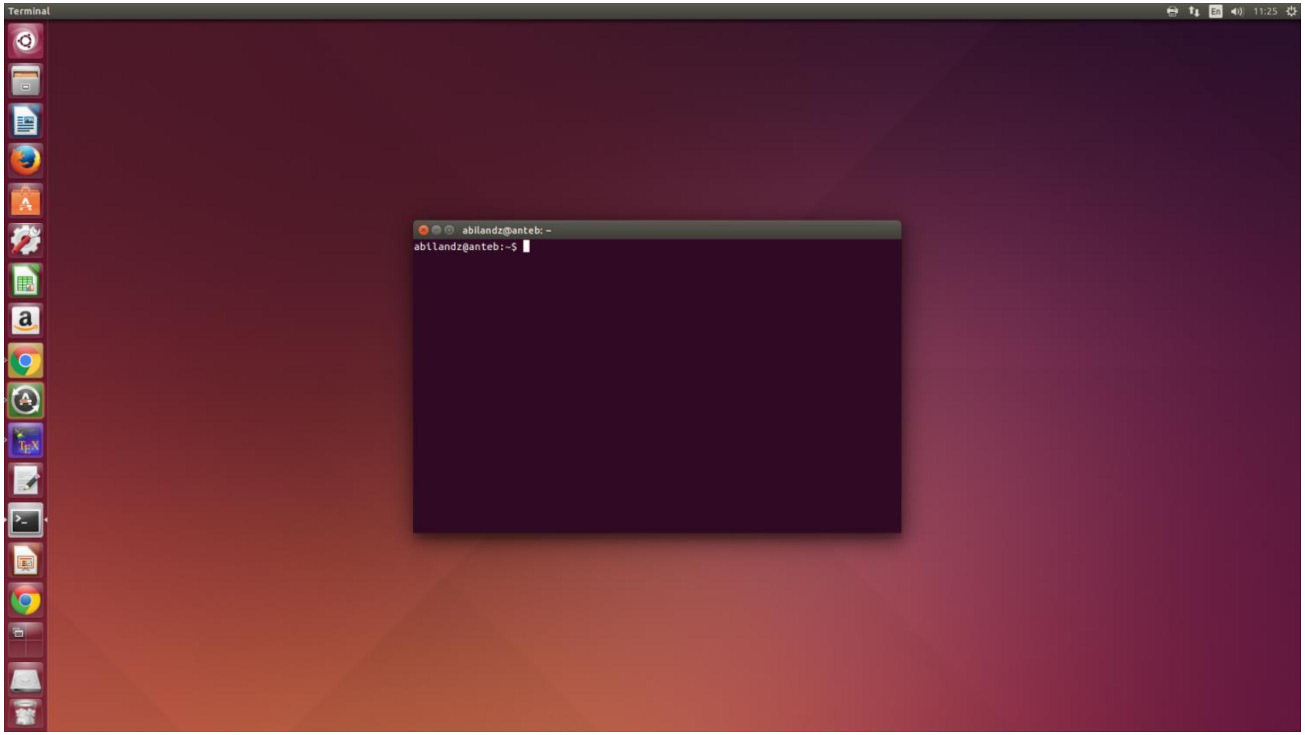
★★★★☆ 247

Ubuntu on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

  - install PuTTY (https://www.putty.org/) and then use it to connect to some machine running Linux, on which you have an account with access rights

Free advert (#2)

- Is this a right course for you?
  - If, after you have opened a terminal in Linux, …



```
abilandz@anteb:~$
```

… you have asked yourself: ‘What now?’, then this is the right course for you!

Free advert (#2)

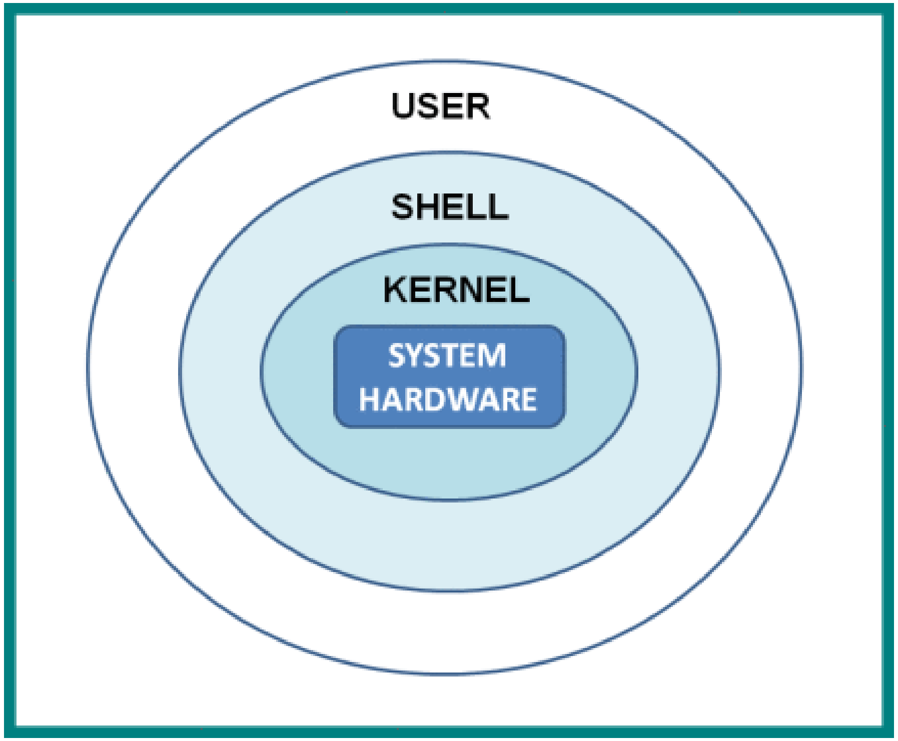
- What we can do in the terminal?
  - Not that much with the mouse…
- Next, you can start typing and pressing ‘Enter’, but especially if you do it for the first time most likely whatever you have typed in the terminal will produce only the error messages
  - Still, that is something, as it clearly means that there is some secret/magic language which is trying to respond to, or to interpret, your command input, as soon as you have typed something in the terminal and pressed ‘Enter’. What is that secret built-in language available in the terminal?

Linux shells

- Loosely speaking, shell is any program that user employs to type commands in the terminal (text window)
- Example shells:
  - sh
  - bash
  - ksh
  - csh
  - fish
  - PowerShell (developed by Microsoft!)
- Since ‘bash’ is the default shell on most Linux distributions nowadays, we focus on it
  - If not set by default, just type ‘bash’ in the terminal, and you are in the ‘bash’ wonderland

Why shell?

- The shell translates the commands you type into a format which the computer can understand
  - It works both ways: User is shielded from Linux kernel, and Linux kernel is shielded from user



A bit of ‘bash’ history

- Written by Brian Fox in 1989... And it’s still alive!!
- ‘bash’ is an acronym for ‘Bourne-again shell’ (the original shell was written in 1977 by Stephen Bourne)
- Written entirely in C
- Executable: /bin/bash
- File extension: .sh
- Command processor / interpreted / scripting language

‘bash’ current status

- ‘bash’ is well maintained and still under development
  - Webpage: https://www.gnu.org/software/bash/
  - Source code: http://git.savannah.gnu.org/cgit/bash.git
- Latest release: version 5.0 (January 7, 2019)
  - The current main maintainer: Chet Ramey

Testing ‘bash’ code online

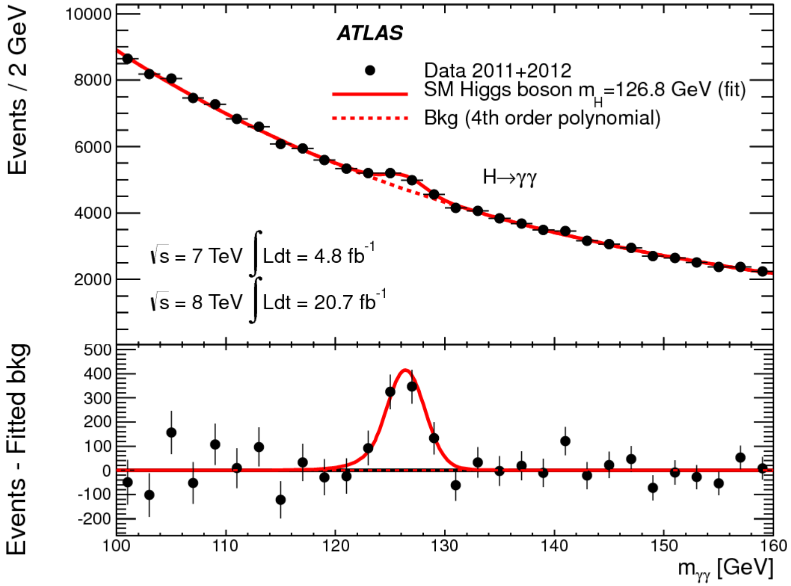
- In the case you do not have currently the access to the machine running Linux, you can test your ‘bash’ code online
  - For instance: https://www.tutorialspoint.com/execute_bash_online.php
- Use this link only as a temporary solution, as this is not a development environment

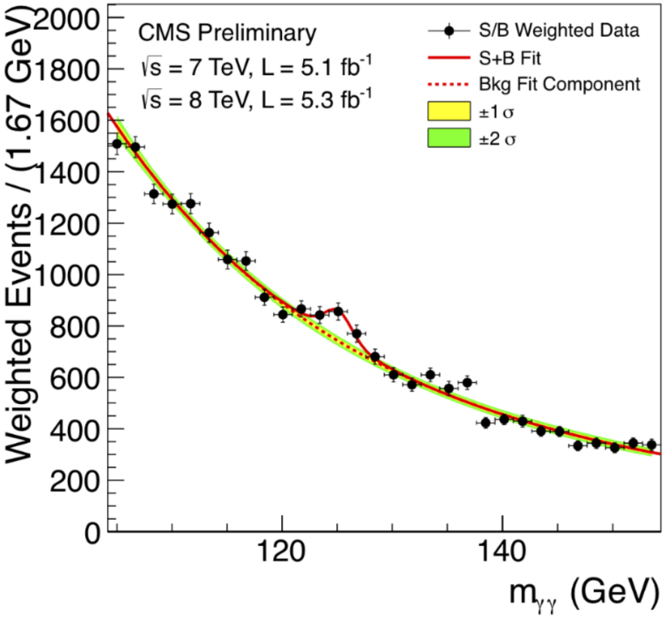
Interpreted vs. compiled languages

- Interpreted:
  - write code & execute line-by-line
  - less reliable (there is no compiler to catch the errors!)
  - source code can be easily read and copied
  - Examples: bash, Python, Mathematica
- Compiled:
  - write code & compile & execute the compiled file (‘binaries’)
  - generally runs faster than interpreted code
  - Examples: C, C++, Java

Free advert (#3)

- Is this a right course for you? ROOT, what's that?
  - If you have ever asked yourself what is the software in which the Higgs discovery plots were actually made…



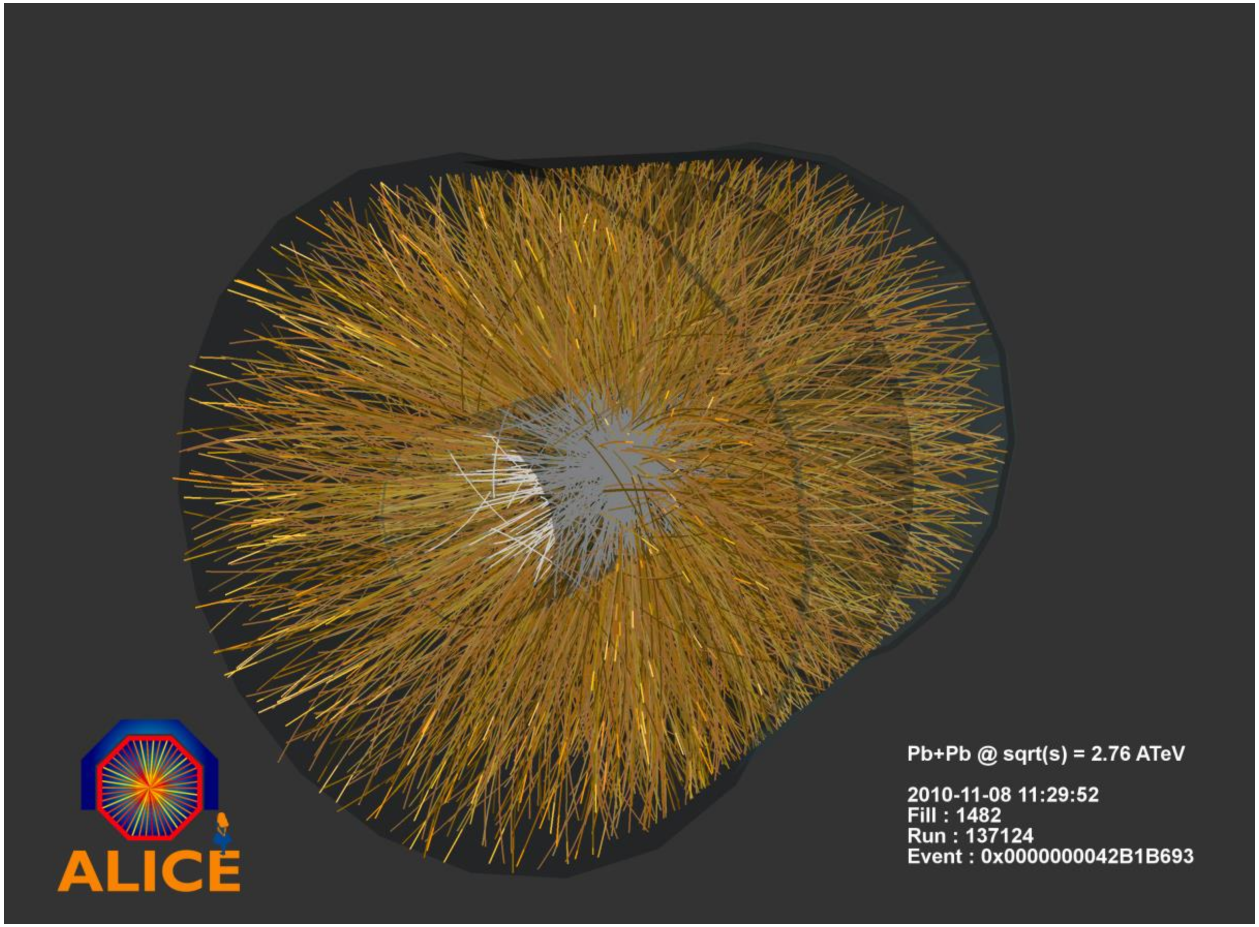


… then this is the right course for you!

  - At the very basic level, we can use ROOT for plotting, histogramming, fitting, etc.

Free advert (#3)

- This is the typical heavy-ion event at Large Hadron Collider reconstructed by ALICE Collaboration

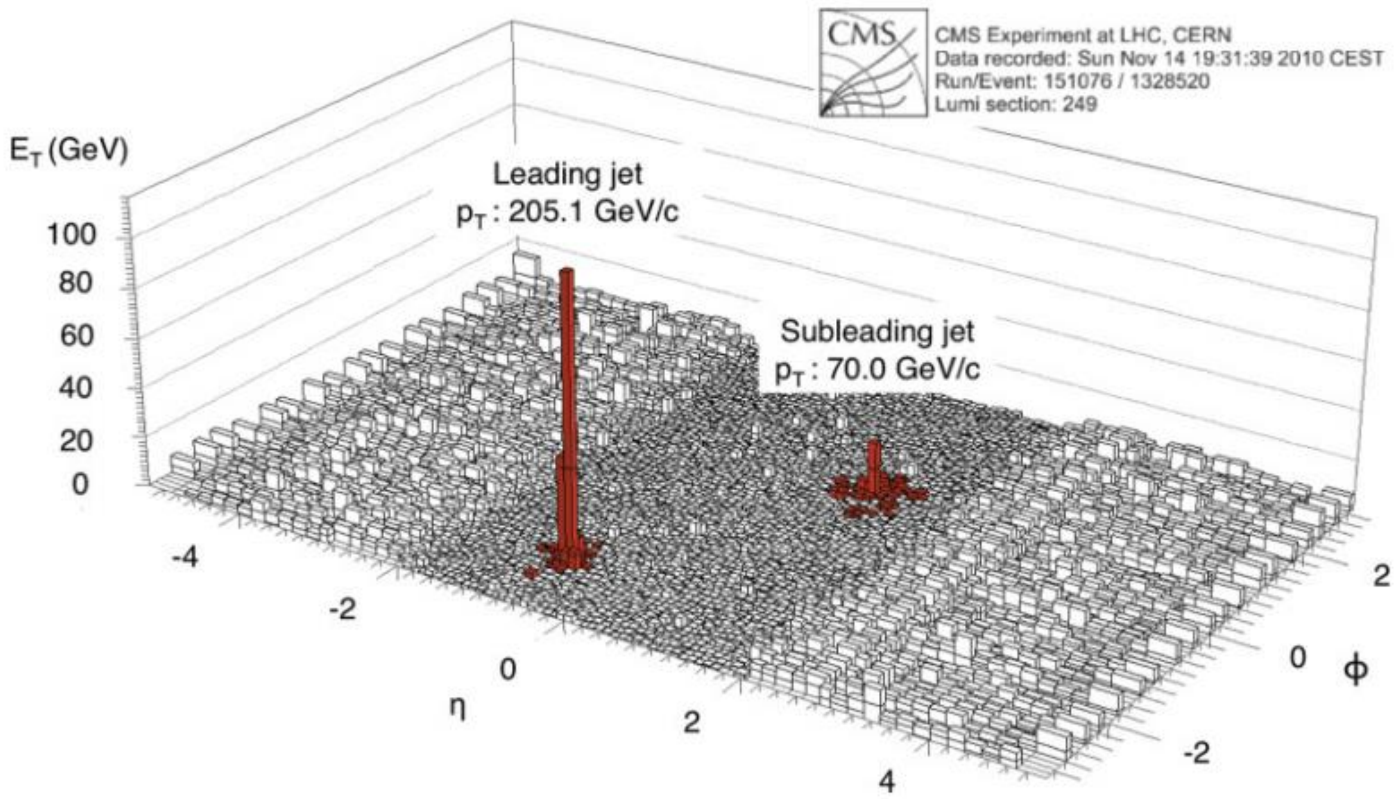


Free advert (#3)

- Trajectories of more than 10000 particles are reconstructed by AliROOT (C++ code specific to ALICE Collaboration built on top of ROOT)
- Most important major collaborations worldwide in high-energy physics currently use ROOT
  - Also the future ones (e.g. CBM at GSI, which will start data taking in 2025, is developing CbmRoot)

Free advert (#4)

- In terms of histogramming quality and performance, it's difficult to beat ROOT...



ROOT

- Object-oriented framework, written mostly in C++ and developed at CERN, for data analysis in high-energy physics
- The development was initiated by René Brun and Fons Rademakers in 1994, and is still under active development
  - Latest release: Version 6.18.04 (September 11, 2019)
- Webpage: https://root.cern.ch/
- Root forum: https://root-forum.cern.ch/

European Research Council
Established by the European Commission

Thanks!