Technological foundations of software development

Maxime Lefrançois – maxime.lefrancois@emse.fr – Version 2024

Table of Contents

ICM – Computer Science Major – Course unit on Technological foundations of Computer Science and M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development

Institut Henri Fayol - MINES Saint-Étienne

Master your working environment

The purpose of this session is to ensure that you are familiar with your computer, your operating system, and the shell-like command line programming environment.

Slides

- **lecture-1.pptx**
- **lecture-1.pdf**
- **lecture-1-2x2.pdf**

Technological f
software de

Pointers

- cheatsheet devhints https://devhints.io/bash
- simplified `man` https://tldr.ostera.io/
- Bash homepage https://www.gnu.org/software/bash/

- zsh homepage https://zsh.sourceforge.io/
- *Programmation shell sous Unix/Linux - ksh, bash, Bourne shell (avec exercices corrigés) (6e édition)*, Christine Deffaix Ré, éditions ENI, Septembre 2019, 978-2-409-02072-8

Lecture Notes

- Reminders - computer
  - CPU
  - CPU vs GPU
  - RAM vs ROM
  - Peripherals and I/O ports
- Operating System
  - Definition
  - Usage share, Smartphones, Supercomputers
  - Unix-like OS vs Windows
- Shell - Console - Terminal
- Windows
  - Programs location
  - Windows environment variables
  - Windows registry
  - Windows Subsystem for Linux
- Unix-like OS
  - Programs location
  - Shell configuration
  - Environment variables
  - Run a program in the console
  - Program Command Line Interface
  - Fundamental Linux Principles
  - The standard file system hierarchy
  - The command prompt
  - Navigating the file system
  - The permissions system
  - Assigning a variable and Parameter expansion
  - Command return values
  - Standard input and output of controls
  - Communication pipes
  - Globbing
  - Control structures

- Conditions
- Invoke a script
- Top 50 Unix commands

TODOs

By Oct. 15th

NOTE

submit your work for Courses 1-2 as *LASTNAME.zip* to https://ecampus.emse.fr/mod/assign/view.php?id=33633 (expiration date/time: `2023-10-16 01:00`)

WARNING

If you are using the Windows operating system, you must ensure that you have installed the Windows Subsystem for Linux (https://docs.microsoft.com/en-us/windows/wsl/install), and that you run commands from the WSL2 unix shell.

NOTE

Windows users, I highly encourage you to read the following tutorials about WSL: Set up a WSL development environment (https://learn.microsoft.com/en-us/windows/wsl/setup/environment), Get started using Visual Studio Code with Windows Subsystem for Linux (https://learn.microsoft.com/en-us/windows/wsl/tutorials/wsl-vscode), Get started with GPU acceleration for ML in WSL (https://learn.microsoft.com/en-us/windows/wsl/tutorials/gpu-compute), Run Linux GUI apps on the Windows Subsystem for Linux (https://learn.microsoft.com/en-us/windows/wsl/tutorials/gui-apps), Getting started with Linux and Bash (https://learn.microsoft.com/en-us/windows/wsl/tutorials/linux)

- 1/5 points - Describe the computer you are using in terms of hardware and operating system(s).
- 1/5 points - You have installed a few useful programs, and can call run successfully the following commands **in a unix shell**. **NOTE**: For each of these programs, you may need to ensure it is found in the `PATH`, and potentially set other environment variables, such as `JAVA_HOME` for maven:
  - git (https://git-scm.com/) - `git --version`
  - a version of the Java JDK (https://en.wikipedia.org/wiki/Java_Development_Kit) - `javac --version`
  - Apache Maven (https://maven.apache.org/) - `mvn --version`
  - Make (https://www.gnu.org/software/make/) - `make --version`
  - Python version above 3.10 - `python --version`
  - Docker Desktop - `docker --version`
- 1/5 points - install ollama (https://ollama.com/), following instructions in the online documentation (https://github.com/ollama/ollama/tree/main/docs). Pull the `llama3.1` (https://ollama.com/library/llama3.1) language model (or some lighter language model).
- 0.5/5 points - Use the ollama api (https://github.com/ollama/ollama/blob/main/docs/api.md) to query the model using the `curl` program.
- 0.5/5 points - Pipe the output of the command above with the jq (https://jqlang.github.io/jq/) command-line JSON processor to print out the response content.
- 0.5/5 points - find and install a command-line program to extract the text content of some PDF file
- 0.5/5 points - Write a bash script `ask_pdf.sh` that satisfies the documentation below:

```
USAGE:
  ./ask_pdf.sh <pdf-file> "<question>"

DESCRIPTION:
  ask_pdf.sh is a command-line tool that answers a specific question based on
  the content of the provided PDF document.

PARAMETERS:
  <pdf-file>
    The full path or name of the PDF document to analyze. This parameter is
    required.

  "<question>"
    The question to ask about the PDF content. This parameter is required and
    must be enclosed in quotes.
```

WARNING

> To earn the points, you need to demonstrate everything works as expected by including a short screen recoding video (<5min), in addition to the sources of your scripts.

NOTE

> language models can clearly help you get part of the points, but I want to be sure it worked on your computer, and that you provide explanations for each component of the answer.

Version 2024
Last updated 2024-09-13 14:08:46 +0200