

# Betriebssysteme

Labor 1

# Inhalt

- Organisatorische Aspekte
- Setup
- UNIX- Befehle zum Arbeiten mit Ordner
- Aufgaben

# Organisatorische Aspekte

- **Endnote:**
  - **20% Labor**
  - **40% praktische Prüfung** (Woche 14)
  - **40% schriftliche Prüfung** (in der Prüfungszeit)
- **Minimale Leistungsstandards:**
  - **90% Anwesenheit** (maximal 2 UNMOTIVIERTE Abwesenheiten)
  - Endnote im Labor  $\geq 5$  (kann in der Nachprüfungszeit NICHT wiederholt werden)
  - Praktische Prüfung  $\geq 5$  (kann in der Nachprüfungszeit NICHT wiederholt werden)
- **E-Mail:**
  - `madalina.dicu@ubbcluj.ro`

# Organisatorische Aspekte

- Laboraufgaben:
  - jeder Student wird eine zugewiesene Aufgabe erhalten (via Datei mit Anwesenheiten und Noten)
  - die Aufgaben werden durch eine Datei mitgeteilt (sowohl auf Teams als auch auf Moodle)
  - für jede Aufgabe ist ein Zeitraum von 2 Wochen zur Lösung vorgesehen
  - die Lösungen der Aufgaben müssen auf Teams hochgeladen werden (Assignment auf Teams), aber diese müssen während des Laborunterrichts persönlich vorgestellt werden, sonst werden diese nicht bewertet.

# Organisatorische Aspekte

- **Bewertungssystem:**
  - **0** -> eine andere Aufgabe wird gelöst/Assignment nicht abgegeben
  - **1** -> man kann keine Fragen zum Code beantworten /Shell-Skripte weisen Syntaxfehler auf
  - **4** -> Laufzeitfehler (runtime crash/error)
  - **5–9** -> alle oben genannten Punkte sind in Ordnung, aber die Implementierung ist noch nicht vollständig
  - **10** -> vollständige Implementierung und korrekte Ausführung

# Organisatorische Aspekte

- **Sonstiges:**
  - Eine Verzögerung von 2 Wochen bei der Übergabe der Aufgaben ist zulässig  
**(-2 Punkte/Woche)**
  - **Note 12:** vollständige Implementierung und korrekte Ausführung der Aufgabe während des Laborunterrichts
  - **Note 11:** vollständige Implementierung und korrekte Ausführung der Aufgabe eine Woche vor dem Termin

# Verbindung zu einem Unix / Linux-Server

Um auf einem Computer mit einem Unix-Betriebssystem zu arbeiten, können wir:

- Direkt auf einem Computer arbeiten, der ein solches Betriebssystem verwendet
- Eine Remote Verbindung über das Netzwerk herstellen
  
- **Server-Maschine** = die Unix-Maschine, mit der wir uns verbinden
- **Kunden-Maschine** = der Computer, von dem aus wir eine Verbindung herstellen
- **SSH-Kunde** = das Programm, mit dem wir uns verbinden (Die Kommunikation mit dem Server erfolgt über ein Protokoll namens **SSH**)

# Setup

- **Windows:**
  - -> WSL 2 (**Windows Subsystem for Linux**) oder eine virtuelle Maschine installieren
- **Linux/MacOS:**
  - -> Your're lucky! (Ihr müsst nichts tun)



# UNIX- Befehle zum Arbeiten mit Ordner

- **man**: wir können die Informationen im Handbuch abrufen und als Textausgabe auf unserem Bildschirm anzeigen.
  - Beispiel:* `$ man ls`
  - für die Ausgabe: **q**
- **ls**: listet Dateien und Verzeichnisse im Dateisystem auf und zeigt detaillierte Informationen dazu an.
  - `ls [OPTIONS] [FILES]`
  - Wichtig: `ls -l -d -a -p`
- **mkdir**: ermöglicht Benutzern das Erstellen neuer Verzeichnisse.
  - `mkdir` steht für "make directory".
  - `mkdir [option] dir_name`
- **rmdir**: entfernt leere Verzeichnisse aus dem Dateisystem.
  - `rmdir` steht für "remove directory".

# UNIX- Befehle zum Arbeiten mit Ordner

- **cd**: wird verwendet, um das aktuelle Arbeitsverzeichnis zu ändern.
  - steht für "change directory".
  - `$ cd ..` -> eine Ebene höher als das aktuelle Verzeichnis
- **pwd**: gibt den Pfad des Arbeitsverzeichnisses ausgehend vom Stammverzeichnis aus.
  - steht für "**P**rint **W**orking **D**irectory".
- **passwd**: wird verwendet, um die Benutzerkontokennwörter zu ändern.
- **file**: zeigt den Typ einer Datei an.
- **ps**: listet die aktuell ausgeführten Prozesse auf und zeigt Informationen zu diesen Prozessen an.
- **df**: wird verwendet, um Informationen zu Dateisystemen über den gesamten Speicherplatz und den verfügbaren Speicherplatz anzuzeigen.
  - steht für "disk free"

# UNIX- Befehle zum Arbeiten mit Ordner

- **du** : wird verwendet, um die Speicherplatznutzung abzuschätzen.
  - steht für "disk usage"
  - mit dem Befehl **du** können die Dateien und Verzeichnisse verfolgt werden, die übermäßig viel Speicherplatz auf der Festplatte beanspruchen.
- **rm** : entfernt jede in der Befehlszeile angegebene Datei. Standardmäßig werden keine Verzeichnisse entfernt.
- **mv** : wird verwendet, um eine oder mehrere Dateien oder Verzeichnisse von einem Ort an einen anderen im Dateisystem zu verschieben
  - steht für "move"
  - Es hat zwei verschiedene Funktionen:
    - es benennt eine Datei oder einen Ordner um.
    - es verschiebt eine Gruppe von Dateien in ein anderes Verzeichnis.
    - Wichtig: **mv -i -f**
- **cat** : es liest Daten aus der Datei und gibt deren Inhalt als Ausgabe an.

# UNIX- Befehle zum Arbeiten mit Ordner

- **diff**: wird verwendet, um die Unterschiede in den Dateien anzuzeigen, indem die Dateien Zeile für Zeile verglichen werden.
  - steht für "difference"
  - es sagt uns, welche Zeilen in einer Datei geändert werden müssen, um die beiden Dateien identisch zu machen.
- **find**: ist ein Befehlszeilenprogramm zum Durchlaufen einer Dateihierarchie.
  - Es kann verwendet werden, um Dateien und Verzeichnisse zu finden und nachfolgende Operationen an ihnen durchzuführen.
- **ln**: wird verwendet, um Verknüpfungen zwischen Dateien zu erstellen.
- **cp**: wird zum Kopieren von Dateien, Dateigruppen oder Verzeichnissen verwendet. Es wird ein genaues Image einer Datei auf einer Festplatte mit einem anderen Dateinamen erstellt.
  - steht für "copy"
  - `$ cp Src_file Dest_file`
  - Wichtig: `cp -r -i -f`

# UNIX- Befehle zum Arbeiten mit Ordner

- **chmod**: legt die Berechtigungen für Dateien oder Verzeichnisse fest. (lesen, schreiben und ausführen/ read, write, execute)
  - steht für "change mode"
  - `chmod options permissions file name`
  - permissions definiert die Berechtigungen für den Eigentümer der Datei (den "Benutzer/user"), die Mitglieder der Gruppe, der die Datei gehört (die "Gruppe/group"), und alle anderen ("andere/others").
- **less**: Dienstprogramm, mit dem der Inhalt einer Textdatei eine Seite (ein Bildschirm) pro Zeit gelesen werden kann.
  - der Zugriff ist schneller, da bei großen Dateien nicht auf die vollständige Datei zugegriffen wird, sondern Seite für Seite.
- **more**: wird verwendet, um die Textdateien in der Eingabeaufforderung anzuzeigen und jeweils einen Bildschirm anzuzeigen, falls die Datei groß ist.
  - ermöglicht dem Benutzer, durch die Seite nach oben und unten zu scrollen.

# UNIX- Befehle zum Arbeiten mit Ordner

- **jobs**: zeigt den Status der im aktuellen Terminalfenster gestarteten Jobs an.
  - Jobs werden für jede Sitzung ab 1 nummeriert. Die Job-ID-Nummern werden von einigen Programmen anstelle von PIDs (=process ID) verwendet (z. B. von den Befehlen fg und bg).
- **fg**: wird verwendet, um einen Hintergrundjob in den Vordergrund zu stellen.
- **bg**: wird verwendet, um Vordergrundjobs im Hintergrund zu stellen.
- **kill**: ist ein integrierter Befehl, mit dem Prozesse manuell beendet werden
  - sendet ein Signal an einen Prozess, der den Prozess beendet.
  - wenn der Benutzer kein Signal angibt, das zusammen mit dem Befehl kill gesendet werden soll, wird das Standard-TERM-Signal gesendet, das den Prozess beendet.

# Aufgaben

1. Erstellen Sie in euren persönlichen Ordner die folgende Struktur von Ordner und Dateien:

```
(persönliche Ordner)
|
+-- abc
|   +-- x (Datei)
|   +-- y (Datei)
|   +-- t1 (Datei)
|   +-- t2 (Datei)
|   +-- t3 (Datei)
|   +-- t (Ordner)
|       +-- a (Datei)
|       +-- b (Datei)
|
+-- zz (Ordner)
|   +-- x (Datei)
|
+-- tt (Ordner)
```

# Mehr Infos

- <https://www.youtube.com/watch?v=AmcSFr8wVt4>
- <https://www.cs.ubbcluj.ro/studenti/examen-de-licenta-si-disertatie/manuale-si-mo-dele-de-subiecte-pentru-examenul-de-licenta/>