

# Einführung

## Linux-Kurs der Unix-AG

Zinching Dang

25. April 2017



# Übersicht

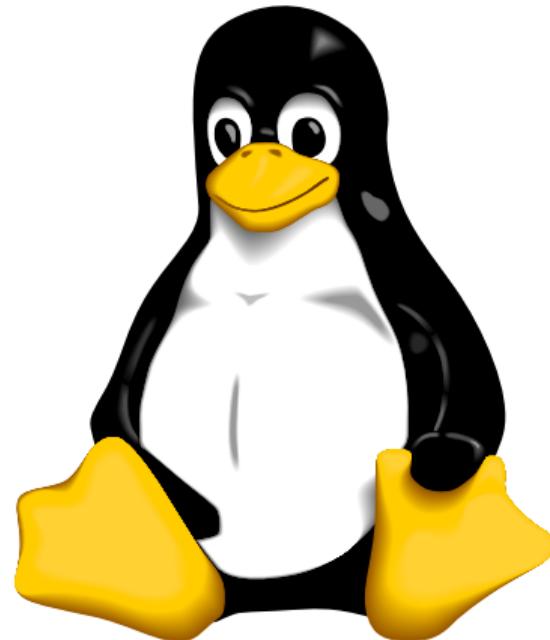
Was ist Linux?

Wo wird Linux eingesetzt?

Geschichte

Erste Schritte

Zusammenfassung & Ausblick



Das Linux-Maskottchen Tux

# Was ist Linux?

## Linux ist

- ▶ ein freier und verbreiteter Betriebssystemkern (Kernel)
- ▶ „open source“: frei zugänglicher Quelltext
- ▶ zusammen mit Anwendungen als Distribution gebündelt
- ▶ vielseitig: zahlreiche Distributionen für unterschiedliche Bedürfnisse

# Wo wird Linux eingesetzt?

## Linux findet man

- ▶ auf Desktop- und Hochleistungsrechnern
- ▶ auf Smartphones und in Unterhaltungselektronik  
(z. B. Tablets, Mediacenter)
- ▶ als Mail- und Webserver
- ▶ als Steuerungsrechner in der Industrie (Embedded System)

## Die Entstehung von Unix

- ▶ 1969: Entwicklung des Betriebssystem Unix (**Uniplexed Information and Computing Service, Unics**) bei Bell Labs
- ▶ 1970er: kostenlose Verbreitung des Quellcodes
- ▶ 1980er: Kommerzialisierung von Unix
- ▶ 1983: Gründung des GNU-Projekts (**GNU's Not Unix**), um freie Anwendungen und ein Betriebssystem zu entwickeln

# Geschichte

## Die Entstehung von Linux

- ▶ 1991: Linux wird als Hobby-Projekt veröffentlicht:

*Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby,  
won't be big and professional like gnu) for 386(486) AT clones. [...]*

*I've currently ported bash(1.08) and gcc(1.40), and things seem to work. [...]*

*Linus (torvalds@kruuna.helsinki.fi)*

*PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable [sic]  
(uses 386 task switching etc), and it probably never will support anything other than  
AT-harddisks, as that's all I have :-(.*

– Linus Torvalds in comp.os.minix, 26. August 1991

# Geschichte

## Linux heute

- ▶ sehr große Community (>13500 Entwickler)
- ▶ wird teils durch professionelle Entwickler vieler verschiedener Firmen (>1300) vorangetrieben
- ▶ auf fast jede Computer-Platform portiert

# Erste Schritte

## Die Kommandozeile unter Linux

- ▶ wird „Shell“ genannt
- ▶ Unterscheidung zwischen „Konsole“ und „Terminal“
- ▶ der „Prompt“ zeigt an, dass die Shell auf Befehle wartet

## Der Prompt

```
linux-kurs@ubuntu:~$
```

Benutzer Rechner aktueller Pfad

# Befehle in der Shell

## Vorgehensweise

- ▶ Befehl eingeben und mit <Enter> bestätigen
- ▶ Ausgabe betrachten, falls vorhanden

## Verzeichnisinhalt auflisten

```
linux-kurs@ubuntu:~$ ls ← Befehl
```

# Befehle in der Shell

## Vorgehensweise

- ▶ Befehl eingeben und mit <Enter> bestätigen
- ▶ Ausgabe betrachten, falls vorhanden

## Verzeichnisinhalt auflisten

```
linux-kurs@ubuntu:~$ ls
Arbeitsfläche   Dokumente   Musik      Vorlagen
Bilder          Downloads   Videos     }
```

Ausgabe

# Befehle in der Shell

## Allgemeines zu Befehlen

- ▶ haben (manchmal optionale) Argumente
- ▶ können durch Optionen ihr Verhalten ändern

## Verzeichnisinhalt auflisten mit Argument und Optionen

```
linux-kurs@ubuntu:~$ ls -hl Dokumente/ ← Argument  
                           ↓  
                           Optionen
```

# Befehle in der Shell

## Allgemeines zu Befehlen

- ▶ haben (manchmal optionale) Argumente
- ▶ können durch Optionen ihr Verhalten ändern

## Verzeichnisinhalt auflisten mit Argument und Optionen

```
linux-kurs@ubuntu:~$ ls -hl Dokumente/
insgesamt 12K
-rw-r--r-- 1 linux-kurs user 4K Nov 6 Hello
-rw-r--r-- 1 linux-kurs user 4K Nov 6 World!
```

Option -hl

Ausgabe

Inhalt des Verzeichnisses Dokumente

# Befehle in der Shell

## Allgemeines zu Befehlen

- ▶ Argumente und Optionen sind nicht immer notwendig
  - ▶ siehe: `ls` und `ls -lh Dokumente`
- ▶ Argumente sind Ausführungsgegenstand des Befehls
  - ▶ siehe: `ls` und `ls Dokumente`
- ▶ mehrere Optionen können zusammengefasst werden
  - ▶ `ls -l -h` wird zu `ls -lh`
- ▶ mit Optionen können andere Funktionalitäten genutzt werden
  - ▶ siehe: `ls` und `ls -lh`

# Befehle in der Shell

## Allgemeines zu Befehlen

- ▶ Groß- und Kleinschreibung ist wichtig
- ▶ Tab-Completion nutzen, um Tippfehler zu vermeiden
  - ▶ <Tab>-Taste drücken, um die Auto-Vervollständigung zu nutzen
  - ▶ nur möglich, falls Eingabe bis dorthin eindeutig
  - ▶ 2x<Tab> listet Möglichkeiten auf, falls nicht eindeutig
- ▶ Benutzung der Pfeiltasten
  - ▶ Pfeil links/rechts, um Fehler zu korrigieren
  - ▶ Pfeil hoch, um ausgeführte Befehle erneut aufzurufen

# Dateisystem-Hierarchie

## Verzeichnisstruktur

- ▶ Wurzel-Verzeichnis /
- ▶ darunter Unterverzeichnisse
  - ▶ /home/ enthält Daten der Benutzer
  - ▶ /home/linux-kurs/ enthält Daten des Benutzers linux-kurs
- ▶ keine Laufwerksbuchstaben
- ▶ USB-Sticks, CDs, etc. werden als Verzeichnisse eingebunden

# Dateisystem-Hierarchie

## Pfadangaben

- ▶ gleich für Dateien und Verzeichnisse
- ▶ können auf verschiedene Arten angegeben werden
- ▶ **Absoluter Pfad:** vom Wurzelverzeichnis / ausgehend
  - ▶ /home/linux-kurs/Bilder/tux.png
- ▶ **Relativ Pfad:** relativ zum aktuellen Verzeichnis
  - ▶ aktuelles Verzeichnis sei: ~/
  - ▶ Bilder/tux.png

# Weitere Befehle

## Verzeichnis wechseln

- ▶ `cd` wechselt in ein anzugebendes Verzeichnis
- ▶ ohne Argument: Wechsel in das eigene Home-Verzeichnis

## Beispiel

```
linux-kurs@ubuntu:~$ cd Dokumente/
```

~ entspricht dem eigenem Home-Verzeichnis /home/linux-kurs/

# Weitere Befehle

## Verzeichnis wechseln

- ▶ `cd` wechselt in ein anzugebendes Verzeichnis
- ▶ ohne Argument: Wechsel in das eigene Home-Verzeichnis

## Beispiel

```
linux-kurs@ubuntu:~$ cd Dokumente/  
linux-kurs@ubuntu:~/Dokumente$ ls  
Hello World!  
linux-kurs@ubuntu:~/Dokumente$
```

# Weitere Befehle

## Verzeichnisse erstellen und löschen

- ▶ `mkdir` erstellt ein Verzeichnis
- ▶ `rmdir` löscht ein leeres Verzeichnis
- ▶ Argument: zu erstellendes bzw. löschendes Verzeichnis

## Beispiel

```
linux-kurs@ubuntu:~$ mkdir Dokumente/Uni  
linux-kurs@ubuntu:~$ ls Dokumente  
Hello Uni World!  
linux-kurs@ubuntu:~$
```

# Weitere Befehle

## Verzeichnisse erstellen und löschen

- ▶ `mkdir` erstellt ein Verzeichnis
- ▶ `rmdir` löscht ein leeres Verzeichnis
- ▶ Argument: zu erstellendes bzw. löschendes Verzeichnis

## Beispiel

```
linux-kurs@ubuntu:~$ rmdir Dokumente/Uni  
linux-kurs@ubuntu:~$ ls Dokumente  
Hello World!  
linux-kurs@ubuntu:~$
```

# Weitere Befehle

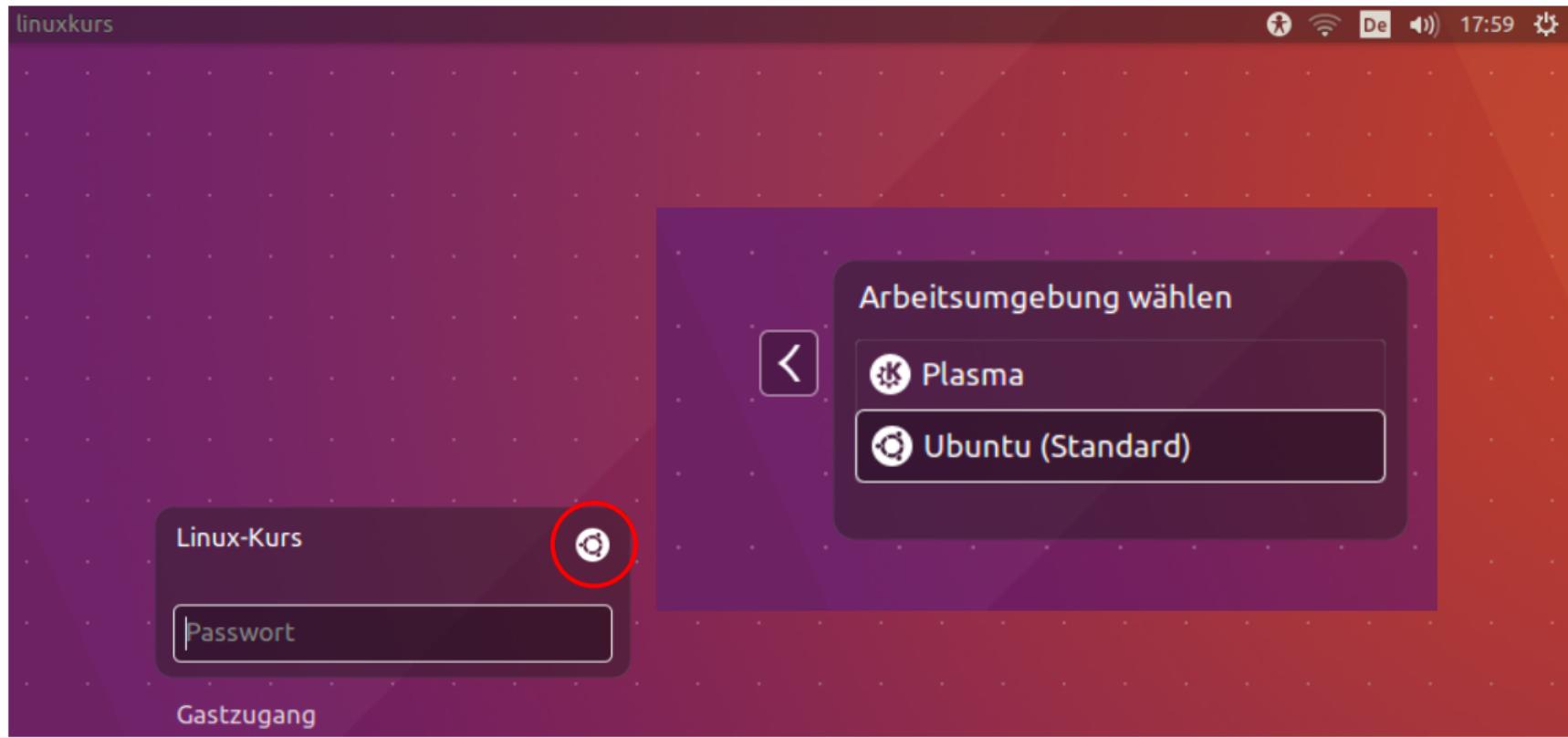
## Dateien editieren

- ▶ `nano` ist ein einfacher Kommandozeileneditor
- ▶ Argument: Dateiname
  - ▶ bearbeitet eine Datei, falls der angegebene Dateiname existiert
  - ▶ andernfalls wird eine neue Datei angelegt

## Bedienung von Nano

- ▶ Text eingeben
- ▶ <STRG+O>: Speichern (ggf. Dateinamen angeben/bestätigen)
- ▶ <STRG+X>: Beenden

# Benutzeroberflächen auswählen



# Zusammenfassung & Ausblick

## Zusammenfassung

- ▶ Einblick in die Entstehung von Linux
- ▶ Erste Schritte auf der Kommandozeile
- ▶ Grundlagen zum Bedienkonzept

## Nächstes Mal

- ▶ Weitere grundlegende Shellbefehle
- ▶ Denkweisen unter Linux