

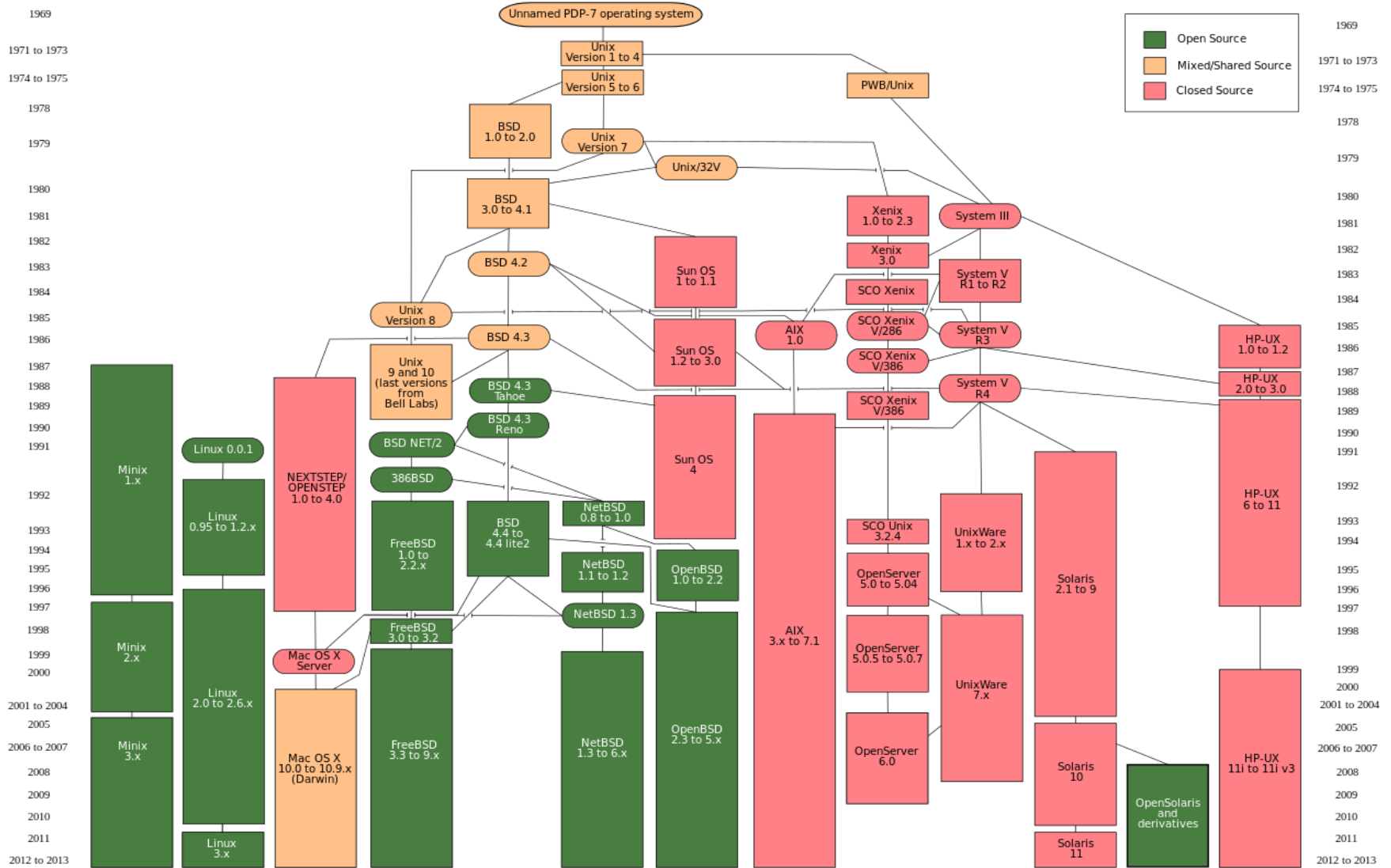
# Paradigmen der Programmierung

## Übung 1 Shell Basics

# Organisatorisches

- Donnerstags, 09:15-10:45, wöchentlich
- Keine Prüfungsvorleistungen, manchmal vielleicht trotzdem Hausaufgaben
- Informatik-Account notwendig (oder eigener Laptop)
- Fragen gerne an [stefan.luedtke2@uni-rostock.de](mailto:stefan.luedtke2@uni-rostock.de)

# UNIX



# Shell

- Unix Benutzerschnittstelle
- “Von Programmierern für Programmierer”
- Viele kleine Tools für spezielle Aufgaben

# Pfade

- Pfade in Unix haben Baum Struktur, Wurzel “/”
- z.B. home-Verzeichnis in /home/username
- Verzeichnisse Wechseln mit `cd`
- Abkürzungen für home, aktuelles, übergeordnetes Verzeichnis: `~ . ..`

# Einfache Tools

cd

echo

cat

cp

rm

mv

mkdir

touch

find

# Hilfe

man

apropos

# Wildcards

- \*
- ?
- [abc]
- [a-z]



# Kurze Übung

- Erstellen Sie ein Verzeichnis uebung1 in ihrem Home-Verzeichnis
- Wechseln Sie in dieses Verzeichnis
- Erstellen Sie eine Datei beispiel.txt in diesem Verzeichnis
- Verschieben Sie die Datei in Ihr Home-Verzeichnis
- Finden Sie alle Dateien in Ihrem Home-Verzeichnis, die mit .pdf enden
- Finden Sie alle Verzeichnisse in Ihrem Home-Verzeichnis

# Zugriffsrechte

`ls -l` zeigt Zugriffsrechte z.B. so an:

`-rw-r--r--`

user group others

Änderung mit `chmod`

`chmod {ugoa} {+-=} {rwx}`

**z.B.** `chmod ug+wx`

# Zugriffsrechte (für Nerds)

- `chmod` versteht auch Oktal-Argumente
- Recht: r-w-x  
Wert: 4-2-1
- Beispiel: `chmod 644` (Eigentümer darf lesen und schreiben, alle anderen nur lesen)
- Was bedeutet `chmod 754`?

# Aufgabe Zugriffsrechte

- Ändern Sie die Zugriffsrechte der Datei beispiel.txt, sodass die Datei von beliebigen Usern gelesen, geschrieben und ausgeführt werden kann
- Fügen Sie noch einige weitere Dateien zum Ordner uebung1 hinzu
- Ändern Sie die Zugriffsrechte aller Dateien im Ordner uebung1, die auf .txt enden, sodass diese von allen Usern gelesen und geschrieben werden können (Tipp: find und Parameter -exec)

# Ein- und Ausgabe

- `stdin` – Standardeingabe, normalerweise über die Tastatur
- `stdout` – Standardausgabe, normalerweise direkt auf der Kommandozeile

# Umlenken von Datenströmen

- `>` – stdout in Datei umlenken (überschreiben)
- `>>` – stdout in Datei umlenken (anhängen)
- `<` – stdin aus Datei lesen
- `|` – stdout in stdin verbinden, Beispiel:
- `echo "Beispiel" | wc`

# Aufgabe

- Speichern Sie die Pfade der ersten 2 Dateien aus dem Verzeichnis uebung1, die auf .txt enden, in der Datei beispiel.txt

# Named Pipes

- Named Pipe anlegen mit  
`mkfifo pipe1`
- Programme können von dieser Pipe lesen und darauf schreiben, z.b.  
`ls -l > pipe1`



# Aufgabe

- `mkfifo pipe1`  
`cat < pipe1`
- Was passiert? Warum?
- Geben Sie in ein neues Terminal-Fenster ein:
- `ls -l > pipe1`
- Was passiert? Warum?
- Was passiert hier?
  1. Terminal: `cat < pipe1`
  2. Terminal: `cat > pipe1`