

# Verlässliche Echtzeitsysteme

Übungen zur Vorlesung

Entwicklung von Zuhause

Phillip Raffeck, Florian Schmaus, Simon Schuster

Friedrich-Alexander-Universität Erlangen-Nürnberg  
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)  
<https://www4.cs.fau.de>

Wintersemester 2020



## Fernzugriff auf die CIP-Rechner

- Zugriff via ssh
- von Windows-Rechnern aus: PuTTY (<https://putty.org>)

↪ Rechnerliste: [www.cip.cs.fau.de/cipPools/roomIndex.en.html](http://www.cip.cs.fau.de/cipPools/roomIndex.en.html)

## Entfernt arbeiten per ssh

- Per ssh anmelden: `ssh <nutzer>@cipXXX.cip.cs.fau.de`

- Inklusive Weiterreichung graphischer Anwendungen:

`ssh -X <nutzer>@cipXXX.cip.cs.fau.de`

*Hinweis:* Erfordert einen lokalen X-Server, generell nicht empfohlen

↪ Besser Xpra (<https://xpra.org>)

- Auf andere Nutzer prüfen: `who`
- Eigene laufende Prozesse: `ps aux`
- Rein graphische Alternative: Xpra ↪ Xpra-Foliensatz



## Terminal-Multiplexer

- Mehrere Terminals
- Im Hintergrund arbeiten lassen
- Session überlebt Verbindungsabbrüche

## Kurzeinführung tmux

- Starten: `tmux`
- Fenster (vertikal/horizontal) teilen: `Strg + b` dann `%` resp. `"`
- Navigieren zwischen Fensterteilen: `Strg + b` und Pfeiltasten
- Abkoppeln: `Strg + b` dann `d`
- Wieder verbinden: `tmux attach`
- Vorhandene Sitzungen auflisten: `tmux ls`
- Sitzungen bleiben im Hintergrund erhalten
- Im CIP-Pool: Prozesschecker beendet (tage)lang laufende Prozesse



### Mehr Informationen

- `man tmux`
- <https://www.hamvocke.com/blog/a-quick-and-easy-guide-to-tmux/>
- <https://robots.thoughtbot.com/a-tmux-crash-course>



## Einsatzzweck

- Hardware-Emulator
- Wir benutzen eCos in einigen Aufgaben.
- Plattform für eCos (x86)

## Verwendung

- `make run`
  - startet eCos im QEMU
  - serielle Ausgabe auf der Konsole
- `make gdb`
  - startet QEMU im Hintergrund
  - startet gdb-Session verbunden mit QEMU
- QEMU beenden: `Strg + a`, dann `x`

