TECHNISCHE UNIVERSITÄT
CHEMNITZ

Wintersemester 2017/2018
Algorithmen und Programmierung
Tutorium

AuP Survival Guide

Peter Tröger und Christine Jakobs

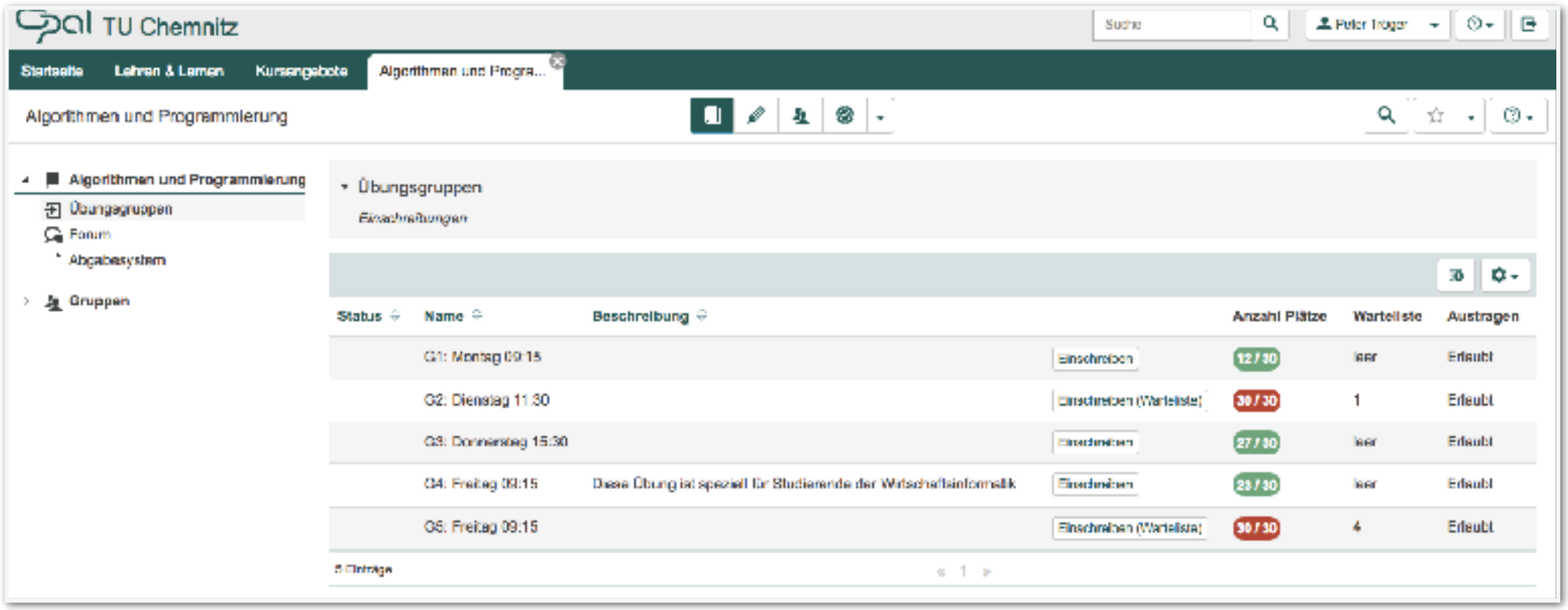
Operating Systems Group, TU Chemnitz

Aufbau

- **Vorlesung**
  - 90 Minuten Vortrag + Diskussion mit Prof. Werner, empfohlen
  - Ergänzende Notizen sinnvoll, Abschreiben der Folien unnötig
- **Übung**
  - Kleine Gruppe + Übungsleiter, empfohlen
  - Diskussion aktueller Vorlesungsinhalte
  - Passende Übungsaufgaben (Pseudo-Code, Algorithmen auf Papier)
- **Tutorium**
  - Zusätzliches Angebot, nur bei Bedarf nutzen
  - Diskussion beliebiger studentischer Fragen, entkoppelt von der Vorlesung
  - Richtige Ort für Probleme mit C / Python / Makefile
- **Trainings- und Prüfungsaufgaben** in selbständiger Arbeit

Übung

- Registrierung für eine der Übungsveranstaltungen in OPAL (https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/15259697155/CourseNode/96292953258489)
  - Dient lediglich der Aufteilung der Studenten
  - Keine Überprüfung auf Erscheinen in der ‚korrekten' Übung
- Verschiedene Übungsleiter, verschiedene Stile



Aufgaben

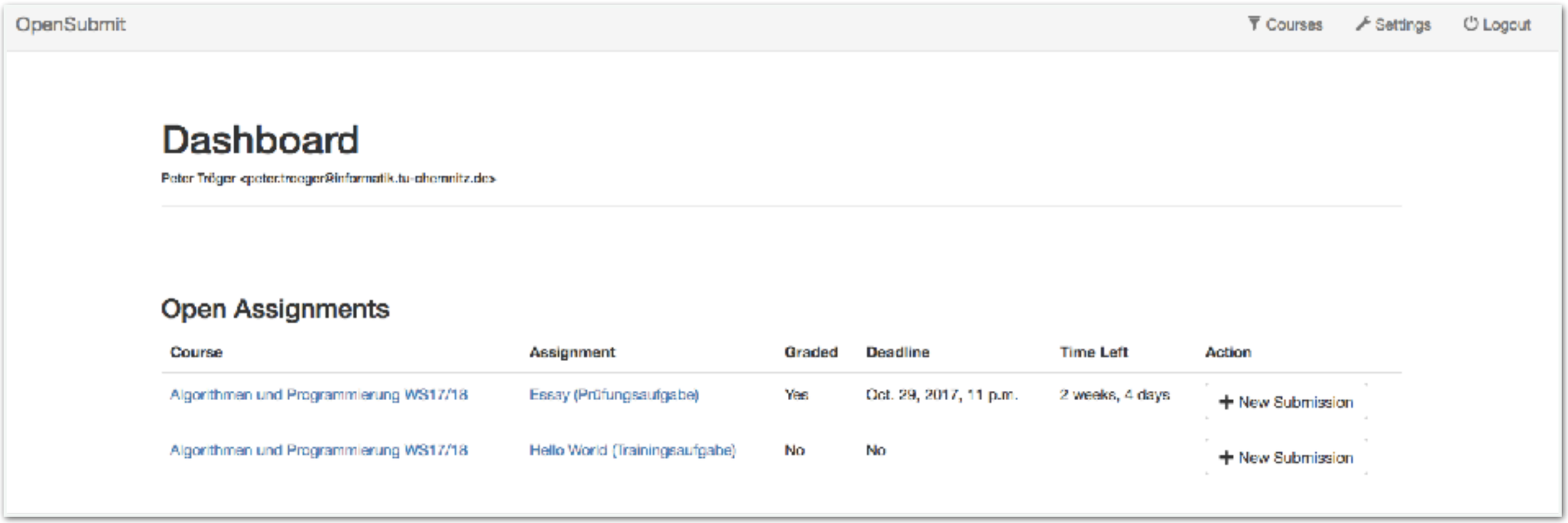
- **Übungsaufgaben**: Bringt der Übungsleiter mit
- **Trainingsaufgaben**: nicht bewertet, selbständige Bearbeitung, Einreichung online
  - Keine Evaluierung durch Menschen, automatisches Feedback
  - Stehen jederzeit zur Verfügung
- **Prüfungsaufgaben**: bewertet, selbständige Bearbeitung, Einreichung online
  - Evaluierung durch Menschen + automatisches Feedback
  - 7 große Aufgabenstellungen über das Semester verteilt
  - mind. 50% der Punkte -> Prüfung / PVL bestanden (s. Vorlesung)

Prüfungsaufgaben

- 2017
  - KW41 - KW43: Essay schreiben (läuft schon)
  - KW44 - KW45: Funktionsrufe, Rekursion
  - KW46 - KW47: Datentypen
  - KW48 - KW49: Iterationen, Schleifen
  - KW50 - KW51: Ausdrücke, Logik
- 2018
  - KW01 - KW02: Modelle, Automaten, Syntaxdiagramm
  - KW03 - KW04: Komplexität

Online-Einreichung

- Einreichung von Lösungen für Trainings- und Prüfungsaufgaben online: (https://osg.informatik.tu-chemnitz.de/submit/)
- Funktioniert mit TU Chemnitz Login
- Automatische Validierung der Einreichung, Feedback per Mail

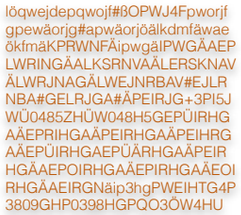


Programmierung

- Die meisten Trainings- und Prüfungsaufgaben drehen sich um Programmierung

Quelltext

Übersetzung



Ausführbares Programm

Editor? Sprache? Werkzeug? Ausprobieren?

Editor

- Programmieren braucht einen Editor
- Quasi-religiöser Streit bei Informatikern (wie beim Browser und Betriebssystem)

- IDE (Integrated Development Environment)
  - Bsp.: XCode, VisualStudio, Eclipse
  - Alles integriert, kompilieren und ausführen auf Knopfdruck
  - Zu viel ‚Magie' für diese Veranstaltung, Makefile's nur umständlich nutzbar
- Text-Editor
  - Bsp.: Sublime, Atom, Notepad++, TextEdit, vi, vim, emacs, Kate, gedit, …
  - Compiler muss von Hand aufgerufen werden
  - Dringende Empfehlung für diese Veranstaltung
    (sie machen ihren Flugschein auch nicht im F35 Kampfjet…)

Programmiersprache / Übersetzung

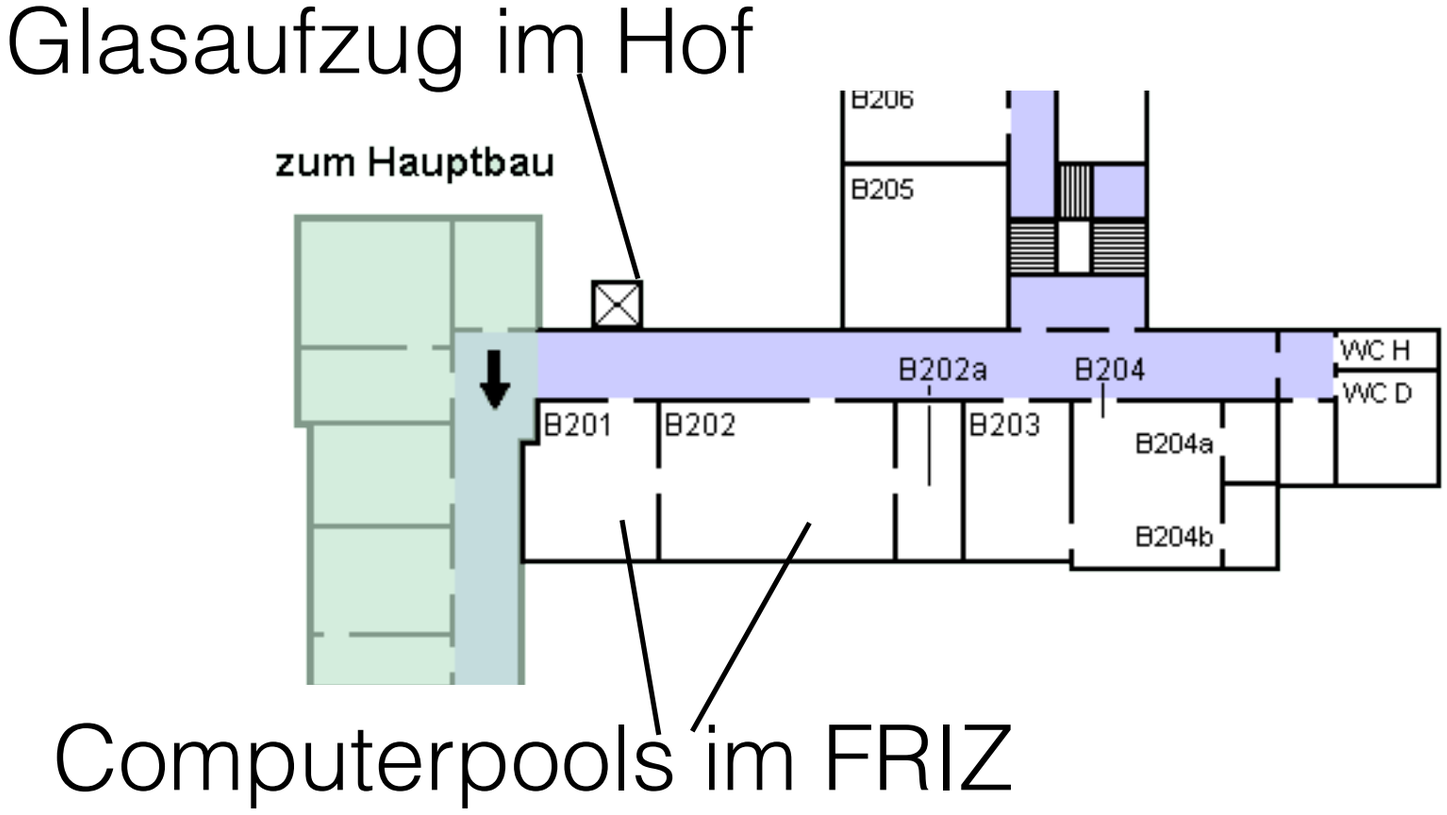
- Programmiersprache C
  - Wird im Informatik-Studium ständig benötigt
  - Grundlage für alle bekannten Betriebssysteme und eingebetteten Systeme
  - Kompilierte / übersetzte Sprache
    - Quelltext muss in Binärdatei umgewandelt werden
    - Nötig nach jeder Änderung
- Programmiersprache Python
  - Leichter zu lernen, schnelle Erfolgserlebnisse
  - Weit verbreitet in der modernen Webentwicklung
  - Interpretierte Sprache
    - Interpreter-Programm führt den Quelltext direkt aus
    - Keine Übersetzung (durch den Entwickler) nötig

Ausprobieren

- Sie brauchen einen geeigneten Computer zum Entwickeln / Testen
- Betrieb von Computerpools in der Universität
  - Pools des Universitätsrechenzentrums (URZ)
    - Nutzername/Passwort via Datenkontrollblatt (https://idm.hrz.tu-chemnitz.de/apps/register/student/)
    - Nutzerservice in Raum 1/072
  - Pools des Fakultätsrechen- und Informationszentrums (FRIZ)
    - Benutzername wie beim URZ, aber trotzdem eigener Account
    - Passwort via Operator in Raum 1/B201

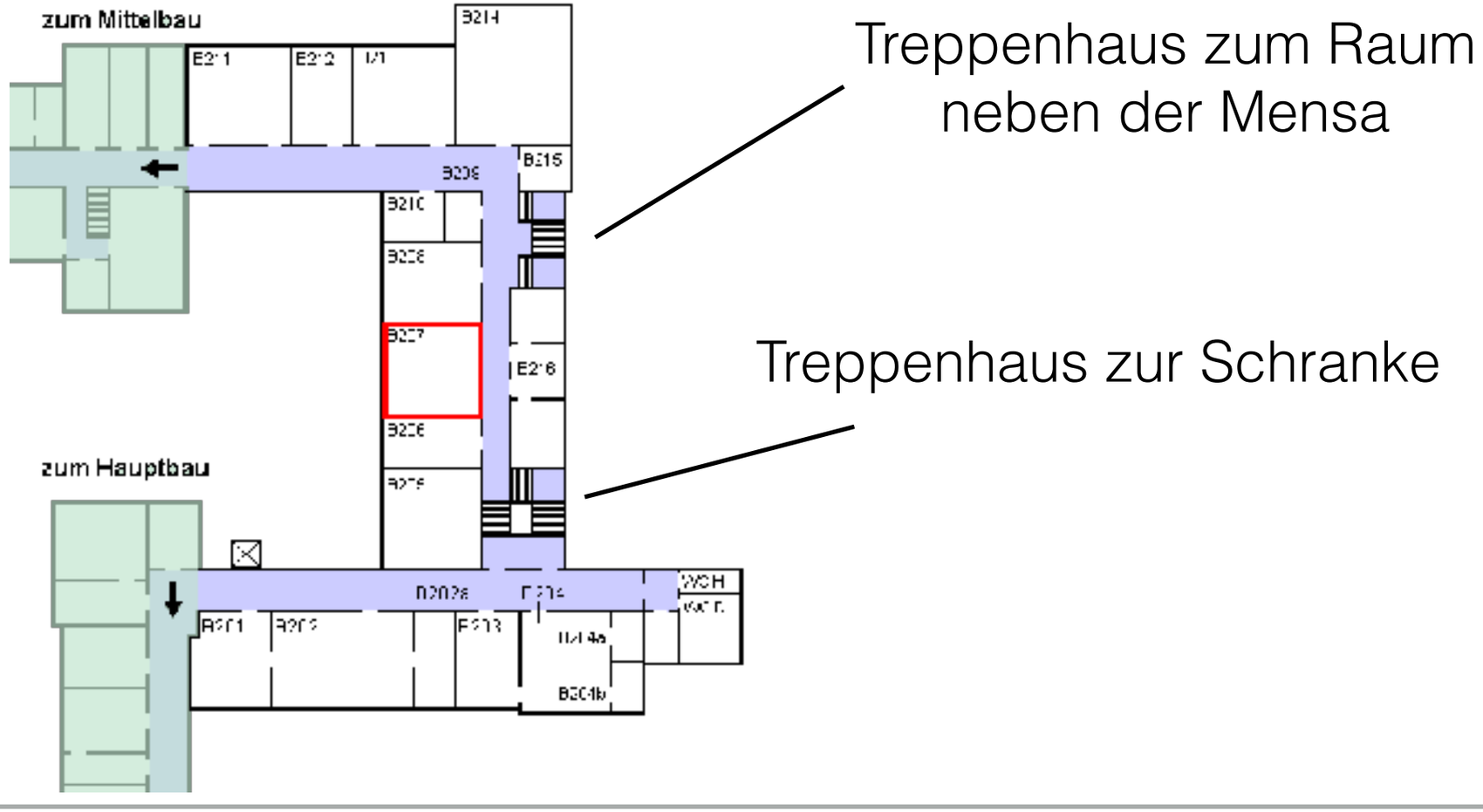
FRIZ Pools

- Zwei FRIZ Pools in diesem Teil der Universität (1/B201 und 1/B202)
- Werden teilweise für Veranstaltungen genutzt, Plan an der Tür
- Zugang mit Studentenausweis
- Alle Rechner mit openSUSE Linux + Windows 7



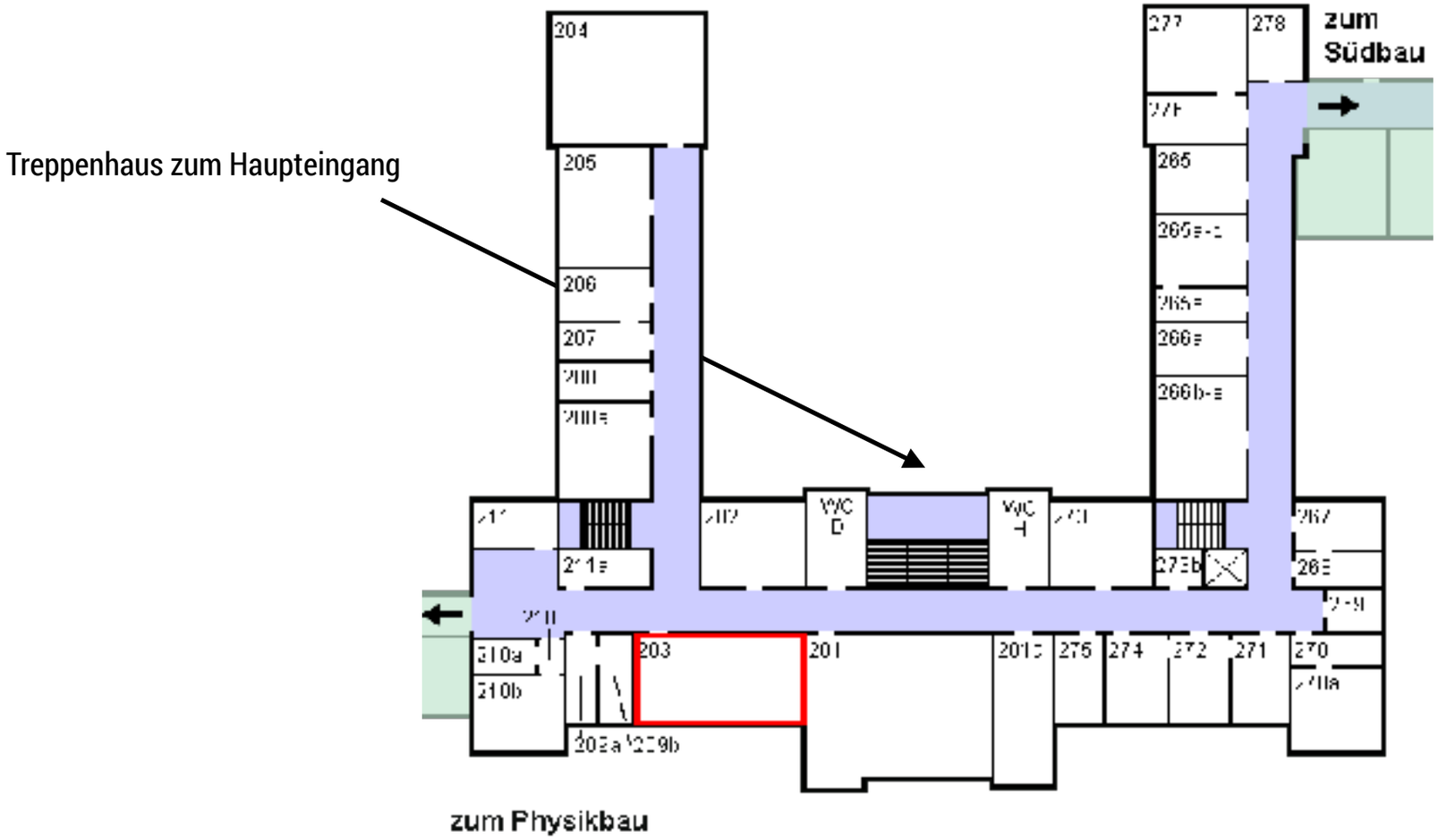
URZ Pools

- Drei URZ Pools in diesem Teil der Universität (1/B207, 1/203, 1/066)
- Werden teilweise für Veranstaltungen genutzt, Plan an der Tür
- Zugang mit Studentenausweis
- Alle Rechner mit Linux + Windows 7 (als virtuelle Machine)



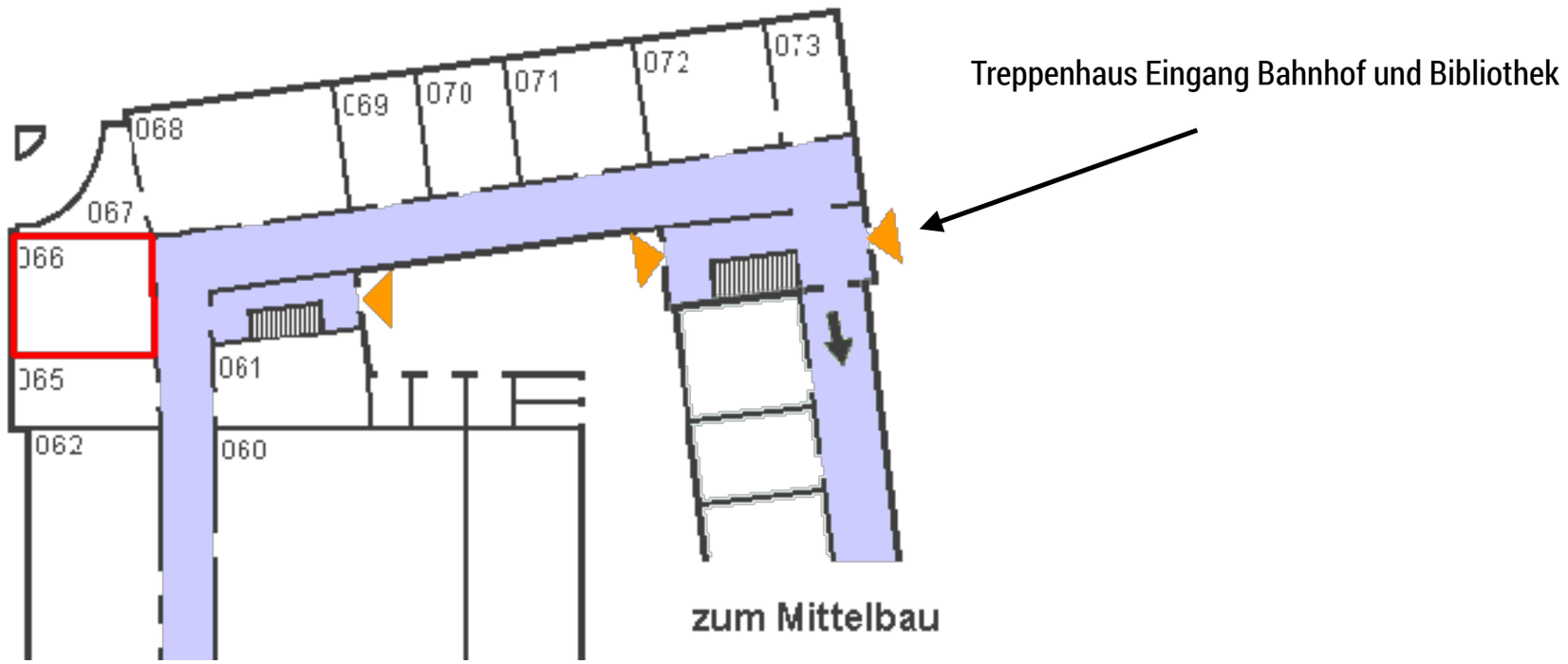
URZ Pools

- Drei URZ Pools in diesem Teil der Universität (1/B207, 1/203, 1/066)
- Werden teilweise für Veranstaltungen genutzt, Plan an der Tür
- Zugang mit Studentenausweis
- Alle Rechner mit Linux + Windows 7 (als virtuelle Machine)



URZ Pools

- Drei URZ Pools in diesem Teil der Universität (1/B207, 1/203, 1/066)
- Werden teilweise für Veranstaltungen genutzt, Plan an der Tür
- Zugang mit Studentenausweis
- Alle Rechner mit Linux + Windows 7 (als virtuelle Machine)



URZ Pools

- Weitere Pools in anderen Teilen der Universität
- Raumplan über den Campusplaner (https://www.tu-chemnitz.de/tu/lageplan/campusfinder/)
- Reichenhainer Straße
  - 7 Pools
  - Betriebssystem Linux und Windows (teilweise nur virtuell)
- Wilhelm-Raabe-Straße
  - 1 Pool
  - Betriebssystem Linux und Windows (virtuell)
- Thüringer Weg
  - 1 Pool
  - Betriebssystem Linux und Windows (virtuell)

Eigener Rechner?

- Eigener Rechner für Programmieraufgaben ist ok
  - Benötigt Installation von Werkzeugen zur Softwareentwicklung
  - gcc: GNU C Compiler
  - make: Hilfe bei der Erstellung komplexer Softwareprojekte
- macOS
  - Installation von gcc und make als Teil von XCode (App Store)
  - Alternative: https://brew.sh
- Linux
  - Installation von gcc und make meist schon passiert (URZ / FRIZ Pool)
  - Installation neuer Software je nach Distribution (yum, apt-get, rpm, …)
- Windows
  - Prinzipiell machbar, aber Werkzeuge heißen und funktionieren anders
- Eingereichte Lösungen für Prüfungsaufgaben **müssen** unter Linux funktionieren!

Beispiel: Hello World (Trainingsaufgabe)

- Aufgabe: Entwicklen Sie ein Programm, dass ‚Hello World' ausgibt
  - Quelltext muss entwickelt werden
  - Zusätzlich ist Makefile gefordert
  - Lokal testen, bis es funktioniert
  - Online einreichen
- Schritt 1: Das C-Programm schreiben

- Ergebnis ist Textdatei ‚hello.c'
- Editor liefert Hervorhebung der Syntax und erste Prüfungen
- Kann man so nicht ausführen

Beispiel: Hello World (Trainingsaufgabe)

- Schritt 2: Das Makefile schreiben
  - Steuert die Übersetzung in ein ausführbares Programm
  - Ergebnis ist Textdatei ‚Makefile‘

Ziel der Übersetzung

Abhängig von

Kommando, wenn sich hello.c ändert

Makefile

hello.c | Makefile

```
1  hello: hello.c
2      gcc -o hello hello.c
3
```

Line 3, Column 5 Tab Size: 4 Makefile

Beispiel: Hello World (Trainingsaufgabe)

- Schritt 3: Lokales Testen
  - Übersetzung durch das Kommando `make` im Terminal auslösen
    - Sucht ein Makefile und generiert die eingetragenen Ziele
  - Man kann den Übersetzer (`gcc`) auch direkt rufen
  - Diverse nützliche Optionen für gcc (-Wall, -g, …)
  - Wenn alles richtig ist, entsteht eine ausführbare Binärdatei
    - Läuft nur auf dem Systemtyp, wo die Übersetzung stattfand
    - Ausführung direkt im Terminal (`./hello`)

- Schritt 4: Online einreichen
  - ZIP-Archiv von Quelltext und Makefile erzeugen
  - Am einfachsten im Terminal: `zip helloworld.zip hello.c Makefile`
  - Hochladen in OpenSubmit

Anmerkungen

- Vorlesung + Übung reichen nicht, um C ausreichend gut zu lernen
  - Online-Kurse im Internet
  - Bücher
  - YouTube - Tutorials
  - Sofort anfangen. Heute.
- Übungsleiter sind Menschen
  - ‚Die schweigende Wand' als übliches Problem
  - Vorbereitung hilft - Folien durchblättern, Skript lesen, Trainingsaufgaben
- Sie sind erwachsen
  - Das deutsche Universitätssystem geht davon aus, dass sie selbständig arbeiten.
  - Wenn Sie nichts tun, passiert auch nichts.
  - Wir helfen gern.

Nächste Schritte

- Vorlesung besuchen
- Für Übung in OPAL registrieren und diese besuchen

- Erste Trainingsaufgabe lösen
- Erste Prüfungsaufgabe lösen

- Account besorgen und Pool finden, oder eigenen Rechner präparieren
- C lernen

Links

- Verschiebungen / Ausfall von Veranstaltungen: https://osg.informatik.tu-chemnitz.de
- Räume und Termine für diesen Kurs: https://osg.informatik.tu-chemnitz.de/lehre/
- Folien / Skript zur Vorlesung: https://osg.informatik.tu-chemnitz.de/lehre/aup/#material
- Welche Übungsgruppen sind frei? https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/15259697155/CourseNode/96292953258489
- Trainingsaufgaben und Prüfungsaufgaben einreichen: https://osg.informatik.tu-chemnitz.de/submit/dashboard/