

## Übung zur Vorlesung Grundlagen Betriebssysteme und Rechnernetze

2. Übungsblatt, Abgabe bis Mo, 17. November 2025, 10:00 Uhr

### **Hinweise zur Bearbeitung und Abgabe der Übung:**

- Die Übungsaufgaben und Zusatzmaterial finden Sie auf Moodle:  
<https://moodle2.uni-potsdam.de/course/view.php?id=47479>
- Abgabe in 3er-Gruppen; tragen Sie sich im Moodle-Kurs in eine Abgabegruppe ein! Auf allen Abgaben Namen und Matrikelnummern der Gruppenmitglieder nicht vergessen!
- Laden Sie Ihre Lösungen der Theorieaufgaben als PDF in Moodle hoch.
- Nutzen Sie für die Praxisaufgaben Git.UP! Klonen Sie sich für die jeweilige Aufgabe die Vorlage aus der Gruppe **GBR Vorlagen**. Sollte es für die Aufgabe keine Vorlage geben, zB: Aufgabe 1.1, erstellen Sie ein eigenes Repository. Achten Sie darauf ihre Git-Projekte im GBR Namespace zu erstellen. Für weitere Details folgen sie dem Manual auf Moodle.
- Halten Sie sich an die in den praktischen Aufgaben gegebenen Dateinamen und Methodennamen!
- Bitte keine Umlaute und Leerzeichen in Verzeichnissen und Dateinamen in der Abgabe verwenden! Ersetzen Sie Umlaute in Dateinamen (ä wird zu ae etc.)!
- Bei Problemen mit den Praxisaufgaben oder den Abgaben in GitLab legen Sie bitte in Ihrem GitLab-Repository ein Issue an und markieren darin einen Lehrenden! Bei Fragen zu den Aufgaben benutzen Sie bitte das Diskussionsforum in Moodle!
- Bitte kommentieren Sie Ihre Programme sinnvoll! Verwenden Sie sprechende Bezeichner und behandeln Sie mögliche Fehler!
- Stellen Sie sicher, dass Ihre Programme auf `tiree.lab.cs.uni-potsdam.de` compilierbar und lauffähig sind!

**Gesamtpunktzahl des Aufgabenblattes: 20 Punkte.**

## Aufgabe 2.1: Prozesserzeugung

Gegeben ist die Vorlage `fork.c`, die ein einfaches Wortklassifikationsmodell benutzt, um englische Wörter als Tier oder Nicht-Tier zu klassifizieren. Um Inferenz auch bei komplexeren Modellen in Zukunft schneller zu machen, soll die Klassifikation parallel in mehreren Prozessen laufen.

Erweitern Sie das Programm `fork.c` so, dass es die folgenden Anforderungen erfüllt:

- Jedes Wort soll in einem eigenen Kindprozess klassifiziert werden.
- Es sollen maximal vier Kindprozesse gleichzeitig laufen.
- Jeder Kindprozess soll neben dem Klassifikationsergebnis auch seine eigene PID und die PID seines Vaterprozesses ausgeben.

(6 Punkte)

## Aufgabe 2.2: Scheduling

- a) In der Übung wurde die FiFo-Scheduling-Strategie in den **Sched-Sim** Simulator implementiert. Erweitern Sie `schedule.c` um die Implementierung der Round-Robin-, Shortest-Job-First- und Fixed-Priority-Scheduling-Strategie. Erweitern Sie falls nötig die Implementierung der Run-Queue in `queue.c` um die benötigten Funktionen. (5 Punkte)

- b) Erstellen Sie für die Algorithmen Round-Robin, Shortest-Job-First und Fixed-Priority je ein Gantt-Diagramm, wobei die Kosten für Scheduler- und Dispatcher vernachlässigt werden sollen. Übernehmen Sie die Laufzeit und Ankunftszeiten der Prozess aus der `main.c`.

*Hinweis: Verwenden Sie für ihre Zeichnung ein Quantum von 100ms.* (3 Punkte)

- c) Bewerten Sie die **Effizienz** der drei Scheduling-Strategien in Bezug auf die mittlere Wartezeit, die mittlere Verweildauer und die mittlere Antwortzeit. Beziehen Sie sich dabei auf die in Aufgabe b erstellten Ablaufdiagramme. (3 Punkte)

### Aufgabe 2.3: Windows Scheduler

Gegeben sei ein **Einprozessor-System** mit Windows 2000. Weiterhin existieren 3 Benutzerprozesse mit der **regulären BasePriority**, von denen zwei rechenbereit sind. Bearbeiten Sie die drei nachfolgenden Szenarien mit den dazugehörigen Teilaufgaben.

- a) Im ersten Szenario ist (Prozess 2) auf das Netzwerk **wartend blockiert**. Prozess 2 erhält zum Zeitpunkt  $t=0$  die CPU, wenn das Netzwerk bereit ist.
  - (i) Ermitteln Sie die Größe der Zeitscheibe von Prozess 2.
  - (ii) Ermitteln Sie die Anzahl an Sekunden, bis Prozess 2 seine **Basispriorität** wieder erreicht hat.
  - (iii) Zeichnen Sie ein modifiziertes Gantt-Diagramm, aus dem die aktuellen Prioritäten der Prozesse ablesbar sind. Das Diagramm muss für jeden Prozess **mindestens 4 Zeitscheiben** umfassen.
- b) Im zweiten Szenario ist Prozess 2 ein **GUI-Thread**. Bearbeiten Sie die Teilaufgaben **a)(i)-a)(iii)** **erneut** für diesen Fall.

*Hinweis: Benutzen Sie die Boost- und Zeitscheibenwerte aus der Vorlesung!*

(1,5 + 1,5 Punkte)