

## Praktische Probeklausur „Betriebssysteme und Systemprogrammierung“ im Wintersemester 2020/2021

### *Wichtige Hinweise:*

*Achten Sie stets darauf, Fehler abzufangen und alle belegten Ressourcen wieder freizugeben, auch beim Abbruch durch einen Fehler. Es sind keine Hilfsmittel erlaubt, außer den im Home-Verzeichnis bereitgestellten Materialien. Die Rechner in diesem Raum sind komplett abgeschottet (kein Internet und kein USB). Die vorgegebenen Dateien sind schreibgeschützt und müssen umkopiert werden. Verwenden Sie hierbei die geforderten Dateinamen pro Aufgabe.*

### **Aufgabe 1: Informationen aus /proc auslesen**

Schreiben Sie ein C-Programm, welches in Standard-Output ausgibt wie viel Speicher verfügbar ist, um eine neue Anwendung ohne Swapping zu starten. Sie finden diese Information in der Datei `/proc/meminfo` unter `MemAvailable`.

Beachten Sie, dass die Kapazitäts-Angabe in MB erfolgen und die Ausgabe folgende Form haben soll:

```
$/aufgabe1
```

```
Verfügbarer Speicher: 2138 MB
```

Der Name für die abzugebende Quelltextdatei lautet: `aufgabe1.c`

(10 min.)

### **Aufgabe 2: Ringpuffer-Synchronisierung**

In dieser Aufgabe sollen zwei Threads (ein Producer und ein Consumer) erzeugt werden, welche auf einen globalen Ringpuffer zugreifen. In der Vorgabe `vorgabe.c` finden Sie dafür das globale `char`-Array mit dem Namen `buffer` und der festen Größe 4. Der Producer-Thread soll 10 Zeichen (ab 'A' aufsteigend im Alphabet) in das Array schreiben. Wenn der Ringpuffer voll ist, so muss der Producer warten bis der Consumer Zeichen herausgenommen hat. Der Consumer muss wiederum warten, wenn keine Zeichen im Puffer sind. Die Signalisierung zwischen Producer und Consumer muss mit Semaphoren realisiert werden.

### *Hinweise:*

*Es sind keine weiteren globalen Variablen erlaubt, nur diejenigen die in der Vorgabe bereits definiert sind!*

Der Name für die abzugebende Quelltextdatei lautet: `aufgabe2.c`

(35 min.)

Weitere Hinweise auf der nächsten Seite ...

Beachten Sie, dass die Ausgabe folgende Form haben soll:

```
student@ubuntu: $ ./aufgabe2
```

```
Producing A ...  
Producing B ...  
Producing C ...  
Producing D ...  
... Consuming A  
... Consuming B  
... Consuming C  
... Consuming D  
Producing E ...  
Producing F ...  
Producing G ...  
Producing H ...  
... Consuming E  
... Consuming F  
... Consuming G  
... Consuming H  
Producing I ...  
Producing J ...  
... Consuming I  
... Consuming J
```