

**Betriebssysteme WiSe 2016/17· Übungsblatt 1**  
Bearbeitungszeit: 27.10.2016 – 10.11.2015 um 09:59 Uhr

Bereiten Sie Ihre **Lösungen** grundsätzlich so vor, dass Sie diese in der Übung Ihren Kommilitonen in geeigneter Form **zeigen** und **diskutieren** können. Geben Sie bitte stets Ihre verwendeten **Quellen** an.

Die Abgabe erfolgt elektronisch über das KVV.

**Aufgabe 1-1 (Architekturen) – 5 Punkte**

Vorhandene Betriebssystementwürfen können grob in zwei Klassen eingeteilt werden:

- die Makrokernarchitektur,
- die Mikrokernarchitektur.

Diskutieren Sie beide Architekturansätze unter Zuhilfenahme mindestens folgender Quellen:

- J. Liedtke, Toward real  $\mu$ -kernels, Communications of the ACM, 39(9):70--77, September 1996,  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.4950&rep=rep1&type=pdf>
- C. Maeda, B.N. Bershad, Networking performance for microkernels, Proceedings of Third Workshop on Workstation Operating Systems, 13:154 – 159, April 1992  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.56.9733&rep=rep1&type=pdf>

**Aufgabe 1-2 (Steuerung von Geräten) – 15 Punkte**

Die Zielplattform wäre über die serielle Schnittstelle mit dem Arbeitsplatzrechner verbunden. Durch geeignete Programme (bspw. `veryminicom`) würde zwischen Board und Rechner eine Modemverbindung hergestellt, so dass Zeichen empfangen und gesendet werden können.

Entwickeln Sie einen Treiber für die serielle Schnittstelle, der es erlaubt, einzelne Zeichen über die Schnittstelle zu senden und zu empfangen. Um den Treiber im späteren Verlauf gut zum Debuggen benutzen zu können, programmieren Sie zudem ein einfaches `printf()`-Äquivalent. Dieses `printf()` soll lediglich eine Teilmenge an Umwandlungen verstehen, ohne irgendwelche Extras wie Flags oder Feldbreiten. Die Funktionalität muss umfassen:

- `%c` – `int` Argument wird nach `unsigned char` umgewandelt und als einzelnes Zeichen ausgegeben
- `%s` – durch `const char *` Argument referenzierte null-terminierte Zeichenkette wird ausgegeben
- `%x` – `unsigned int` Argument wird in hexadezimaler Darstellung ausgegeben
- `%p` – `void *` Argument wird in hexadezimaler Darstellung ausgegeben

Zu Demonstrationszwecken soll die Abgabe jeden Tastendruck mit einer schön formatierten Ausgabe quittieren.

**Hinweise:** Die in unserem Fall verwendete serielle Schnittstelle wird im Handbuch als Debug-Unit (DBGU) bezeichnet. Sie wird über QEMU direkt abgebildet. Dokumentation zu variablen Argument-Listen gibt es bspw. via `man stdarg`. Außerdem wird es in der Vorlesung weitere Hinweise geben.