



2. Übung zur Vorlesung Systemprogrammierung I

Abgabe per Email bis zum 27.05.05 vor den Übungen an
sp05@vs.informatik.uni-ulm.de

Aufgabe 1: Interrupts (2+3+2)

In den Vorgaben ist eine Methode InsertHandler für das Einfügen von Interrupts vorhanden.

- Erläutern Sie, wo der Interrupt-Handler eingefügt wird.
- Wie sieht in Plurix der Ablauf eines Interrupts vom Controller bis zum Aufruf des jeweiligen Handlers aus?
- Wie wird in Plurix die Nummer des aktuellen Interrupts / Exception ermittelt?

Aufgabe 2: Tastatur (2+3+2)

- Lesen Sie die Scancodes der Tastatur aus und geben Sie diese auf dem Bildschirm aus. Verwenden Sie für die Abfrage der Tastatur eine einfache Schleife.
- Erweitern Sie Ihren Treiber und berücksichtigen Sie zusätzlich die Tasten „Shift“, „Alt“ und „Strg“ und geben Sie diese zusammen mit den Scancodes aus (siehe Beispiel).
- Ändern Sie Ihren Treiber, so daß er interrupt-basiert arbeitet.

Beispielausgabe:

```
Scancode: 2 // 1 gedrückt
Scancode: 130 // 1 losgelassen
Scancode: 8 Modifier: shift // shift + 7 gedrückt
Scancode: 136 Modifier: shift // shift gedrückt; 7 losgelassen
Scancode: 5 Modifier: shift, strg // shift+Strg+4 gedrückt
Scancode: 5 Modifier: shift // strg losgelassen
Scancode: 5 // shift losgelassen
Scancode: 133 // 4 losgelassen
```

Aufgabe 3: Tastatur-Ringpuffer (3)

Entwickeln Sie nun einen verbesserten Tastaturhandler, der über einen 128 Bytes großen Ringpuffer für Scancodes verfügt. Bei einem Pufferüberlauf soll eine Fehlermeldung ausgegeben werden. Ferner soll Ihr Tastatur-Treiber folgende Methoden anbieten:

```
int ReadKey()
boolean KeyAvailable()
```

ReadKey: liest jeweils einen Scancode aus dem Tastaturpuffer aus und gibt diesen zurück. Ist kein Scancode im Puffer vorhanden, so wartet ReadKey solange, bis sich ein Scancode im Puffer befindet.

Key Available: liefert true zurück, wenn sich mindestens ein Scancode im Tastaturpuffer befindet, ansonsten wird false zurückgegeben.