

Aufgabe 1: Interrupts

Beantworten Sie folgende Fragen, halten Sie sich dabei weitestgehend an die in der Vorlesung benutzte Terminologie.

1. Erklären Sie den Unterschied zwischen Interrupts und Polling.
2. Welche Varianten beim Interruptpolling gibt es?
3. Erklären Sie *kurz* „IVT“ und „ISR“:
 - Was sind „IVT“ und „ISR“?
 - Was steht jeweils drin?
 - Wer legt sie an?
4. Kann eine Interruptbehandlung von einem neuen Interrupt unterbrochen werden?
5. Zeichnen Sie eine Daisychain mit 5 I/O-Geräten.
6. Warum sind Daisy-Chains sinnvoll?
7. Beschreiben Sie den Ablauf einer Interrupt-Behandlung nach dem Daisy-Chain-Verfahren.
8. Sind Daisy-Chains fair?

Aufgabe 2: Syscalls

Machen Sie sich mit Syscall Aufrufen auf Assembler Ebene vertraut. Schreiben Sie dann ein einfaches `cat` Programm, dass mindestens 100 Zeichen von der Konsole einlesen kann und diese direkt wieder auf der Konsole ausgibt. Nutzen Sie hierzu keinen Wrapper, sondern schreiben direkt Ihre `_start`-Funktion mit `-nostdlib`. Da Sie dann zum Beenden des Programmes explizit den Syscall `sys_exit()` aufrufen müssen, reicht kein einfaches `RET`. Eine Syscall-Übersicht finden sie unter https://blog.rchapman.org/posts/Linux_System_Call_Table_for_x86_64/

cat Unser `cat`-Programm liest ausschließlich von der Standardeingabe `stdin` und gibt den eingelesenen Buffer gleich wieder zur Standardausgabe `stdout` aus. War die Eingabe „leer“ (es wurden also keine Zeichen eingelesen) terminiert das Programm. Andernfalls wird nach der Ausgabe wieder eingelesen in einer Schleife.

Anmerkung: Der Syscall `write` garantiert nicht, dass er den gesamten Buffer ausgeschrieben hat, sondern muss ggf. selber in einer Schleife aufgerufen werden.