

Bitte beachten Sie die allgemeinen Hinweise auf Übungszettel 1

Aufgabe 1: Interrupts

Begriffe

Beschreiben Sie kurz **in eigenen Worten** den Unterschied zwischen Interrupts und Polling und legen Sie dabei auch die jeweiligen Vor- und Nachteile dar.

Abarbeitung von Interrupts

Die folgende Tabelle definiert eine Liste von Interrupts, die zu bestimmten Zeitpunkten auftreten und eine gegebene Zeit brauchen, um abgearbeitet zu werden:

Name	Priorität (höherer Wert := höhere Priorität)	Abarbeitungszeit (in Zeitslots)	Ankunftszeit (Zeitslot)
I0	1	4	1
I1	1	1	2
I2	2	2	2
I3	3	3	3
I4	2	3	5
I5	3	2	6
I6	1	1	7

Stellen Sie in jeweils einer Tabelle dar, in welchem Zeitslot welcher Interrupt aktiv ist und abgearbeitet wird.

Gehen Sie dabei für

- (a) von einem System mit **Sequential Interrupt Processing** und für
- (b) von einem System mit **Nested Interrupt Processing** aus.

Tragen Sie ein x ein, falls kein Interrupt aktiv ist.

Aufgabe 2: System Calls in C

In dieser Aufgabe sollen Sie nicht die `stdio.h` Bibliothek verwenden. Führen Sie Fehlerbehandlung durch und erläutern Sie im Quellcode (als Kommentar) warum Sie sich für genau diese Fehlerbehandlung entschieden haben.

1. Machen Sie sich mit den Linux Low-Level-IO system calls `open()`, `read()` und `write()` vertraut (Stichwort man pages). Suchen Sie weiterhin system calls zum Erstellen von Ordnern und Löschen von Dateien heraus.
2. Schreiben Sie eine Funktion mit der Signatur `int copy(char *sourcename, char *targetname)`, die den Inhalt einer Datei dann und nur dann in eine Zieldatei kopiert, wenn die Zieldatei noch nicht existiert.
3. Implementieren Sie unter Zuhilfenahme Ihrer `copy` Funktion einen einfachen kommandozeilenbasierten Papierkorb. Das Programm soll bei Ausführung dann und nur dann im aktuellen Verzeichnis den versteckten Unterordner `.ti3_trashcan` erstellen, wenn dieser Ordner noch nicht existiert. Unter Linux ist ein Ordner versteckt, wenn er mit einem Punkt beginnt. Folgende Funktionalitäten sollen implementiert werden:
 - (a) DELETE
Der Aufruf `./trashcan -d filename` verschiebt die Datei 'filename' in den versteckten Ordner.
 - (b) LIST
Der Aufruf `./trashcan -l` listet alle im versteckten Ordner enthaltenen Dateien auf.
 - (c) RECOVER
Der Aufruf `./trashcan -r filename` verschiebt die Datei 'filename' aus dem versteckten Ordner in das aktuelle Verzeichnis.
 - (d) FINALLY DELETE
Der Aufruf `./trashcan -f filename` löscht die Datei 'filename' aus dem versteckten Ordner.