

Aufgabe 13

Sie wollen einen **Link von** einer Datei, die auf einem (ext4-formatierten) **USB-Stick** gespeichert ist, auf ihren Linux-Desktop anlegen. Dazu verwenden Sie das Kommando

```
ln /Volumes/USB-STICK/start.html ~/Desktop/start.html
```

a) Warum scheitert dieses Kommando?

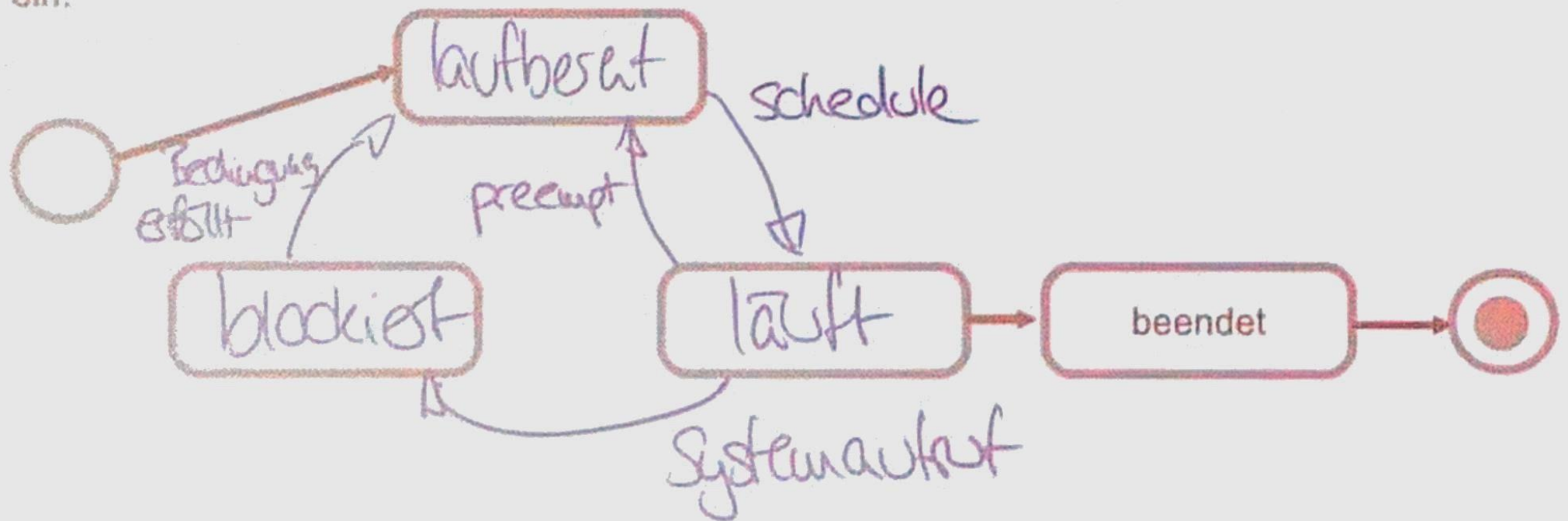
Hardlink kann nur innerhalb einer Partition angelegt werden, weil nur dann der Verweis auf den i-node eindeutig ist

b) Wie können Sie dieses Problem lösen?

Durch Anlegen eines symbolischen Links (mit ln -s). Der enthält nur den Pfad zur ursprünglichen Datei

Aufgabe 1

a) Beschriften Sie die drei grundlegenden Zustände eines Prozesses unter UNIX/Linux und zeichnen Sie die fehlenden möglichen Übergänge zwischen den Zuständen als Pfeile ein.



b) Es gibt verschiedene Gründe dafür, ein Programm mit mehreren nebenläufigen Threads zu schreiben. Nennen Sie zwei Gründe und begründen Sie in Stichworten, warum Multithreading in diesen Fällen sinnvoll ist. Wann kann eine Programmieraufgabe nicht sinnvoll in mehreren Threads gelöst werden?

1. Nutzung paralleler CPUs / CUP-Kerne, die nur so genutzt werden können

2. Bei Programmen mit GUI, weil dann eine flüssige Reaktion des Users auch dann gewährleistet ist, wenn die "Background-Threads" blockieren

Nicht sinnvoll, wenn Teilaufgaben stark voneinander abhängen

Aufgabe 3

a) Beschreiben Sie in Stichworten das **Round-Robin** Verfahren zur Verteilung der CPU-Zeit auf die Prozesse.

Die lauffähigen Prozesse erhalten der Reihe nach je eine Zeitscheibe CPU-Zeit zugeteilt

b) Nennen Sie zwei Möglichkeiten, wie man die **unterschiedlichen Prioritäten der Prozesse** in das Round-Robin Verfahren integrieren kann.

a) Variation der Länge einer Zeitscheibe

b) Häufigkeit, mit der eine Zeitscheibe zugeteilt wird

c) Das optimale Verfahren für das Prozeß-Scheduling ist beim **batch-processing** das **Shortest-Job-First** Verfahren. Trotzdem wird dieses Verfahren bei Betriebssystemen kaum verwendet. Warum?

Bei interaktiven Systemen wären alle Prozesse außer einem blockiert. Zudem lässt sich nicht sinnvoll entscheiden, welches Job der kürzeste ist

Aufgabe 6

a) Was unterscheidet eine **Speicherseite** (page) von einem **Seitenrahmen** (page-frame)?

- Seite \rightarrow virtuelles Speicher
- Seitenrahmen \rightarrow physikalisches Speicher

b) Nennen Sie zwei Gründe, warum eine Speicherseite **aktuell nicht angesprochen/ adressiert** werden kann.

- keine Zugriffsrechte
- Seite muss erst erstellt werden bzw. ist ausgelagert

c) Bei der IA32 (Intel x86) Architektur wird zur Umsetzung einer Adresse im virtuellen Speicher auf die zugehörige Adresse im physikalischen Speicher eine **zweistufige Seitentabelle** (page directory-table=top-level und page-table=second-level) verwendet. Die Speicherseiten sind je 4K-Byte groß. Wieviele Seitentabellen (top-level + second-level) kann es maximal pro Prozess geben, d.h. wieviel Kacheln können/müssen max. bei einem Prozess für dessen Seitenumsetzung genutzt werden? Erläutern Sie in Stichworten, wie Sie auf diese Zahl kommen.

