Seminarvortrag „Swapping“
Schwerpunkt Linux

Konzepte von Betriebssystem-Komponenten (KVBK)

Thomas Mayer

Gliederung

Was ist Swapping?

Aufbau eines Swapbereiches

Swapcache

Ablauf von Swapping Out und Swapping In

Speicherrückgewinnung

Was ist Swapping?

- Auslagerung von Daten aus dem Arbeitsspeicher auf die Festplatte

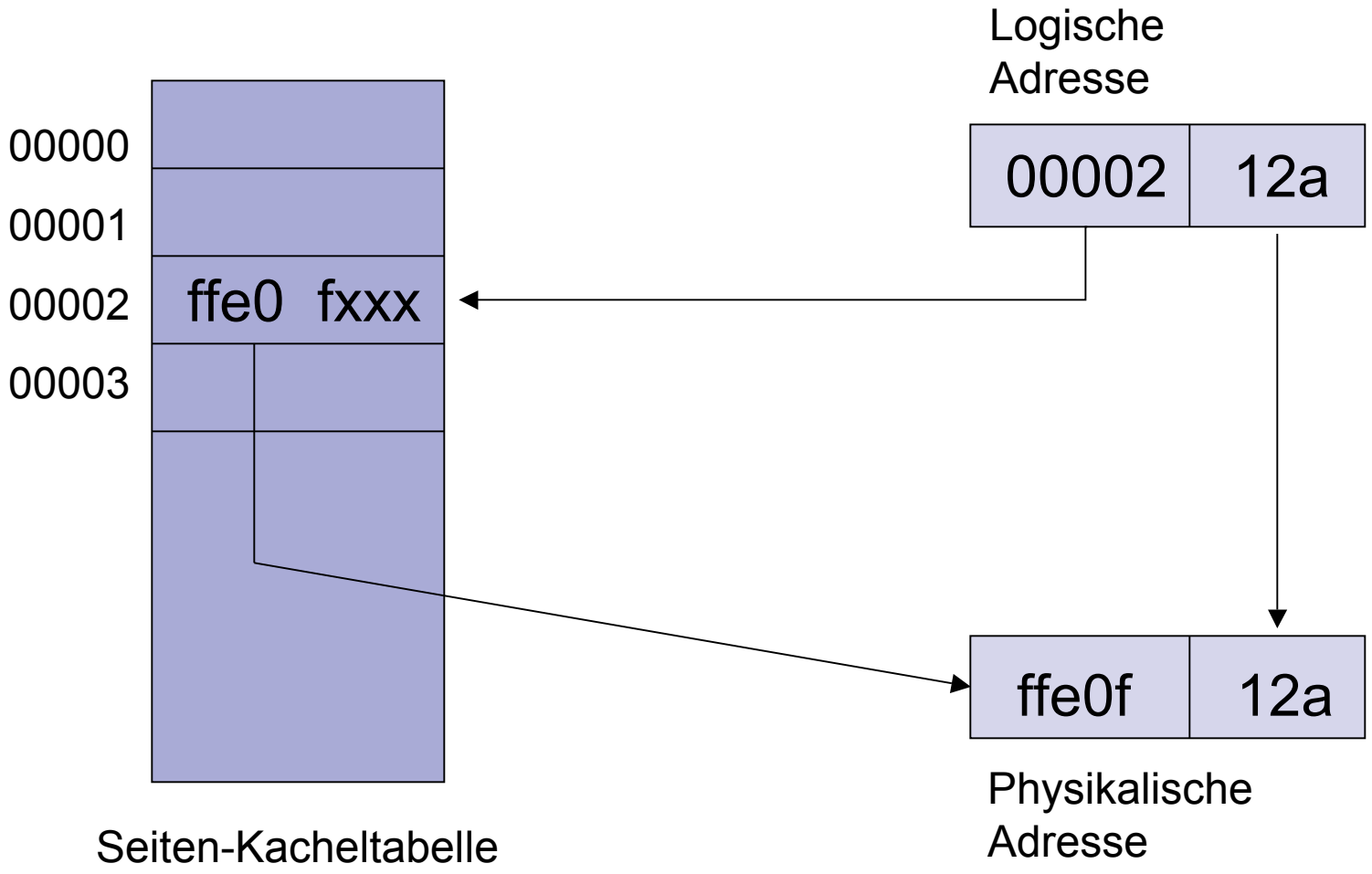
Vor- und Nachteile von Swapping

(+) Erweiterung des von Prozessen nutzbaren Adressraums

(+) Vergrößerung des nutzbaren Arbeitsspeichers

(-) langsamerer Zugriff als beim „echten Arbeitsspeicher“

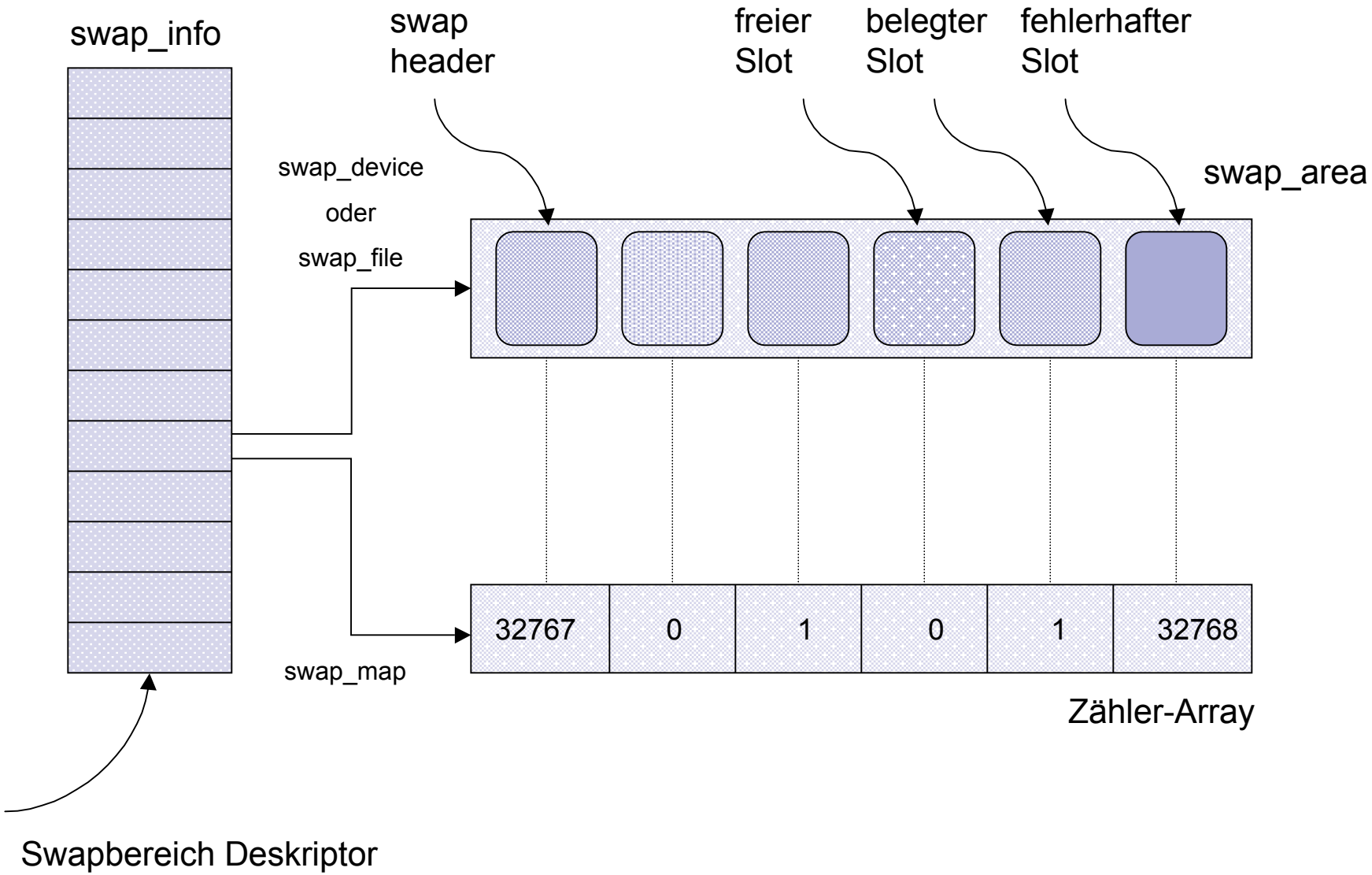
Rückblick: Paging



Der Swapbereich

- Aufbau aus Slots mit jeweils 4096 Bytes
- Speicherung genau einer Seite in jedem Slot
- Der erste Slot beinhaltet den Swap Header

Swapbereich



Aufbau des Swap Headers

- Aufteilung in **`magic`** und **`info`** Bereich

Der **`magic`** Bereich

beinhaltet Informationen über den Swap-Algorithmus

`SWAP-SPACE` entspricht Version 1

`SWAPSPACE2` entspricht Version 2

Swapbereich

Der **`info`** Bereich

`info.last_page`

Angabe des letzten benutzbaren Slots

`info.nr_badpages`

Anzahl der fehlerhaften Slots

`info.padding`

Padding Bytes (Auffüllende Bytes)

`info.badpages`

kennzeichnet Position der fehlerhaften Slots

Der Swapbereich Deskriptor

- jeder Swapbereich hat einen **`swap_info_struct`** Deskriptor

`flags`

zeigt an, ob der Swapbereich aktiv, bzw. beschreibbar ist

`swap_map`

deutet auf ein Zähler-Array, ein Zähler für jeden Slot

`prio`

Priorität, mit der auf den Swapbereich zugegriffen wird

Swapbereich

`sdev_lock`

Spinlock, schützt den Deskriptor von konkurrierenden Zugriffen in SMP Systemen

`max`

Maximale Anzahl der Seiten im Swapbereich

`pages`

Anzahl der verwendbaren Seiten

`next`

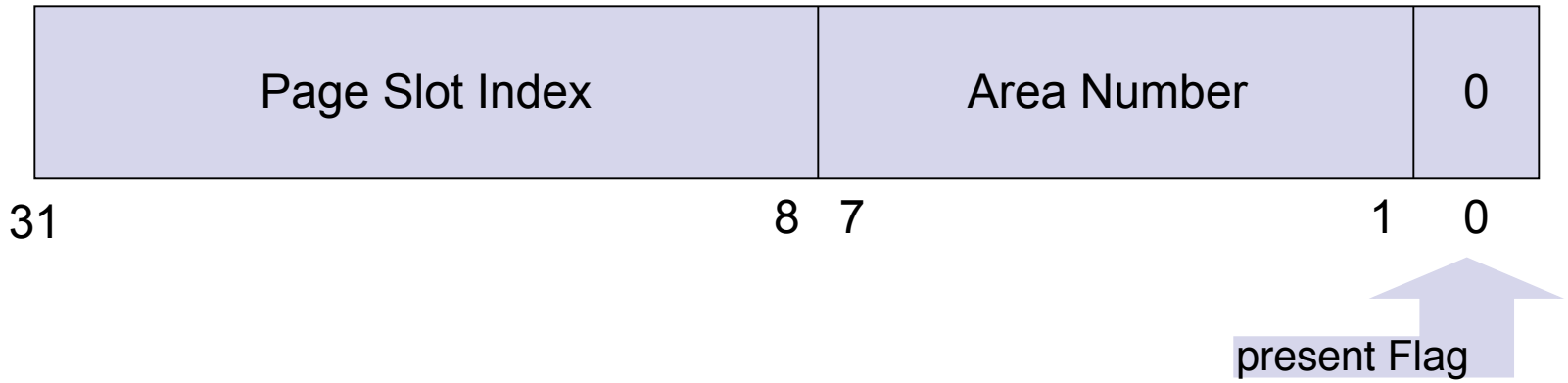
Zeiger auf den nächsten Swapbereich Deskriptor

Der Swap Out Page Identifier

- Indikator, ob sich eine Seite im Speicher oder im Swapbereich befindet

- hat das letzte Bit den Wert 0 wurde die Seite ausgelagert

- hat es den Wert 1 befindet sich die Seite im Arbeitsspeicher



Erstellen, Aktivieren und Deaktivieren von Swapbereichen

Erstellung und Aktivierung

- Definieren eines Swapbereiches mit mkswap
- Aktivierung mit Hilfe von **`swapon( )`**
- **`swapon( )`** ruft den Systembefehl **`sys_swapon( )`** auf
- Überprüfung der Berechtigung des Benutzers
- Berechnung der Prioriät
- Schreiben des Swap Headers
- Überprüfung der Slots, Kennzeichnung mit 0, bzw. 32768
- Eintrag des Deskriptors in die Swapliste

Swapbereich

Deaktivierung

- Deaktivierung durch **`swapoff( )`**

- Verschieben der ausgelagerten Daten in den Hauptspeicher

- Zugehörigkeiten werden mit **`try_to_unuse( )`** herausgefunden

- Seite wird gegen erneutes Auslagern gesperrt

- sequentielles Abarbeiten -> sehr zeitaufwendig

Anfordern und Freigeben von Slots

Beginnend beim Anfang des Swapbereichs

- Swapbereich immer voll ausgenutzt
- kaum leere Slots im belegten Swapbereich

Beginnend beim zuletzt angeforderten Slot

- schnellere Methode
- stärkere Fragmentierung
- Verwendung in Linux

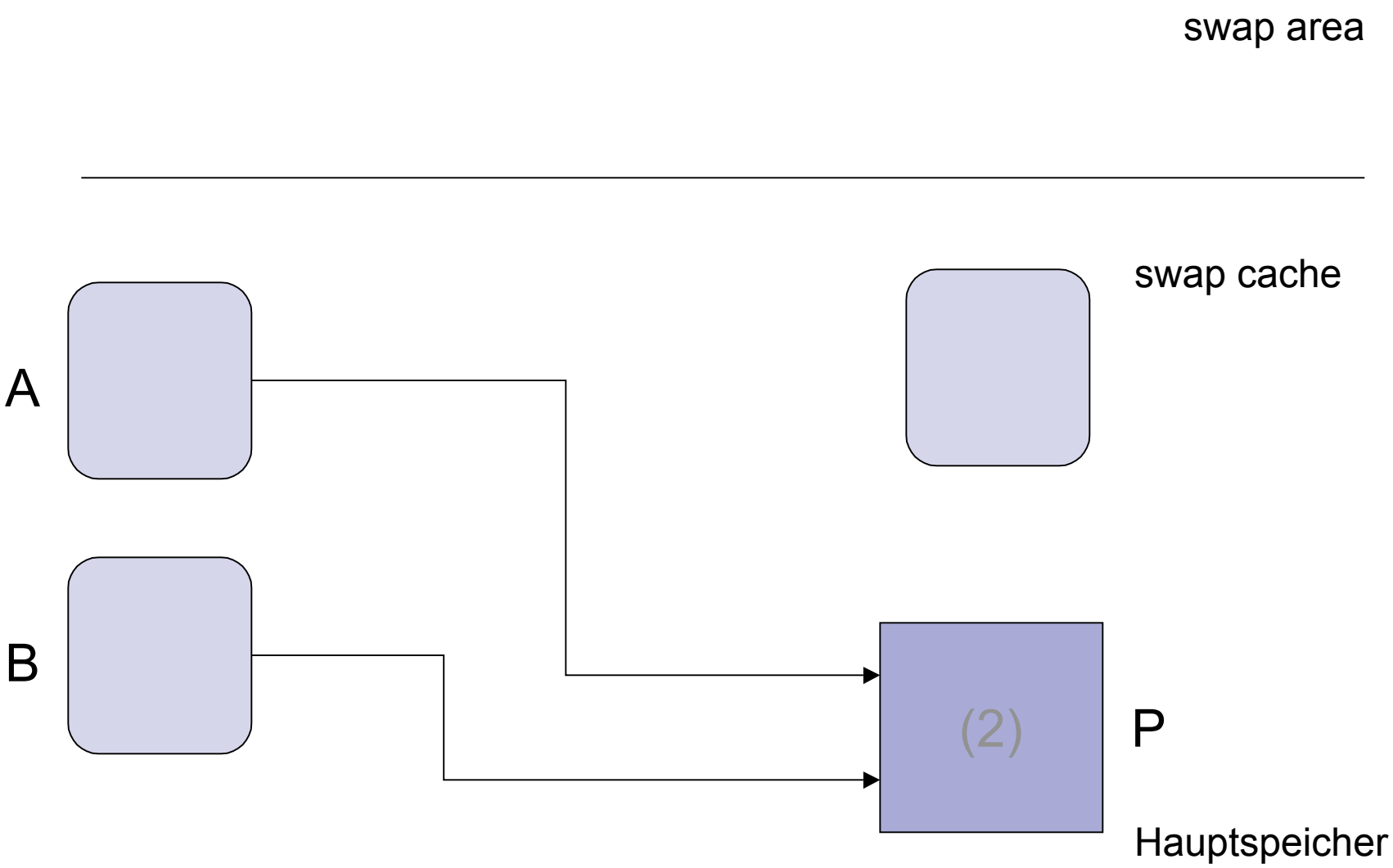
Der Swapcache

- mehrere Prozesse greifen auf die gleiche Seite zu:
  Probleme, wenn einer der Prozesse die Seite auslagern will

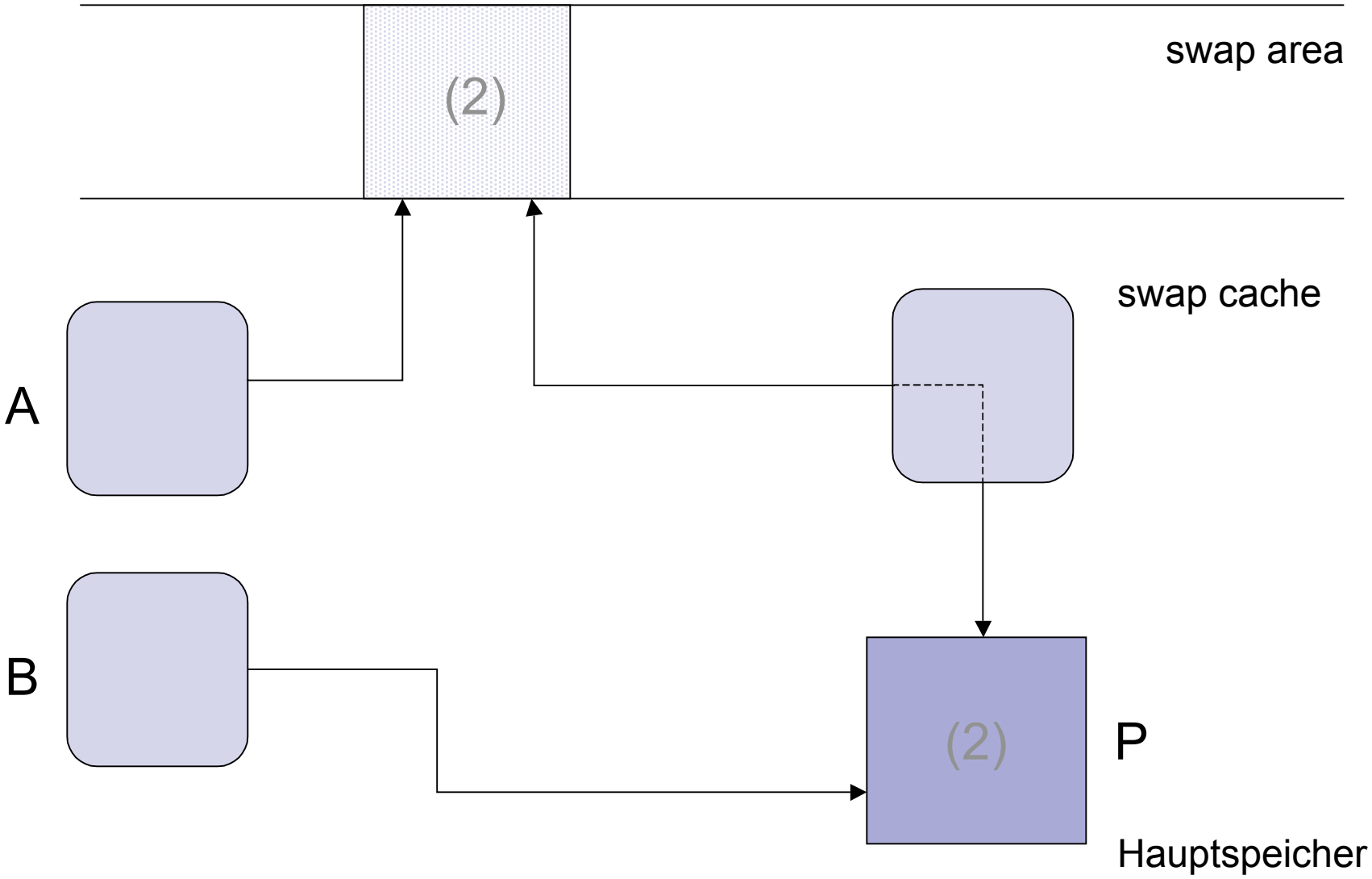
- denkbare Lösung:
  Seiten werden entweder für alle oder für keinen Prozess ausgelagert

- Realisierung in Linux:
  Einsatz eines Swapcaches, der sowohl auf die ausgelagerte, als auch auf die noch im Hauptspeicher verbleibende Seite verweist

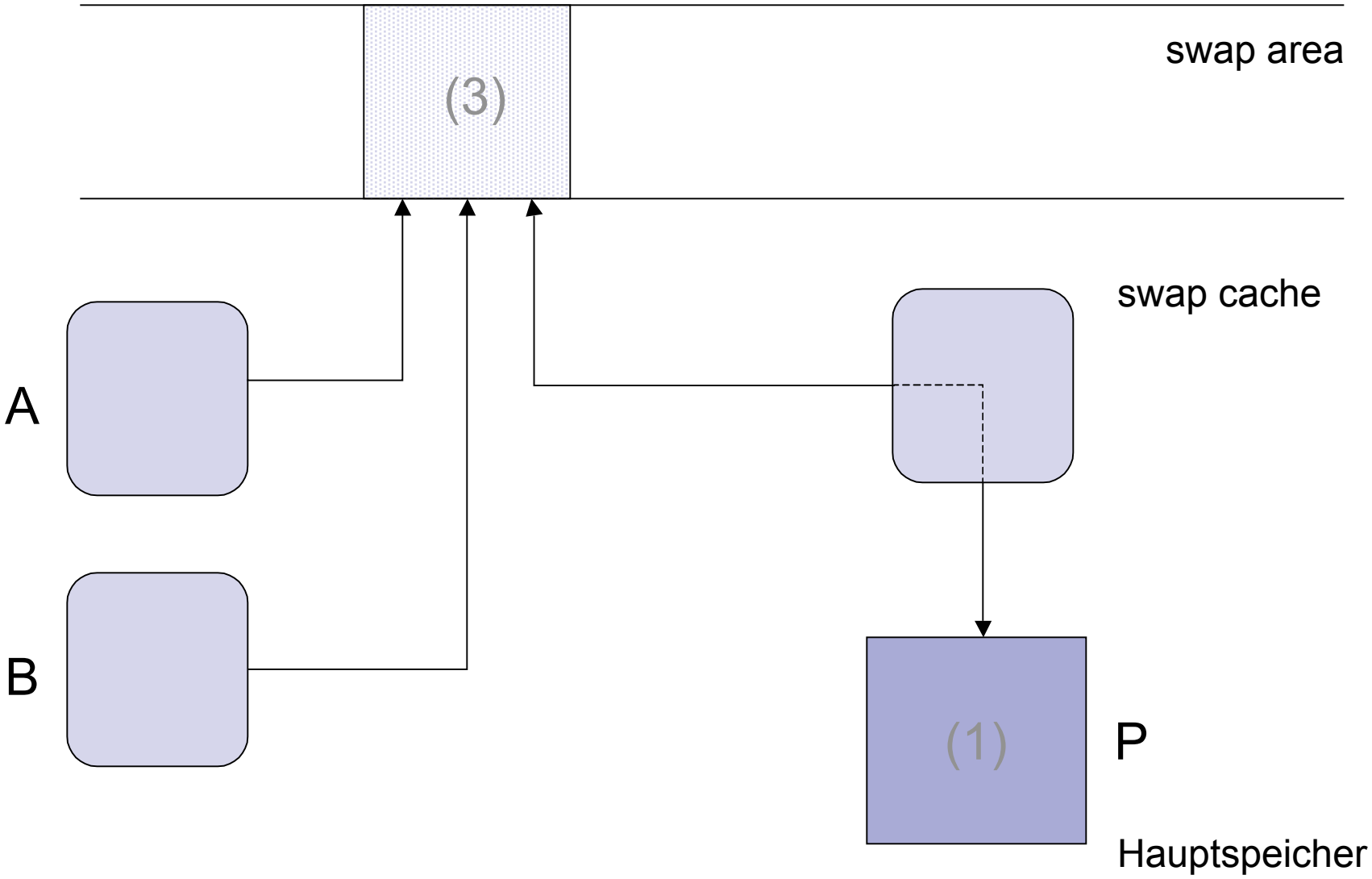
Swapcache



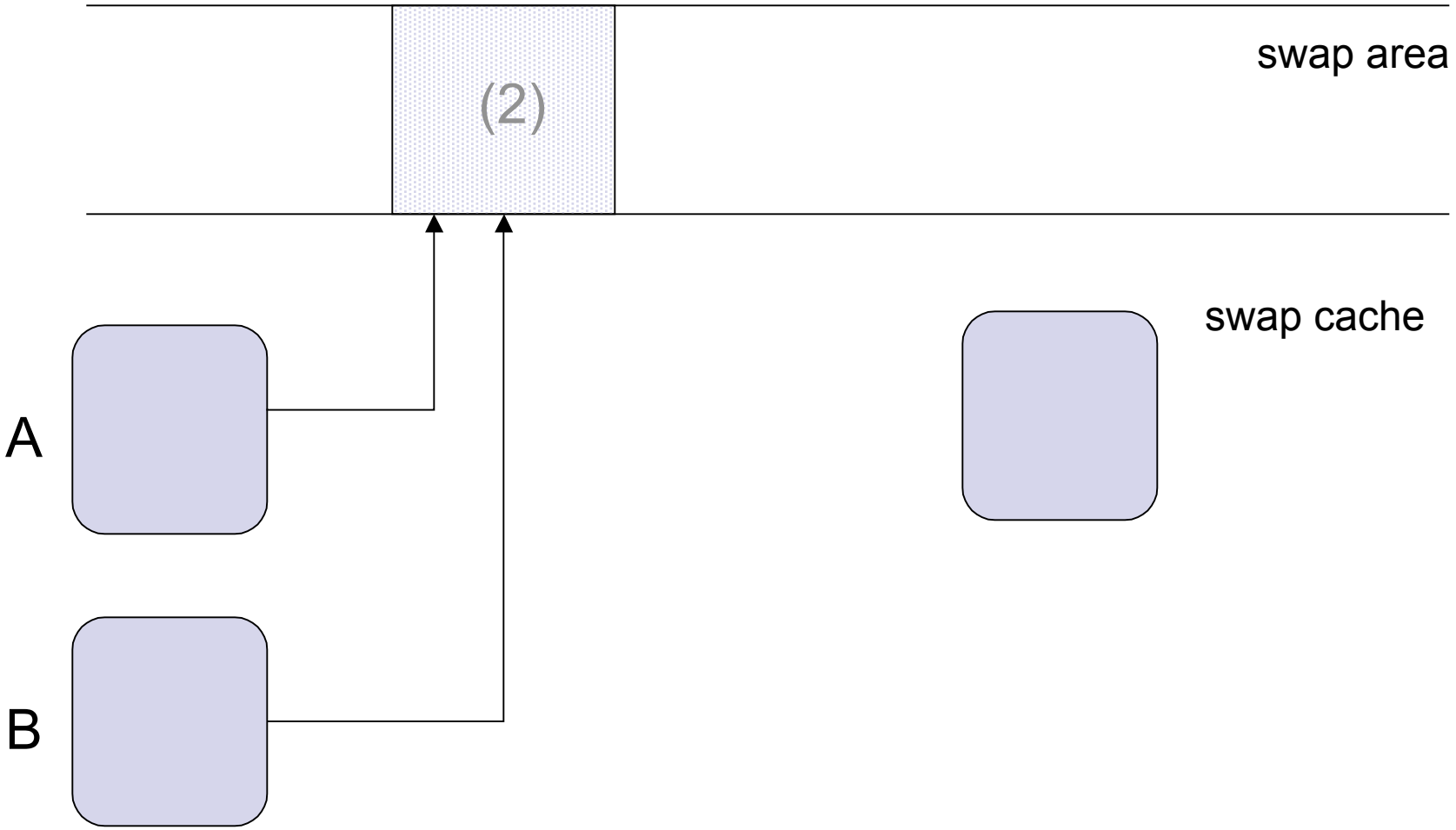
Swapcache



Swapcache



Swapcache



Helferfunktionen

`lookup_swap_cache( )`

findet die Seite im Swapbereich, auf die der Identifier passt

`add_to_swap_cache( )`

fügt eine Seite in den Swapcache ein

`delete_from_swap_cache( )`

löscht eine Seite aus dem Swapcache

`free_page_and_swap_cache( )`

entfernt unbenutzte Seiten aus dem Swapcache

Swapping Out

- Auslagerung nicht mehr benötigter Seiten aus dem Arbeitsspeicher in den Swapbereich
- Zuvor Suchen nach einem frei verfügbaren Slot

`try_to_swap( )`

Versucht im Arbeitsspeicher Platz zu schaffen

lokalisiert und markiert entsprechende Speicherbereiche mit

`mark_page_accessed( )`

- Überprüfung der Zugehörigkeit der Seite
- Sperren der Seite und Löschen deren Eintrag aus der Seitentabelle
- Schreiben in den Swapcache
- neuer Vermerk in der Seitentabelle

Swapping In

- Zurückschreiben einer ausgelagerten Seite in den Arbeitsspeicher
- erforderlich bei erneutem Zugriff auf die Seite

`do_swap_page( )`

ruft folgende Funktionen auf:

`swapin_readahead( )`

notwendig um aus dem Swapbereich zu lesen

`read_swap_cache_async( )`

notwendig um eine Seite einlagern zu können

`mark_page_accessed( )`

Seite wird gekennzeichnet und gesperrt

- Seitentabelle wird aktualisiert und Seite freigegeben

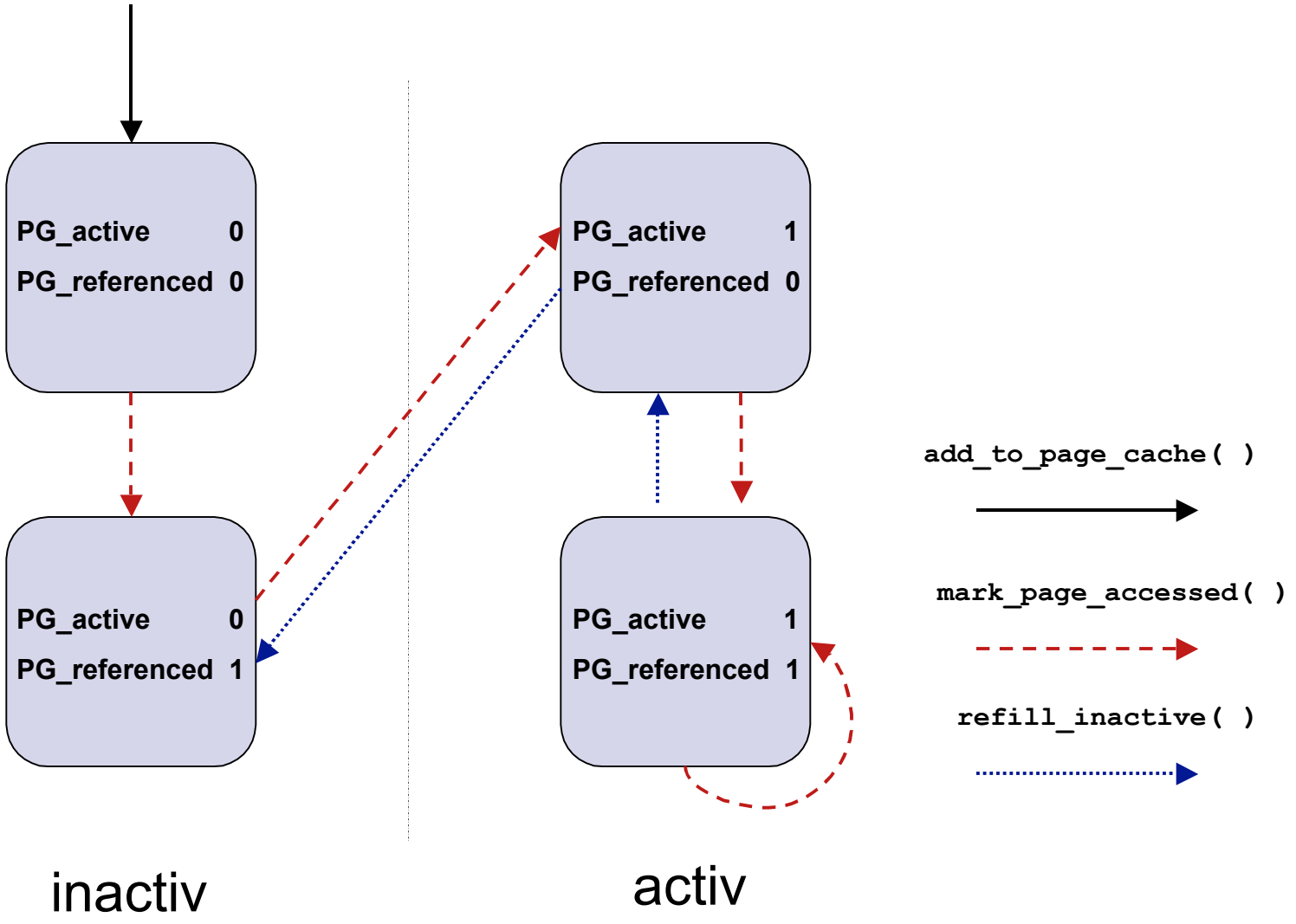
Speicherrückgewinnung

- Realisierung unter Linux:

  LRU („Least Recently Used“) – Strategie um zu entscheiden, welche Seiten im Cache bleiben und welche ausgelagert werden

- Verwendung einer LRU-Liste, die wiederum aus zwei Listen (**`activ`** und **`inactiv`**) besteht

Speicherrückgewinnung



Speicherrückgewinnung

- Eine Seite wird beim ersten Aufruf in die **`inactiv`** Liste geschrieben
- Die Seite wird beim ersten Aufruf als referenziert markiert
- Sie wird in die activ Liste übernommen (**`active`** Bit gesetzt, **`referenced`** Bit gelöscht)
- Beim erneuten Aufruf wird das **`referenced`** Bit wieder gesetzt
- Die Seite bleibt so lang in der **`activ`** Liste, wie auf sie zugegriffen wird (Zyklische Methode)
- Wird auf die Seite über längere Zeit nicht mehr verwiesen, wird sie wieder auf die **`inactiv`** Seite geschrieben

Speicherrückgewinnung

Die **`try_to_free_page( )`** Funktion

- Kernelfunktion um Seiten wieder freizugeben

- ruft die **`shrink_caches( )`** Funktion mit steigender Priorität auf, um eine angegebene Anzahl von Seiten freizugeben

- Scheitert die Funktion, wird ein Prozess im Usermode beendet

- **`shrink_cache( )`** realisiert das eigentliche Freigeben von Seiten, indem sie auf der inactiv Liste nach geeigneten Slots sucht

Swap-Datei unter Linux

Zum Schluss ein praktisches Beispiel:
Anlegen einer Swapdatei unter Linux

neuer Eintrag in der /etc/fstab :

```
/dev/vg00/lvol8    /users  hfs      rw,suid  0  2
/dev/vg00/lvol8    /users  swapfs min=3840, lim=12800, pri=2  0  2
```

- Auslagerung ins Verzeichnis /users auf der Platte /dev/vg00/lvol8
- Verwendung von mind. 3840 und max. 12800 Blöcken
- Priorität auf 2 (wird erste verwendet, wenn der Swapbereich mit pri=1 voll ist)

Swap-Datei unter Linux

Anlegen einer Swap-Datei:

```
dd if=/dev/zero of=swapfile bs=1024 count=65536
```

- Blockgröße beträgt 1024 Bytes
- Anzahl der benutzen Blocks
- Anschließend Aktivierung mit swapon

Welche Quellen habe ich verwendet?

Understanding the Linux Kernel (2nd Edition)
Daniel P. Bovet, Marco Cesati, 2002, Sebastopol

Galileo Computing - Swapping
(aus der Reihe „Wie werde ich ein Linux-Guru?“)
www.galileocomputing.de, Galileo Press GmbH, 2003, Bonn

Linux Paging, Caching und Swapping
Timo Schreiber, 2000, Uni Karlsruhe

Understanding The Linux Virtual Memory Manager
Mel Gorman, 2003, University of Limerick