

Systeme Wissensfragen

AlphaNerd

8. März 2016

Kapitel 2: Überblick

- 1) Verschiedene Arten von Betriebssystemen werden für verschiedene Verwendungszwecke benutzt. Nennen sie 4 Arten von Betriebssystemen.

1. Echtzeit-BS
2. Server-BS
3. Embedded-Systems-BS
4. PC-BS
5. Mainframe-BS

Kapitel 3: Dateisysteme

- 1) Nennen sie 5 Dateiattribute (Metadaten) die vom Betriebssystem angelegt werden.

Attribute	Bedeutung
Schutz	Wer kann auf die Datei zugreifen
Passwort	Passwort für den Zugriff auf die Datei
Urheber	ID der Person, die die Datei erzeugt hat
Eigentümer	Aktueller Eigentümer
Read-only-Flag	0: Lesen/Schreiben; 1: nur Lesen
Hidden-Flag	0: normal; 1: in Listen Sichtbar
System-Flag	0: normale Datei; 1: Systemdatei
Archiv-Flag	0: wurde gesichert; 1: muss noch gesichert werden
ASCII/Binär-Flag	0: ASCII-Datei; 1: Binärdatei
Random-Access-Flag	0: nur sequenzieller Zugriff; 1: wahlfreier Zugriff
Temporary-Flag	0: normal; 1: Datei bei Prozessende löschen
Sperr-Flags	0: nicht gesperrt; nicht null: gesperrt
Datensatzlänge	Anzahl der Bytes in einem Datensatz
Schlüsselposition	Offset des Schlüssels innerhalb des Datensatzes
Schlüssellänge	Anzahl der Bytes im Schlüsselfeld
Erstellungszeit	Datum und Zeitpunkt der Dateierstellung
Zeitpunkt des letzten Zugriffs	Datum und Zeitpunkt des letzten Zugriffs
Zeitpunkt der letzten Änderung	Datum und Zeitpunkt der letzten Änderung der Datei
Aktuelle Größe	Anzahl der Bytes in der Datei
Maximale Größe	Anzahl der Bytes für maximale Größe der Datei

- 2) Wie unterscheiden sich die rwx Rechte für Dateien und Verzeichnisse?

Read gleich. **Write** bei Verzeichnissen: neue Objekte anlegen. **Execute** bei Verzeichnissen: Betreten und Objekte ausführen. (Anmerkung: Es ist kein execute recht nötig um Objekte anzulegen in Verzeichnissen! D.h. ohne in das Verzeichnis wechseln zu dürfen, können Objekte angelegt werden.)

- 3) Welche Auswirkung hat das setzen des SUID- bzw. SGID-Flags auf die Dateirechteverteilung?

SUID: Ausführung (für alle User) mit Rechten des Besitzers.

SGID: Ausführung (für alle User) mit Rechten der besitzenden Gruppe.

- 4) Wie verhalten sich mit `ln -s QUELLE ZIEL` erstellte Dateien im Vergleich zu mit `ln QUELLE ZIEL` erstellte Dateien? Sind Verzeichnisse als `QUELLE` gültig?

`ln -s QUELLE ZIEL`: Symbolischer Link, als Zeiger, unterschiedliche Rechte als Quelle, bei Löschen von Quelle zeigt er ins leere, Partitionsunabhängig, eigene Inode, eigener Name.

`ln QUELLE ZIEL`: Hard Link, eigene Instanz des Dateiobjekts, bei Erstellen gleiche Rechte.

Nur bei Softlinks möglich, bei Hardlinks könnte es Kreisschlüsse geben.

- 5) Geben sie die minimale Zugriffszeit auf das n-te Byte bei a) Sequentiellen Speichertypen und b) Random-Access-Memory an.

a) $O(n)$, da über alle vorheriger Bits.

b) $O(1)$, da konstanter Zugriff.

- 6) Geben sie 6 Systemaufrufe (Operationen) auf a) Dateien und b) Verzeichnisse an.

a) Delete, Create, Read, Write, Open, Close, GetAttribute, SetAttribute, rename, append, seek, ...

b) Delete, Create, ReadDir, OpenDir, CloseDir, GetAttribute, SetAttribute, rename, ..

- 7) Welche Vor- und Nachteile besitzt die Realisierung von Dateien als zusammenhängende Belegung (2 Nennungen)? Wo wird diese eingesetzt?

Positiv: Lesegeschwindigkeit, keine Interne Fragmentierung, kein Speicheroverhead

Negativ: Dateigröße muss feststehen, hohe externe Fragmentierung, Verwaltungsaufwand durch Defragmentierung.

Verwendung: Medien die nur einmalig beschrieben werden, z.B. CD, Backup, ..

- 8) Welche Vor- und Nachteile besitzt die Realisierung von Dateien als verkettete Listen auf der Festplatte (3 Nennungen)? Wie verändern sich diese beim benutzen einer FAT?

Vorteile: Interne Fragmentierung nur bei letztem Glied, keine externe Fragmentierung, Dateien beliebiger größer anlegbar.

Nachteile: Langsamer zugriff, (Overhead durch Zeiger)

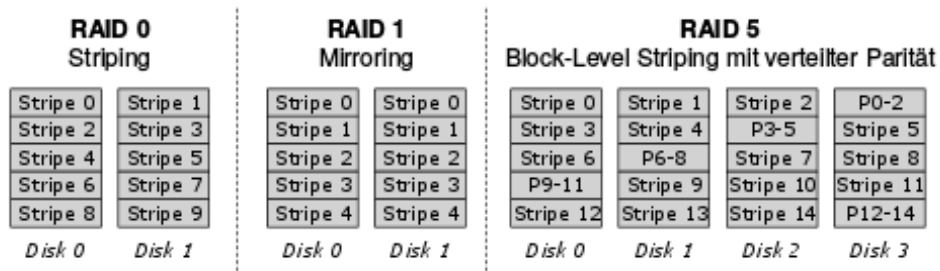
FAT: Schnellere Zugriffe, durch Fat im Hauptspeicher. Aber FAT belegt Speicherplatz im Hauptspeicher (unabhängig von aktueller Anzahl an benutzten Dateiblöcken).

- 9) Welchen Einfluss hat die Wahl der Blockgröße auf die Performance eines Fat32 Dateisystems?

Kleinere Blöcke bedeuten weniger interne Fragmentierung, jedoch ist die FAT größer (mehr Zeiger). Größere blöcke komplementär. Maximale Größe des Dateisystems wird durch Blockgröße begrenzt.

- 10) RAID 0: Striping
 RAID 1: Mirroring
 RAID 5: Block-Level Striping mit verteilter Parität.

Skizzieren sie die Realisierung von RAID 0/1/5.



Kapitel 4: Prozesse

- 1) Erklären sie den Unterschied zwischen Programm, Prozess und Threads. Welche Informationen werden jeweils verwaltet (jeweils 3 Nennungen).

Ein Programm ist ein implementierter Algorithmus, folgende Informationen werden verwaltet:

1. Programmdatei
2. Programmvorschrift
3. Metadaten

Ein Prozess ist eine Instanz eines Programmes, folgende Informationen werden verwaltet:

1. Programmcounter (PC)
2. Registerinhalte
3. Variablen Belegung

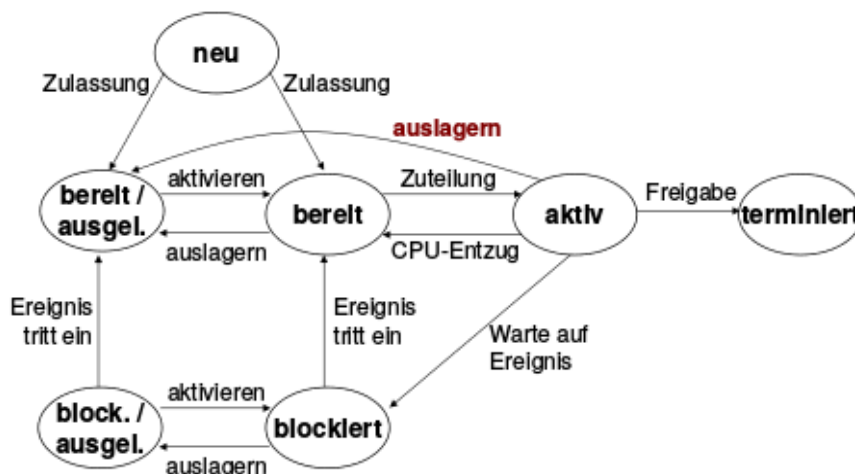
Ein Thread ist ein Unterprozess mit geteiltem Namensraum, folgende Informationen werden verwaltet:

1. Programmcounter (PC)*
2. Registerinhalte
3. Variablen Belegung

- 2) Wie funktioniert Pseudoparallelität?

Abwechselnde Ausführung von Prozessen mit geringem Quantum.

- 3) Zeichnen sie ein Zustandsdiagramm für Prozesse mit Nichtpräemptivem und präemptivem Scheduling, mit und ohne Auslagerung.



Kapitel 5: Nebenläufigkeit

- 1) Erläutern sie den Begriff „Racecondition“.

Zwei oder mehr Prozesse benutzen die selbe Ressource und das Ergebnis hängt von der Ausführungsreihenfolge ab.

- 2) Nennen sie die 4 Anforderungen an Lösungen für das Problem der kritischen Region.

1. Wechselseitiger Ausschluss
2. Kein Prozess darf ewig warten auf Eintritt in die Kritische Region
3. Keine Annahme über Geschwindigkeit und Anzahl der Prozessorkerne
4. Prozess ausserhalb der kritischen Region dürfen keine anderen Blockieren

- 3) Wie wollen die Tutoren unbedingt einen Widerspruchsbeweis strukturiert haben?

Siehe 4.)

- 4) Beweisen sie, dass durch diese Implementierung ein wechselseitiger Ausschluss garantiert ist. Welcher Nachteil ergibt sich durch diese Realisierung?

<pre>/* Prozess 0 */ wiederhole { solange (turn ≠ 0) tue nichts; /* kritische Region */ turn := 1; /* nicht-kritische Region */ }</pre>	<pre>/* Prozess 1 */ wiederhole { solange (turn ≠ 1) tue nichts; /* kritische Region */ turn := 0; /* nicht-kritische Region */ }</pre>
---	---

Beweis:

Behauptung: Der Wechselseitige Ausschluss ist garantiert.

Beweis: Durch Widerspruch

Annahme: o.B.d.A geht P1 zuerst in die kritische Region zum Zeitpunkt t_1 . P0 geht nach P1 zum Zeitpunkt t_0 in die kritische Region ein. Zu t_1 ist $turn=1$. Folglich muss zwischen t_0 und t_1 muss $turn$ auf 0 gesetzt werden.

Widerspruch: $turn$ kann nicht auf 0 gesetzt werden, während P1 in der kritischen Region ist.

Schlussfolgerung: Behauptung wahr.

Der Eintritt in die kritische Region ist nur abwechselnd möglich. Prozesse können verhungern.

- 5) Nennen sie den Nachteil von reinen Softwarelösungen für die Lösung für das Problem der kritischen Region.

Aktives Warten.

- 6) Nennen und erklären sie einen der Assembler Befehle der für eine atomare Ausführung sorgt.

TSL (Test and set Lock) und bei Intel x86: XCHG. Es das Setzen und Lesen der Lockvariable atomar ausgeführt.

- 7) Welchen Nachteile birgt eine reine Hardwarelösung?

Aktives Warten.

8) Welche Daten (3) und Operationen (2) besitzt ein Mutex?

Daten:

1. ID
2. LOCK
3. queue

Operationen

1. mutex_unlock(ID)
2. mutex_lock(ID)

9) Welche Situation beschreibt a) $count_s < 0$ b) $count_s = 0$ und c) $count_s > 0$ bei einer Semaphore s?

- a) Weitere Weckrufe stehen schon aus, nächster Prozess legt sich auch schlafen
- b) Keine Weckrufe sind bisher gespeichert, nächster Prozess legt sich schlafen
- c) Frei, nächster Prozess darf fortfahren.

10) Implementieren sie eine Lösung für das Produzenten-Konsumenten-Problem für 2 Produzenten, 2 Konsumenten und einer Buffergröße von n, der Buffer sei Anfangs leer. Achten sie auf die Initialisierung ihrer Semaphoren.

```
semaphore exclu; count_exclu = 1; // "mutex"-Sem. für krit. Regionen
semaphore empty; count_empty = n // zählt freie Plätze
semaphore full; count_full = 0; // zählt belegte Plätze
```

Prozedur **producer**

```
{
    wiederhole
    {
        → item = produce_item(); // produziere nächstes Objekt
        down(empty);
        down(exclu);
        insert_item(item);        // füge Objekt in Puffer ein
        up(exclu);
        up(full);
    }
}
```

Prozedur **consumer**

```
{
    wiederhole
    {
        down(full);
        down(exclu);
        item = remove_item();    // entferne Objekt aus Puffer
        up(exclu);
        up(empty);
        consume_item(item);      // konsumiere Objekt
    }
}
```

Kapitel 6: Deadlocks

- 1) Welche 4 Voraussetzungen müssen für das Auftreten von Deadlocks gegeben sein?
 1. Wechselseitiger Ausschluss
 2. Zyklisches Warten: Es gibt eine zyklische Kette von Prozessen, von denen jeder auf eine Ressource wartet, die dem nächsten Prozess in der Kette gehört
 3. Besitzen und Warten: Prozesse, die schon Ressourcen reserviert haben, können noch weitere Ressourcen anfordern
 4. Kein Ressourcenentzug: Ressourcen, die einem Prozess bewilligt wurden, können nicht gewaltsam wieder entzogen werden
- 2) Wann existiert immer eine Ausführungsreihenfolge die zu keinem Deadlock führt?

Genug Ressourcen für sequenzielle Ausführung der Prozesse.
- 3) Welche Voraussetzungen benötigt der Bankieralgorithmus um eine Deadlock freie Ausführungsreihenfolge zu garantieren (2)?
 1. Im Voraus bekannt: Welche und wie viele Ressourcen die einzelnen Prozesse maximal anfordern werden
 2. Anforderung übersteigt für keinen Prozess die zur Verfügung stehenden Ressourcen
- 4) Warum existiert für manche vom Bankieralgorithmus als unsicher eingestufte Ausführungen trotzdem eine Deadlock freie Restausführung (2 Nennungen)?

Da der Bankieralgorithmus nimmt zwei Dinge an:

 1. Alle Prozesse ihre Anforderungen auf einen Schlag stellen.
 2. Die Prozesse erst nach ihrer Terminierung die Ressourcen freigeben.
- 5) Welche Matrizen benötigt der Bankieralgorithmus?
 1. A: Noch angeforderte Ressourcen
 2. E: Erhaltene Ressourcen
 3. V: Vorhandene Ressourcen
 4. M: Maximale Anforderungen
 5. F: Freie Ressourcen
- 6) Warum ist durch den Bankieralgorithmus das Deadlockproblem nicht restlos gelöst (3 Nennungen)?
 1. Maximalen Anforderungen müssen bekannt sein.
 2. Mehr Anforderungen als Ressourcen kann es trotzdem eine deadlockfreie Restausführung geben.
 3. Anzahl der Prozesse nicht immer konstant/statisch.

7) Welche Bewältigungsstrategien gibt es für die Behebung von Deadlocks (2 Nennungen)?

1. Abbruch aller verklemmten Prozesse
2. Rückführung aller verklemmten Prozesse auf einen festgelegten Kontrollpunkt und Neustart
3. Schrittweiser Abbruch aller verklemmten Prozesse, bis die Verklemmung nicht mehr existiert
4. Schrittweiser Ressourcenentzug, bis die Verklemmung nicht mehr existiert

Kapitel 7: Scheduling

1) Nennen und beschreiben sie die 3 Arten von Scheduling.

1. Langfristiges: Welche Prozesse werden zugelassen.
2. Mittelfristiges: Welche Prozesse werden ausgelagert.
3. Kurzfristiges: Welche Prozesse bekommen CPU zugeteilt.

2) Worin liegt der Unterschied zwischen Benutzer- und Systemorientiertem Scheduling (3 Nennungen)?

Der Benutzer will eine möglichst geringe durchschnittliche Durchlaufzeit. Systemorientiert bedeutet, es soll ein möglichst hoher Durchsatz erreicht werden.

3) Bestimmen sie die Auswahlfunktion, sowie die Zuordnung zu nicht- beziehungsweise präemptivem Scheduling von folgenden Strategien:

1. First Come First Served (FCFS)
 2. Round Robin (RR)
 3. Shortest Job First (SJF)
 4. Shortest Remaining Time (SRT)
 5. Highest Response Ratio Next (HRRN)
-
1. $\max(w)$, nicht präemptiv.
 2. $\max(w)$ (nach queue), präemptiv, mit Quantum.
 3. $\min(s)$, nicht präemptiv.
 4. $\min(s-e)$, präemptiv, jedes mal wenn neuer Prozess bereit ist.
 5. $\max((w+s)/s)$, nicht präemptiv

4) Erläutern sie kurz Feedback beziehungsweise Unixscheduling.

Feedback: Mehrere Prioritäten, jeweils mit eigener Warteschleife. Innerhalb einer Priorität FCFS und in letzter RR.

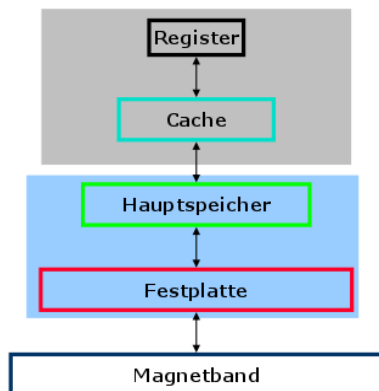
- 5) Analysieren sie die Schedulingstrategien aus Aufgabe 3 in Hinsicht auf a) Bevorzugung von kurzen/langen Prozessen, b) Livelock Gefahr, c) eventuellen Wissensanforderungen und d) Effektivität im Sinne der Benutzer- beziehungsweise Systemorientierter Anwendung.

	a) Bevorzugung	b) Livelockgefahr	c) Wissensanforderung	d) Benutzer/System
FCFS	Bevorzugt lange	Nein	Nein	Für beide nicht interessant
RR	Bevorzugt Prozesse ohne E/A	Nein	Nein	Nicht sehr interessant
SJF	Bevorzugt kurze Prozesse	Ja, für lange Prozesse	Die Gesamtlaufzeit muss bekannt sein	Gut für Benutzer
SRT	Bevorzugt kurze	Ja, für lange Prozesse	Die Gesamtlaufzeit muss bekannt sein	Beste für Benutzer
HRRN	Bevorzugt kurze	Nein	Ja, Gesamtlaufzeit muss bekannt sein	Ausgeglichen

o

Kapitel 8: Speicherverwaltung

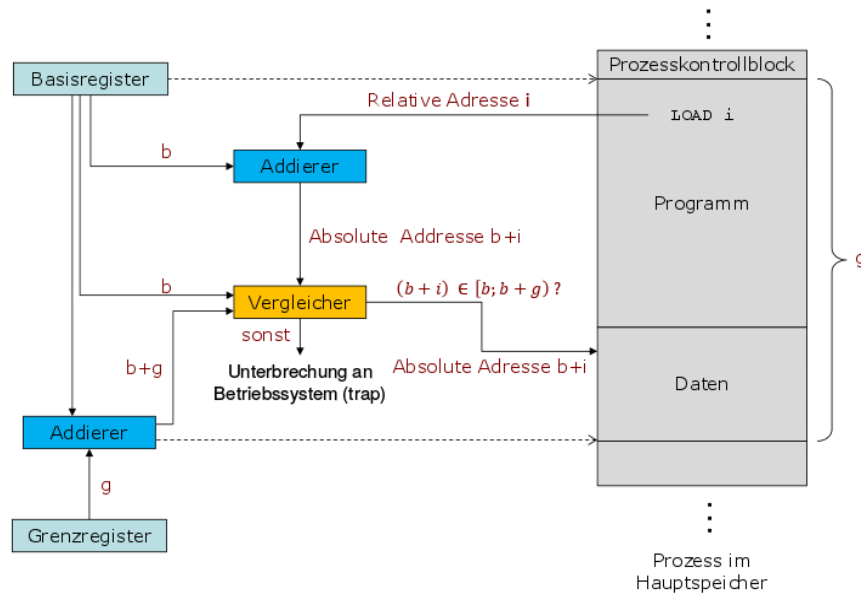
- 1) Skizzieren sie die Speicherhierarchie (5 Nennungen).



- 2) Welche 5 Anforderungen werden an die Speicherverwaltung gestellt? Erläutern sie diese kurz.
1. **Relokation:** Auslagern und Einlagern von Prozessen aus dem Hauptspeicher muss möglich sein.
 2. **Schutz:** Namensräume von Prozessen müssen vor Zugriffen durch andere Prozesse gesichert sein.
 3. **Gemeinsame Nutzung:** Prozesse können über ein Shared-Memory kollaborieren.
 4. **Physische Organisation:** Implementierung des Speichers.
 5. **Logische Organisation:** Modularität möglich.
- 3) Wie stehen physikalische, logische und relative Adressen im Zusammenhang?

Die physikalische ist reale Adresse im Speicher und die logische ist diejenige, die innerhalb eines Programmes verwendet wird. Die relative ist ein Spezialfall der logischen Adresse, es wird bezug auf (z.B) Programmanfang genommen.

4) Wie können Namensräume von Prozessen vor einander geschützt werden? Skizzieren sie.



5) Welche Vor- und Nachteile treten bei den drei Varianten der Partitionierung auf? (Jeweils 3, 3 und 1 Nennungen).

1. Statische Partitionierung (feste und variable Größe)
 - Vorteile: Keine externe Fragmentierung, Geringer Verwaltungsaufwand.
 - Nachteile: Begrenzte Anzahl und Größe von Prozessen, hohe Interne Fragmentierung.
2. Dynamische Partitionierung
 - Vorteile: Keine interne Fragmentierung, Prozesse variabler Länge.
 - Nachteile: Hohe externe Fragmentierung nach Ein-/Auslagerung und hoher Verwaltungsaufwand.
3. Buddy System
 - Vorteile: Weniger als halbe Blockgröße interne Fragmentierung. Externe Fragmentierung löst sich selbstständig auf. Geringer Verwaltungsaufwand.
 - Nachteile:

6) Nennen sie die drei Speicherzuteilungsalgorithmen, welcher ist am effektivsten? Welche Nebeneffekte treten bei den anderen beiden auf?

1. Next Fit: Etwas schlechter als First Fit, typischer Effekt: Fragmentierung am Ende des Speichers.
2. First Fit: Ist beste.
3. Best Fit: Suche braucht Zeit. Hohe Fragmentierung durch viele kleine Partitionen.

7) Welche Größe muss das Offset-Feld beim einfachen Paging mindestens haben, wenn die Seitengröße 2^{11} Bit beträgt (Zeilenlänge 2^4 Bit)?

8) Erklären sie den Begriff des virtuellen Speichers. Welche Vor- und Nachteile ergeben sich dadurch (jew. 2 Nennungen)? Durch welches Prinzip ist die Verwendung trotzdem sinnvoll? Was würde ohne dieses Prinzip passieren?

Virtueller Speicher ist die Einheit aus Hauptspeicher und Swapping Area.

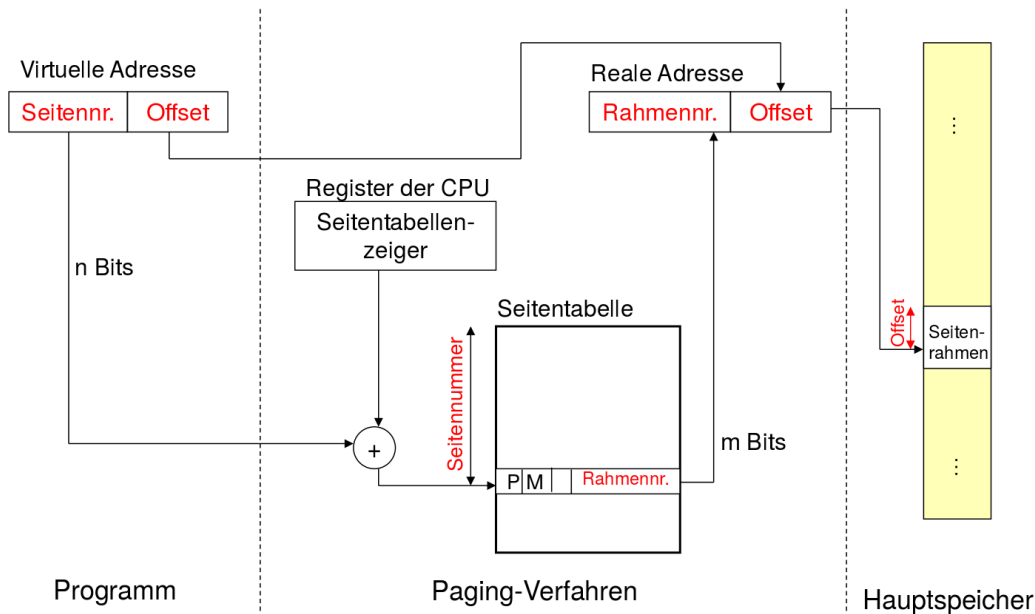
- Vorteil: Mehr Speicher, größerer Namensraum, Größe von Prozessen müssen nicht im voraus bekannt sein.
- Nachteile: Höherer Verwaltungsaufwand durch Ein-/Auslagerung,

Sinnvoll durch das Prinzip der Lokalität, da sonst zuviel Thrashing stattfinden würde.

9) Welche Einträge besitzt die Seitentabelle bei Paging mit virtuellem Speicher?



10) Skizzieren sie die Adressumsetzung für Paging mit virtuellem Speicher.



11) Beschreiben sie kurz das Prinzip der mehrstufigen Seitentabellen.

Zuerst wird die Hauptseitentabelle adressiert und von dort die Adresse der richtigen Untertabelle entnommen. Dann analog zum einfachen Paging.

12) Beschreiben sie kurz das Prinzip der invertierten Seitentabellen, gehen sie dabei auch auf die Adressumsetzung ein.

Für eine gegebene Seitennummer wird über eine Hash-Funktionen die Stelle in der Seitentabelle berechnet, an der die entsprechende Rahmennummer steht. Sofern hier die Rahmennummer einer anderen Seite steht, wird die Überläuferkette verfolgt, bis Rahmennummer zur gegebenen Seitennummer ausgegeben werden kann.

13) Beschreiben sie die Grundidee der Speicherverwaltung durch Segmentierung.

Dynamisch wachsend und schrumpfende Speicherabschnitte bilden Segmente. Adressumsetzung ist ähnlich dem des Paging.

14) Nennen sie 2 Austauschstrategien.

1. **F**irst **I**n **F**irst **O**ut (FIFO)
2. **L**east **R**ecently **U**sed (LRU)
3. Clock-Algorithmus

Kapitel 9: Sicherheit

- 1) Worin besteht der Unterschied zwischen Betriebs- und Angriffssicherheit?

Die Betriebssicherheit bezieht sich auf Aspekte, die ohne das Zutun äußerer Einflüsse entstehen können. Die Angriffssicherheit bezieht sich nur auf eben diese.

- 2) Nennen sie die 4 Sicherheitsziele.

1. Privacy
2. Datenintegrität
3. Systemverfügbarkeit
4. Vertraulichkeit der Daten (data confidentiality)

- 3) Nennen sie 2 Verschlüsselungsstrategien.

1. Symmetrisch Verschlüsselungsstrategien (Caesar-Cypher)
2. Asymmetrisch Verschlüsselungsstrategien (RSA)