# C2 Organisation des Hauptspeichers, Prozessverwaltung

Die Organisation des Hauptspeichers hängt stark ab von der Rechnerarchitektur.
Dabei gibt es folgende Möglichkeiten:

1. Für alle Prozesse gibt es nur einen Adressraum
2. Jeder Prozess hat seinen eigenen Adressraum oder evtl. sogar mehrere.

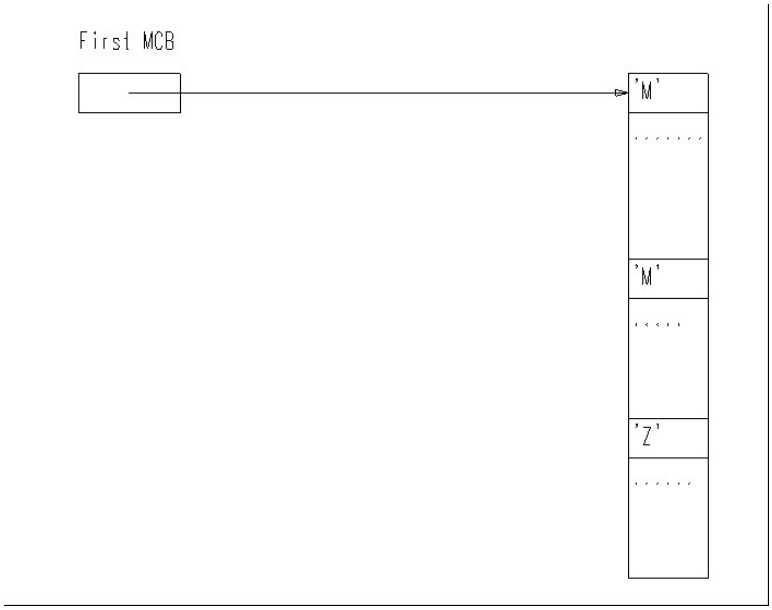
Bei (2) hat man i.A. eine Hardwareunterstützung zur Umsetzung der logischen Adressen in die physikalischen das **Paging**. Dabei muss einem logischen Adressbereich kein zusammenhängender physikalischer Adressbereich zugeordnet werden. Die Umsetzung der logischen Seitenadresse in die physikalische Kacheladresse erfolgt mit Hilfe einer Seiten-Tabelle. Diese enthält für jede Seite die physikalische Kachelnummer oder eine Referenz auf dem Platz im Pagingbereich auf der Platte, wenn die Seite ausgelagert oder überhaupt noch nicht geladen wurde.
Beispiel:
32 Bit logische Adresse=
20 Bit Seitennummer+ 12 Bit Adresse in der Seite (Seitengröße 4 KByte)
30 Bit physikalische Adresse ( Hauptspeicherbereich 1 GByte)=
18 Bit Kachelnummer+ 12 Bit Adresse in der Seite

Hierbei muss allerdings die Seitentabelle bereits 1 Mio. Einträge mit je 4 Byte enthalten. Daher benutzt man häufig ein 2-schichtiges Verfahren.
32 Bit logische Adresse=
10 Bit Segmentnummer+ 10 Bit Seitennummer+ 12 Bit Adresse in der Seite
Für jedes ganz oder teilweise geladene Segment wird eine Seitentabelle mit je 1024 Einträgen benötigt. Ferner die Segmenttabelle mit 1024 Einträgen, die jeweils auf die Seitentabelle verweisen.

Bei (1) dagegen benötigt man jeweils wechselnde zusammenhängende Adressbereiche.

Bei MS-DOS wurde hierfür folgende Struktur benutzt:

```
type mcb=record
 Typ:char; {'M' in Kette 'Z' Letzter Eintrag  Mark Zbikowski}
 owner:word; { PSP des Besitzers }
 size:word; { in 16 Byte Einheiten }
 ....
end;
```




Wird ein Speicherplatz gesucht, muss ein Element mit dem Owner=0 gefunden werden.
Ein Speicherplatz kann nur verlängert werden, wenn dahinter ein Bereich mit owner=0 ist.
Bei Beendigung eines Programmes (DOS-Aufruf EXIT) werden alle Speicherbereiche, bei denen das Programm der Owner ist, freigegeben. Haben 2 benachbarte Bereiche owner=0, werden sie zusammengelegt.
Bei der Belegung eines freien Speicherbereiches gibt es 2 Strategien:

- First-Fit: Es wird der erste ausreichend große Bereich genommen
- Best-Fit: Es wird der kleinste, aber noch ausreichend große Bereich genommen.

Best Fit ist dabei nicht unbedingt immer besser als First-Fit, da kleine Restbereiche entstehen, die nicht mehr genutzt werden können. Reicht der vorhandene Speicher nicht mehr aus, müssen Programme auf der Platte ausgelagert werden. Soll das Programm später in einem anderen Speicherbereich fortgesetzt werden, muss das Programm relocatable sein, d.h. alle Adressen müssen relativ zu einer Anfangsadresse interpretiert werden (Programmrelativ).

Hat man für einen Prozess mehrere eigene Adressräume, können dies sein:

- Codebereich
- Daten
- Stack(Stapel)
- Halde

Jeder der Bereiche bekommt bestimmte Attribute.
Der Codebereich kann nur ausgeführt werden, er kann weder gelesen noch beschrieben werden.
Die übrigen Bereiche können nur gelesen oder geschrieben werden, aber nicht ausgeführt werden. Kann der Stackbereich nicht ausgeführt werden, ist eine Möglichkeit, mit Hilfe eines Stacküberlaufs eine Sicherheitslücke zu nutzen, verbaut.
Da in den Stack jedoch auch geschrieben werden muss, ist es weiter möglich eine Rückkehradresse auf einen anderen Codebereich umzubiegen. Um dies zu verhindern, muss im Compiler Code erzeugt werden, der entsprechende Prüfungen vornimmt.

Ein Linux-Projekt in dieser Richtung finden Sie unter:
www.trustedlinux.org

Literatur zu Buffer-Overflows:
c't 23/2001, S. 216: Stephan Kallnik, Daniel Pape, Daniel Schröter, Stefan Strobel: Buffer-Overflows und wie man sich davor schützt

Für jeden Prozess hält das Betriebssystem eine Prozessbeschreibung.
Ein solcher Prozessbeschreibungsblock enthält u.a. folgende Informationen:

- Zeiger auf den Aufrufer
- Zeiger auf den Job-File-Table (JFT)
- Zeiger auf die Umgebungsvariablen
- Parameterstring beim Aufruf

Arbeitet man mit Paging braucht bei Speichermangel nicht der gesamte Adressbereich ausgelagert werden, sondern nur einzelne Seiten. Wird später eine logische Seite referiert, der keine physikalische Seite (Kachel) zugeordnet ist, wird eine Exception ausgelöst. Das Betriebssystem lädt dann die fehlende Seite. Zur Entscheidung, welche Seite ausgelagert werden kann, gibt es 2 Strategien:

- Häufigkeit (Least-Frequently-Used LFU )
  Es wird die Seite ausgelagert, die am wenigsten Häufig benutzt wurde, in der Annahme, dass sie auch in Zukunft nur selten benutzt wird.
- Zeitpunkt (Least-Recently-Used LRU )
  Es wird die Seite ausgelagert, die am längsten nicht mehr benutzt wurde, in der Annahme, dass sie auch in Zukunft nicht mehr benötigt wird.

Beide Strategien setzen voraus, dass die Hardware in der Seitenbeschreibung entsprechende Informationen bereithält, die bei jedem Seitenzugriff aktualisiert werden.

Von dem Auslagern der Prozesse auf die Platte zu unterscheiden ist der Cache-Speicher (Pufferspeicher). Dieser befindet sich zwischen Prozessor und Hauptspeicher und hält Kopien einzelner Hauptspeicherinhalte für schnellere Prozessorzugriffe bereit.

Bei einem Single-Task-System, wie MS-DOS kennt das Betriebssystem die Adresse der aktuellen Prozessbeschreibung. Der aktuelle Prozess kann über einen Systemaufruf (SetCurrentProcess) geändert werden. Wird der Prozess beendet, wird der aufrufende Prozess der aktuelle Prozess. Vorher werden alle Speicherbereiche freigegeben und alle offenen Dateien werden geschlossen.

Laufen in einem Multi-Task-System, wie UNIX mehrere Prozesse gleichzeitig, können durch Reservieren von Ressourcen Deadlocks entstehen.
Beispiel: Es stehen 2 Geräte A und B zur Verfügung.
Prozess 1 reserviert Gerät A
Prozess 2 reserviert Gerät B
Prozess 1 möchte Gerät B (wird angehalten, weil nicht frei)
Prozess 2 möchte Gerät A (wird ebenfalls angehalten, weil nicht frei)

Dieser Deadlock kann nur dadurch vermieden werden, dass bei Prozessstart alle benötigten Ressourcen bekannt sind und für den Prozess reserviert werden.