

Einführung in die technische Informatik

Christopher Kruegel chris@auto.tuwien.ac.at

<http://www.auto.tuwien.ac.at/~chris>

Allgemeines

- Was kommt zum Test?
 - Buch „Einführung in die technische Informatik“
Ab *inklusive* Kapitel 4.4
 - Vorlesung
 - Wissensfragen und Rechenbeispiele (Verhältnis: 60% - 40%)
- Wissensfragen
 - detailliert (!)
 - stichwortartig zu beantworten
 - Akronyme und Begriffe (z.B., reentrant, preemptiv)
 - Aufzählungen (z.B., page replacement algorithms)

Betriebssysteme

Automation Systems Group

- Aufgaben
 - Management von Ressourcen
 - Präsentation einer einheitlichen Schnittstelle für Anwendungen
- 4 wichtige Gebiete
 - *Prozessmanagement*
 - *Speichermanagement*
 - Dateisystem
 - Eingabe und Ausgabe
- System Calls
 - Funktionen, die ein Betriebssystem den Anwendungen zur Verfügung stellt

Einführung in die technische Informatik

3

Prozessmanagement

Automation Systems Group

- Konzept
 - Betriebssystem stellt eine Reihe von virtuellen Computern zur Verfügung
 - Ausführung eines Programms auf einem dieser virtuellen Computer heißt **Prozess**
 - virtuelle Computer vermitteln den Eindruck, dass jeder Prozess auf einer eigenen CPU mit eigenem Speicher läuft
 - in Wirklichkeit wird zwischen den Prozessen hin- und hergeschaltet und diese laufen auf einer einzigen realen CPU
- Quasi-Parallelität oder logische Parallelität

Einführung in die technische Informatik

4

Prozessmanagement

Automation Systems Group

- Programm
 - statisches Objekt im Dateisystem
 - Folge von Instruktionen
 - unveränderlich über die Zeit
- Prozess
 - dynamische Abbild eines Programms
 - Folge von ausgeführten Instruktionen auf der CPU
 - existiert nur für eine bestimmte Zeit
 - mehrere Prozesse können das selbe Programm ausführen

Prozessmanagement

Automation Systems Group

- Prozessablauf
 - Prozesse können erzeugt werden
 - durch das Betriebssystem
 - `init task`
 - durch andere Prozesse
 - `fork()`
 - Prozesse können beendet werden
 - freiwillig
 - `exit()`
 - wegen eines Fehlers (Absturz)
 - durch einen anderen Prozess
 - `kill()`

Prozessmanagement

Automation Systems Group

- Prozesszustand
 - Prozesse kann auf der CPU exekutiert werden (RUNNING)
 - Prozess bereit sein, auf der CPU exekutiert zu werden (READY)
 - Prozess kann auf externe Ereignisse warten (BLOCKED)
- Zustandsübergänge
 - ausgelöst durch
 - Prozess selbst (Termination)
 - Scheduler
 - Ressourcen werden verfügbar
 - beschrieben durch Zustandsgraph

Prozesse

Automation Systems Group

- Betriebssystem verwaltet Prozesse mittels *process table*
- Process table
 - speichert Prozessdeskriptor (oder process control block) für jeden Prozess
- Prozessdeskriptor speichert relevante Informationen eines Prozesses
 - Prozess-ID
 - Zustand
 - Priorität
 - Besitzer des Prozesses (z.B., UID)
 - Zugriffsrechte (z.B., EUID)
 - Context (Registerinhalte)
 - Verweise auf benutzte Speicherbereiche
 - Verweise auf offene Files

Prozesse

- Prozesse
 - besteht aus Ressourcen (Speicherbereich, Dateien)
 - Ausführung von Instruktionen (threads of execution)
- Zustand des Prozesses bei der Ausführung
 - Werte in den Registern (speziell des Program Counters)
 - Verlauf (History) der Ausführung und lokale Variablen am Stack
- *Threads*
 - mehrere, parallele Ausführungen von Instruktionen, aber mit den selben Ressourcen

Threads

- Nützlich für Anwendungen mit mehreren Aufgaben
 - ein Prozess wartet auf Eingabe, ein anderer führt Berechnungen durch
 - z.B., Server Applikationen
- Threads sind wegen des gemeinsamen Adressraums
 - effizienter zu koordinieren
 - effizienter zu erzeugen
 - besser beim Wechsel zwischen Threads (context switch)
- Natürlich kann man alles auch nur mit Prozessen realisieren
 - IPC (inter-process communication)

Scheduling

- Viele Prozesse, aber nur eine CPU
 - Betriebssystem muss zwischen Prozessen wechseln
 - Betriebssystem braucht gelegentlich die CPU selbst
- Wechsel ist aufwendig
 - Zustand der Ausführung (Register) des alten Prozess muss gespeichert werden
 - neuer Prozess muss ausgewählt werden
 - Zustand des neuen Prozess muss geladen werden
- Auswahl des neuen Prozesses
 - Scheduling

Scheduling

- Ziele
 - Durchsatz (hoch)
 - Antwortzeit (niedrig)
 - CPU Auslastung
 - Fairness
- Unterscheidung
 - nicht-preemptive Verfahren
 - ein Prozess kann die CPU behalten, bis er fertig ist, oder freiwillig die Kontrolle abgibt
 - preemptive Verfahren
 - ein Prozess kann vom Betriebssystem gegen seinen Willen unterbrochen werden

Scheduling

- Algorithmen
 - first-come, first-serve
 - round-robin
 - priority scheduling
 - shortest job first
 - shortest remaining time
- Scheduling wird festgelegt durch
 - Angabe von den Zeitpunkten, zu denen Prozesse zu laufen beginnen (sich im Zustand RUNNING befinden)
- maximal ein Prozess kann sich im Zustand RUNNING befinden
- mehrere Prozesse können auf die CPU warten (READY)

Scheduling

- Praktisches Beispiel
- Angabe
 - Art des Scheduling Algorithmus
 - Angabe ob preemptives oder nicht-preemptives Scheduling vorliegt
 - Liste von Prozessen mit Information
 - wann ein Prozess in den Zustand READY wechselt
 - möglicherweise wie lange ein Prozess die CPU benötigt
 - möglicherweise Priorität
- Gesucht
 - Zeitpunkte, wann welche Prozesse zu laufen beginnen

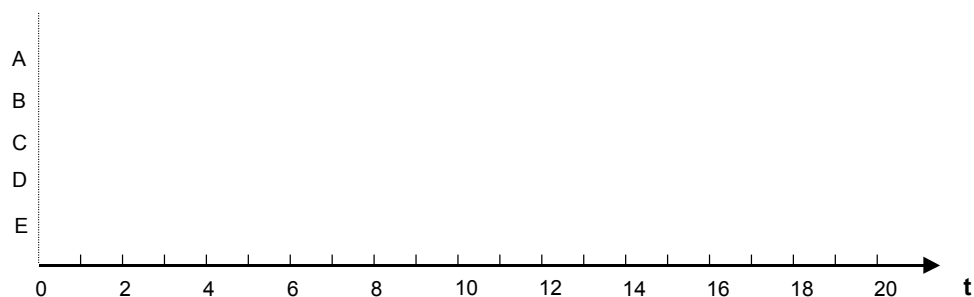
Scheduling

- Beispiel
 - nicht-preemptives priority scheduling
 - höherer Prioritätswert bedeutet höhere Priorität

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |

Scheduling

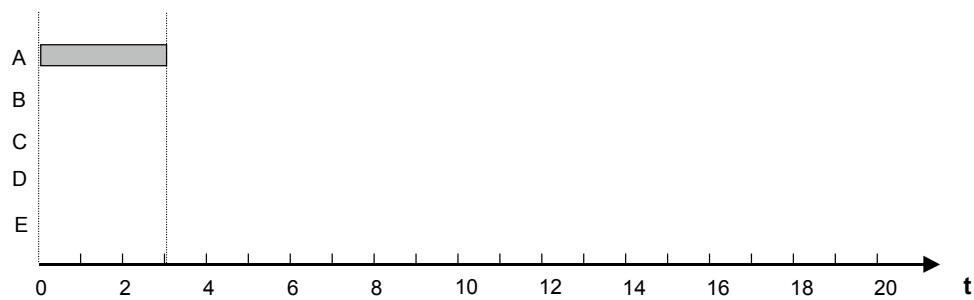
| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



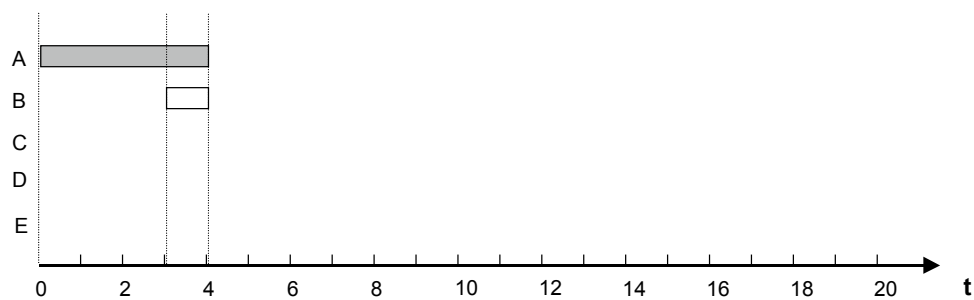
Einführung in die technische Informatik

17

Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



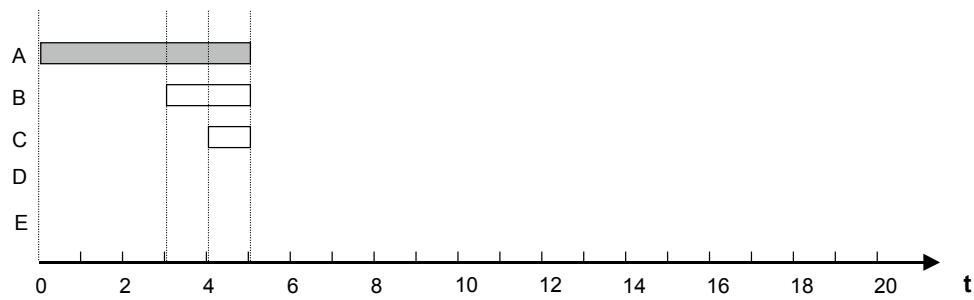
Einführung in die technische Informatik

18

Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



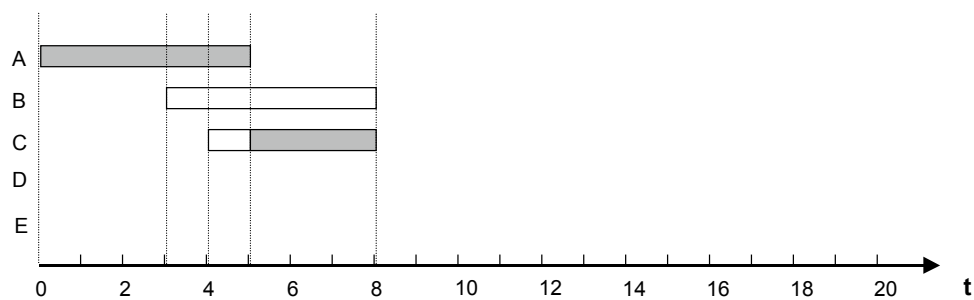
Einführung in die technische Informatik

19

Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



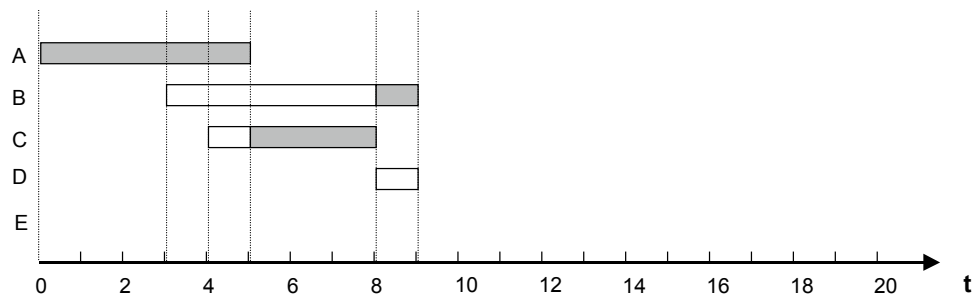
Einführung in die technische Informatik

20

Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



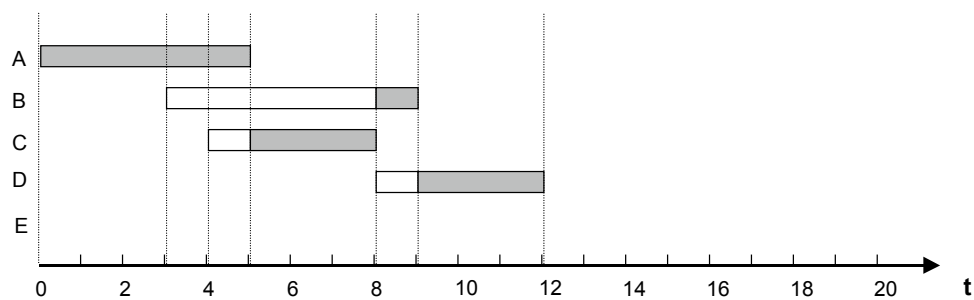
Einführung in die technische Informatik

21

Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |

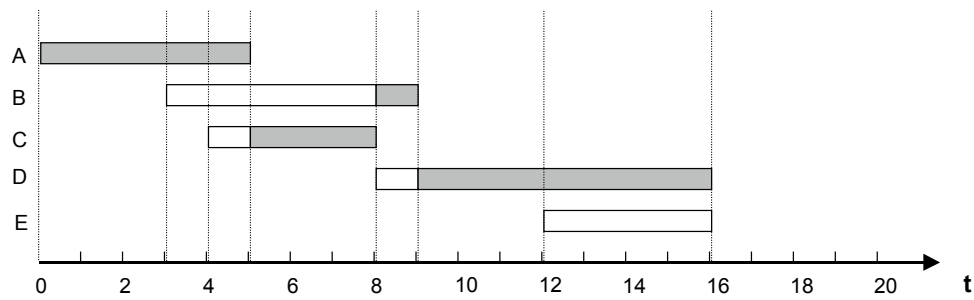


Einführung in die technische Informatik

22

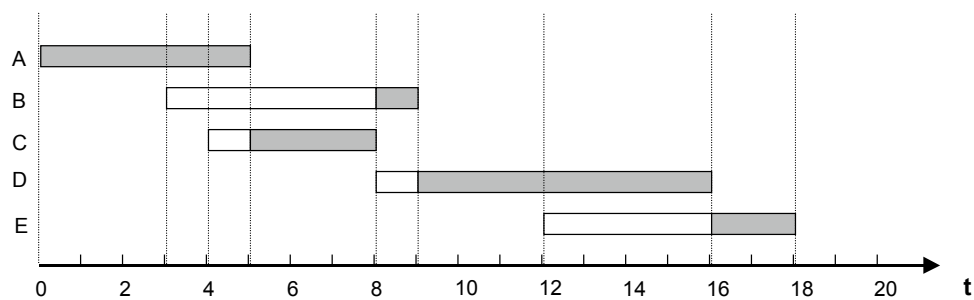
Scheduling

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |

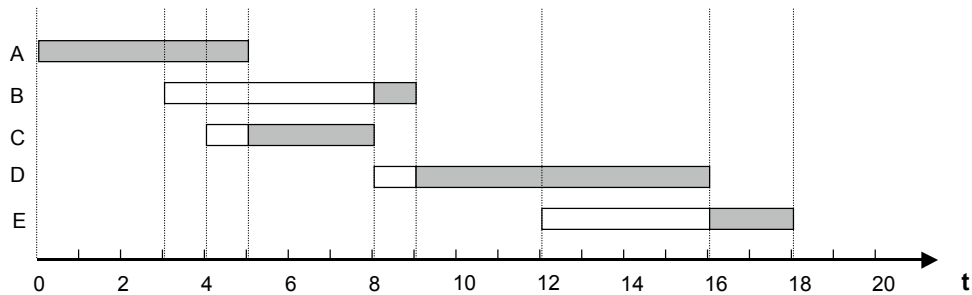


Scheduling

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



Scheduling



| Prozess | A | C | B | D | E |
|---------------------|---|---|---|---|----|
| Zeitpunkt (RUNNING) | 0 | 5 | 8 | 9 | 16 |

Scheduling

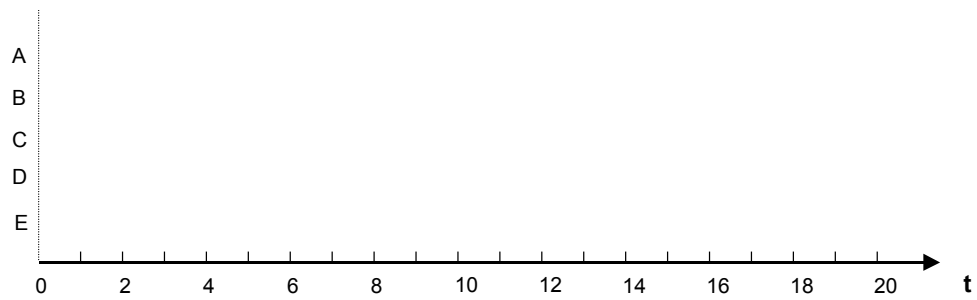
- Beispiel
 - *preemptives* priority scheduling
 - höherer Prioritätswert bedeutet höhere Priorität

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |

Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



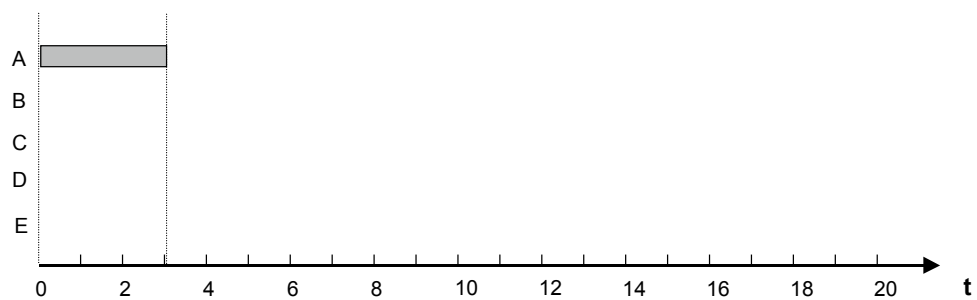
Einführung in die technische Informatik

27

Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



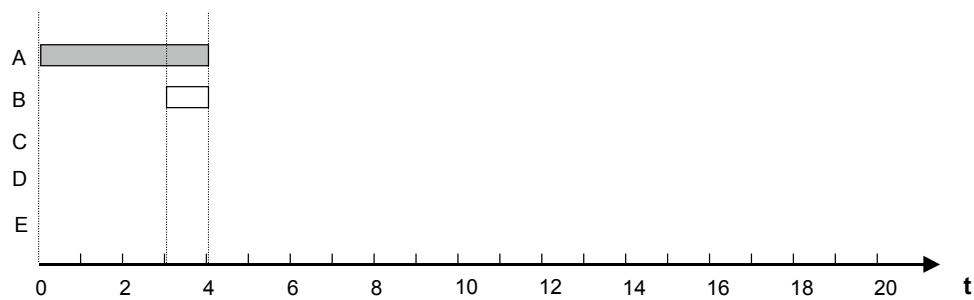
Einführung in die technische Informatik

28

Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



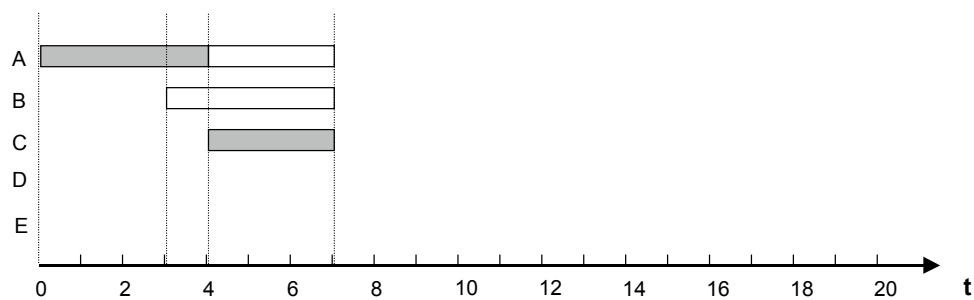
Einführung in die technische Informatik

29

Scheduling

Automation Systems Group

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |

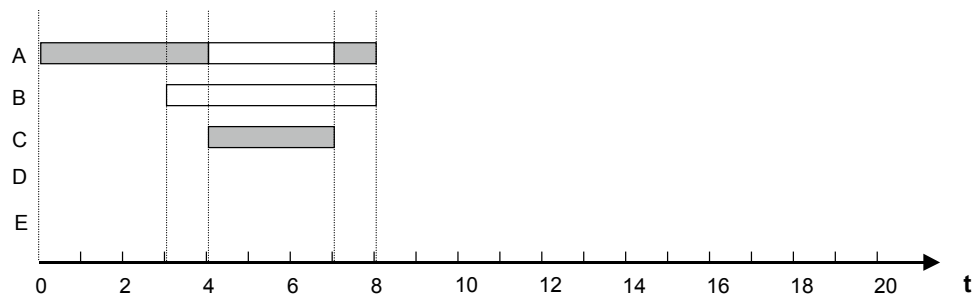


Einführung in die technische Informatik

30

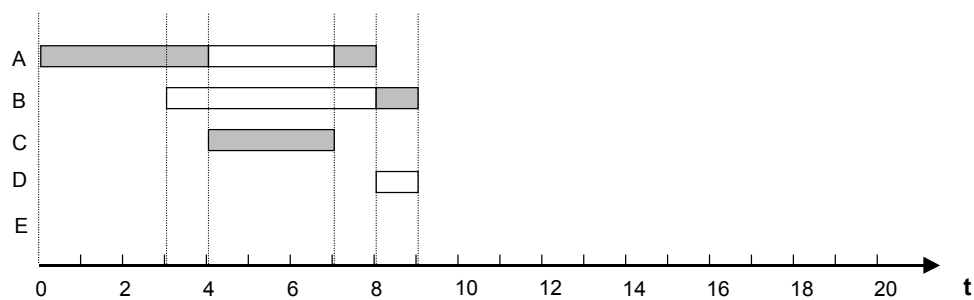
Scheduling

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



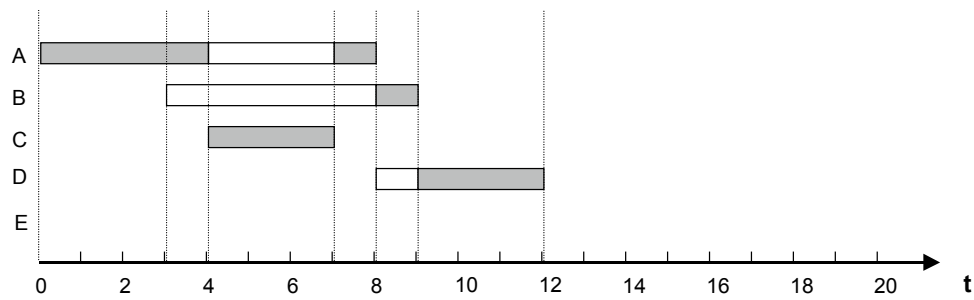
Scheduling

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



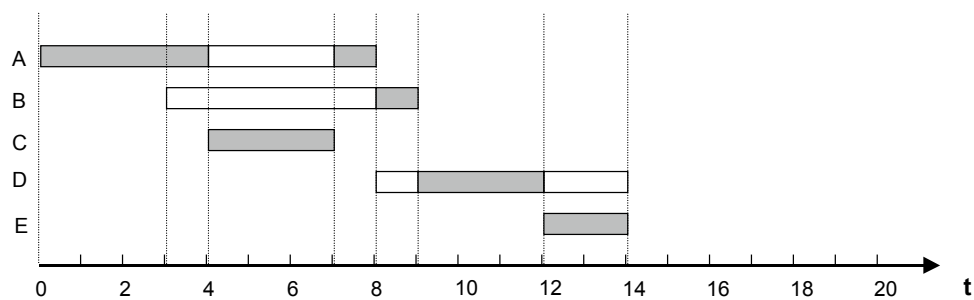
Scheduling

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



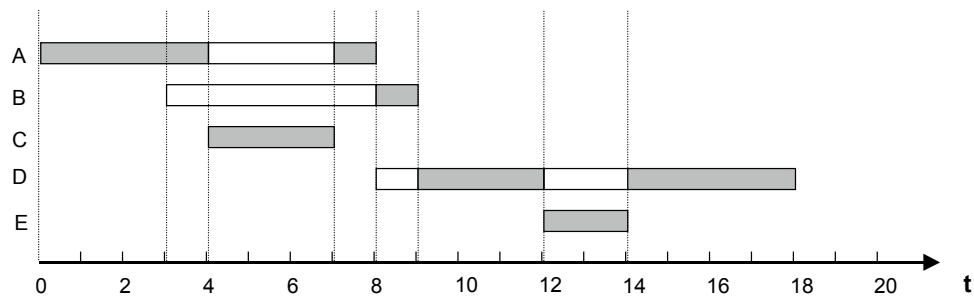
Scheduling

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |

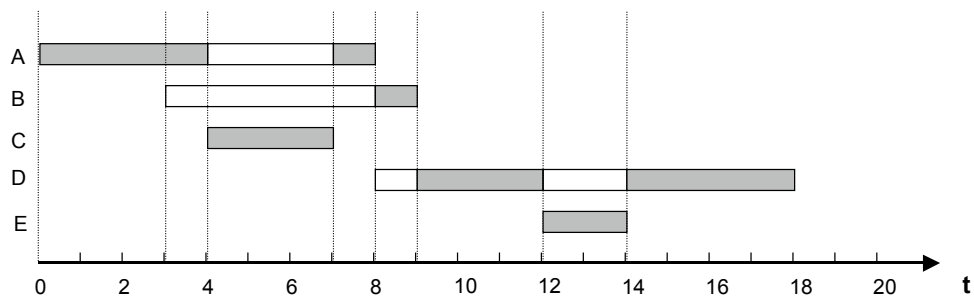


Scheduling

| Prozess | Startzeitpunkt | Laufzeit | Priorität |
|---------|----------------|----------|-----------|
| A | 0 | 5 | 2 |
| B | 3 | 1 | 1 |
| C | 4 | 3 | 4 |
| D | 8 | 7 | 0 |
| E | 12 | 2 | 3 |



Scheduling



| Prozess | A | C | A | B | D | E | D |
|---------------------|---|---|---|---|---|----|----|
| Zeitpunkt (RUNNING) | 0 | 4 | 7 | 8 | 9 | 12 | 14 |

Speicherverwaltung

Automation Systems Group

- Ideal
 - Speicher ist groß
 - Speicher ist schnell (niedrige Zugriffszeiten)
- Praxis
 - Speicherhierarchie
 - schneller Cache
 - mittlerer Hauptspeicher
 - langsamer Plattenspeicher
 - physikalischer Speicherplatz muss unter Prozessen aufgeteilt werden

Speicherverwaltung

Automation Systems Group

- das Betriebssystem stellt jedem Prozess exklusiv einen riesigen Speicherbereich zur Verfügung
 - üblicherweise, das Maximum des adressierbaren Bereichs
 - bei 32-bit Architektur, d.h., 32-bit Adressen → 4 Gbyte
 - dieser Speicher ist allerdings nicht wirklich vorhanden, also nur virtuell
 - virtuelle Adressen
- physikalischer Speicher ist begrenzt und muss unter allen Prozessen aufgeteilt werden
 - physikalische Adressen

Speicherverwaltung

Automation Systems Group

Lösung des Problems

- Aufteilen des virtuellen Adressraums in kleinere Stücke
- diese Stücke können nicht mehr weiter unterteilt werden, und werden als ganzes in den physikalischen Teil geladen
- 2 Probleme
 1. Wie unterteile ich den virtuellen Adressraum?
 2. Wie bekomme ich die richtige physikalische Adresse, wenn ich eine virtuelle Adresse gegeben habe?

Speicherverwaltung

Automation Systems Group

1. Problem: Aufteilung des virtuellen Adressraums

- 2 Möglichkeiten
 1. Aufteilung in unterschiedlich grosse Teile, die direkt auf Teile des Programms abgebildet werden können (z.B., Codebereich, Stack)
 - Aufteilung erfolgt nicht transparent
 - Teile (segments) sind nicht gleich groß
 - [Segmentierung](#)
 2. Aufteilung in gleich grosse Teile, die vom Programm unabhängig sind
 - Aufteilung erfolgt transparent
 - Teile (pages, frames) sind gleich groß
 - [Paging](#)

Speicherverwaltung

Automation Systems Group

2. Problem: Umsetzen einer virtuellen Adresse in die passende physikalische Adresse

- jede virtuelle Adresse liegt genau in einem Segment oder in einer Page
- virtuelle Adresse kann angegeben werden als
 - Start-Adresse des entsprechenden Segments (der entsprechenden Page)
plus ein Offset (Abstand zur Start-Adresse)
- Beispiel mit Paging
 - Gegeben ist eine virtuelle Adresse = $0x1234$ (4660)₁₀
 - Pages sind $0x100$ (256)₁₀ groß
 - wie lautet die Start-Adresse und der Offset?

Speicherverwaltung

Automation Systems Group

- Beispiel mit Paging (Lösung)
 1. Page-Nummer berechnen
$$0x1234 / 0x100 = 0x12$$
 (18)₁₀
 2. Start-Adresse berechnen
$$0x100 * 0x12 = 0x1200$$
 (4608)₁₀
 3. Offset berechnen
$$0x1234 - 0x1200 = 0x34$$
 (52)₁₀
$$0x1234 = \underbrace{(0x100 * 0x12)}_{\text{Start-Adresse}} + \underbrace{0x34}_{\text{Offset}}$$

Speicherverwaltung

Automation Systems Group

- Was bringt diese Darstellung als Start-Adresse und Offset?

Wenn ein Segment (oder Page) in den physikalischen Speicher geladen ist (beginnend bei Adresse A), dann muss nur die virtuelle Start-Adresse durch die physikalische Start-Adresse A ersetzt werden, um die Umsetzung zu erledigen.

- Vorteil
 - alle Adressen eines Segments (einer Page) können mit einer Operation umgewandelt werden
- Fragen
 - Wie bekomme ich die physikalische Start-Adresse, wenn ich die virtuelle Start-Adresse habe?
 - Wer genau macht diese Umsetzung?

Einführung in die technische Informatik

43

Speicherverwaltung

Automation Systems Group

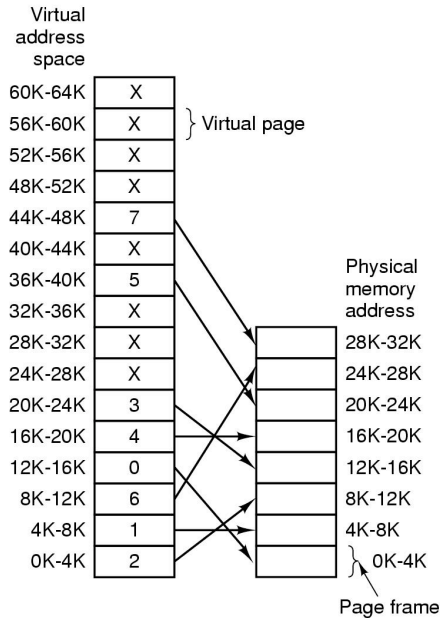
- Paging
 - virtuelle Adressen sind in Pages unterteilt
 - typischerweise zwischen 512 Bytes und 4 KBytes groß
 - physikalischer Speicher ist in gleich grosse Page Frames (oder Frames) unterteilt

Einführung in die technische Informatik

44

Speicherverwaltung

Automation Systems Group



Einführung in die technische Informatik

45

Speicherverwaltung

Automation Systems Group

- Pages haben eine Zweierpotenz Größe
- Grund
 - einfache Berechnung der Page Nummer
 - virtuelle Adresse zerfällt in Page Nummer und Offset
 - Page Nummer einfach aus Adresse ablesen

Einführung in die technische Informatik

46

Speicherverwaltung

Automation Systems Group

- Beispiel
 - 16-bit virtuelle Adresse, Page ist $0x100 = 256 (2^8)$ Bytes groß
 - letzten 8 Bits der Adresse sind Offset
 - ersten 8 Bits der Adresse sind Page Nummer

- virtuelle Adresse 0xA37E

| | | | | | | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Page Nummer | | | | | | | | Offset | | | | | | | |

- Page Nummer = 0xA3, Offset = 0x7E

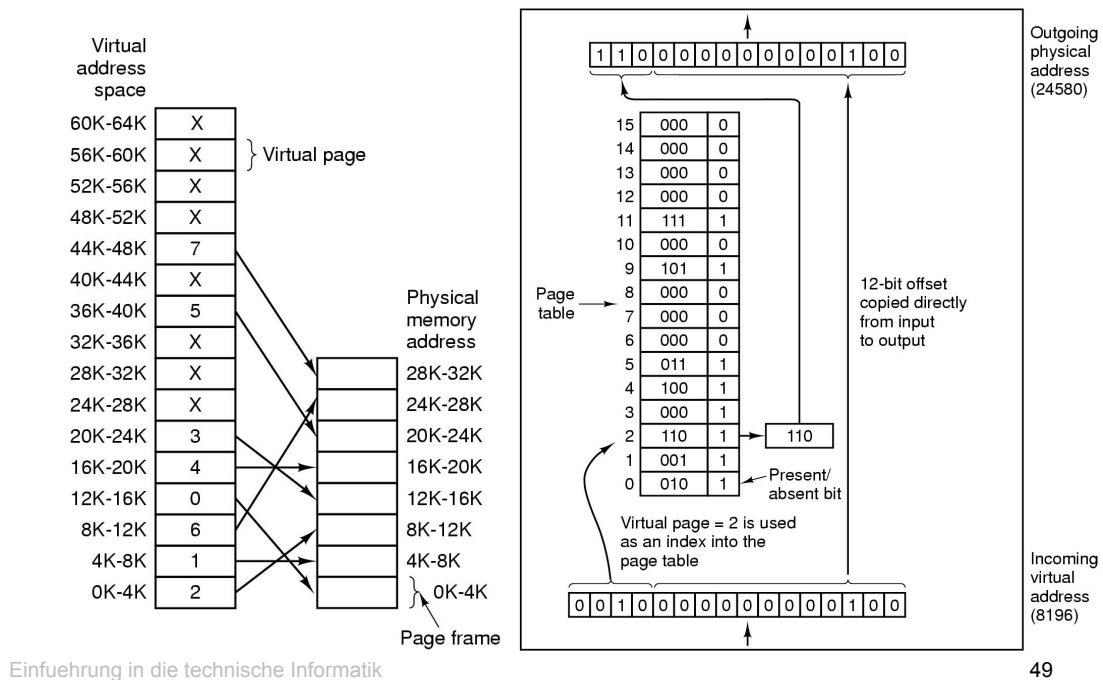
Speicherverwaltung

Automation Systems Group

- Frage
 - Wie bekomme ich physikalische Start-Adresse, wenn ich die virtuelle Start-Adresse habe?
- Page Table
 - virtuelle Start-Adresse wird nicht benötigt
 - Page Nummer reicht aus
 - Page Table speichert für jede Page, wo der entsprechende Page Frame im Speicher liegt
 - außerdem, ein Bit (present bit), welches angibt, ob die Page überhaupt geladen ist
 - falls auf eine Page zugegriffen wird, die nicht im physikalischen Speicher liegt → *page fault*

Speicherverwaltung

Automation Systems Group



49

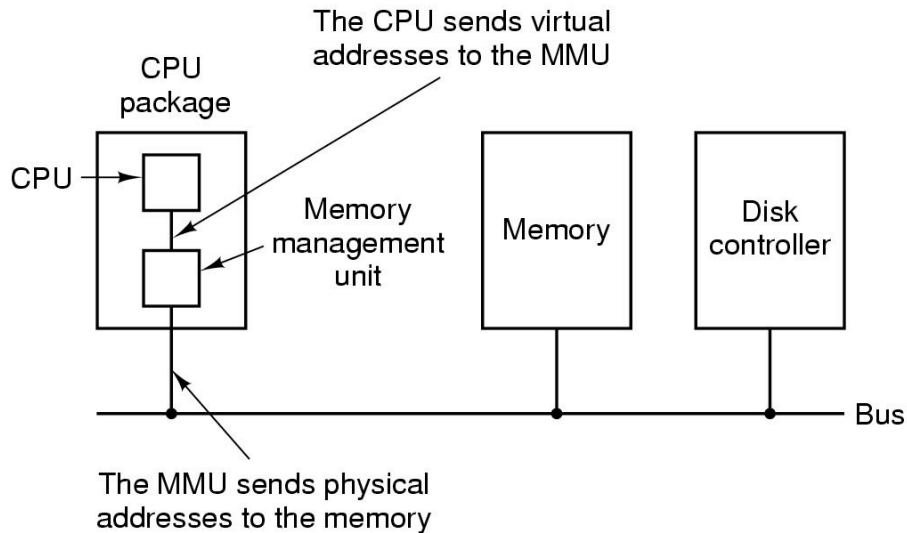
Speicherverwaltung

Automation Systems Group

- Frage
 - Wer macht die Umsetzung?
- **Memory Management Unit (MMU)**
 - Hardware Baustein, der vom Betriebssystem entsprechend geladen wird
 - eine Page Table pro Prozess ist notwendig
 - muss bei jedem Context Switch passend erledigt werden
- Page Tables brauchen einen Eintrag pro Page
 - kann sehr viel werden
 - 32-bit Adressraum mit 4 KByte Pages ergibt 2^{20} Page Table Einträge
 - Teile müssen in den Hauptspeicher ausgelagert werden
 - Page Table Hierarchie

Speicherverwaltung

Automation Systems Group



Einfuehrung in die technische Informatik

51

Speicherverwaltung

Automation Systems Group

- Beispiel
 - Umsetzen einer virtuellen Adresse in die entsprechende physikalische Adresse
 - gegeben ist Page Table, Page Größe, und virtuelle Adresse
 - gefragt ist die physikalische Adresse

Einfuehrung in die technische Informatik

52

Speicherverwaltung

Automation Systems Group

- Angabe
 - Page Größe ist 4 KBytes, virtuelle Adressen haben 32-bit, physikalische Adressen haben 24-bit

| Page Nummer | Frame Nummer |
|-------------|---------------|
| 0x00 | 0x7C |
| 0x01 | 0x8A |
| 0x02 | (not present) |

virtuelle Adressen a) 0x000001BCD
 b) 0x000002FFE

Speicherverwaltung

Automation Systems Group

- 1. Schritt

Zerlegen der virtuellen Adresse in Page Nummer und Offset
4 KByte Pages bedeutet 12 Bits Offset (weil $4096 = 2^{12}$),
daher ist die Page Nummer 20 Bit groß

a) 0x 00 00 1B CD

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| Page Nummer | | | | | | | | | | | | | | | | Offset | | | | | | | | | | | | | | | | |

Page Nummer = 0x01
Offset = 0xBCD

Speicherverwaltung

Automation Systems Group

- 2. Schritt

Page Nummer = 0x01

Offset = 0xBCD

Nachschlagen der entsprechenden Frame Nummer in der Page Table

| Page Nummer | Frame Nummer |
|-------------|---------------|
| 0x00 | 0x7C |
| 0x01 | 0x8A |
| 0x02 | (not present) |

Frame Nummer = 0x8A

Speicherverwaltung

Automation Systems Group

- 3. Schritt

Page Nummer = 0x01

Offset = 0xBCD

Frame Nummer = 0x8A

Zusammensetzen von Frame Nummer und Offset (24 Bit)

| | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---|---|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| Frame Nummer | | | | | | | | | | | | Offset | | | | | | | | | | | |

physikalische Adresse = 0x 08 AB CD

Automation Systems Group

Automation Systems Group

Speicherverwaltung

Automation Systems Group

- Was passiert bei einem *page fault*?
 - Betriebssystem lädt Page in ein freies Frame im Speicher
 - passt Page Table entsprechend an
 - setzt danach Anwendung fort
- Problem
 - was passiert, wenn alle Frames belegt sind
 - ein existierendes Frame muss überschrieben werden
 - veränderte Frames müssten zurückgeschrieben werden
 - daher, besser unmodifizierte Frames nehmen
- Entscheidung
 - page replacement algorithm

Einführung in die technische Informatik

59

Speicherverwaltung

Automation Systems Group

- Page replacement algorithm
 - Ziel
 - Minimiere die Anzahl der page faults
 - Optimal
 - ersetze die Page, die am weitesten in der Zukunft gebraucht wird
 - unmöglich, aber gut fuer Vergleiche
 - FIFO
 - first-in, first-out
 - ersetze älteste Page
 - LRU
 - least-recently-used
 - ersetze die am längsten nicht gebrauchte Page

Einführung in die technische Informatik

60

Speicherverwaltung

Automation Systems Group

- Beispiel
 - gegeben ist die Anzahl der Frames, die benutzt werden können, sowie eine Reihenfolge von Zugriffen (*reference string*) auf die Pages
 - gefragt sind die Entscheidungen des page replacement algorithm, d.h., welche Pages befinden sich nach jedem Zugriff im Speicher (in den Frames)
- Angabe
 - zu verwenden ist LRU
 - 4 Frames sind verfügbar
 - reference string
0 2 1 3 5 4 6 3 7 4 7 3 3 5 5 3 1

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| |
|--|
| |
| |
| |
| |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 0 |
| |
| |
| |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 2 |
| 0 |
| |
| |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 1 |
| 2 |
| 0 |
| |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 3 |
| 1 |
| 2 |
| 0 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 5 |
| 3 |
| 1 |
| 2 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 4 |
| 5 |
| 3 |
| 1 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 6 |
| 4 |
| 5 |
| 3 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| |
|---|
| 3 |
| 6 |
| 4 |
| 5 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 7 |
| 3 |
| 6 |
| 4 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| |
|---|
| 4 |
| 7 |
| 3 |
| 6 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| |
|---|
| 7 |
| 4 |
| 3 |
| 6 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| |
|---|
| 3 |
| 7 |
| 4 |
| 6 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| |
|---|
| 3 |
| 7 |
| 4 |
| 6 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 5 |
| 3 |
| 7 |
| 4 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| |
|---|
| 5 |
| 3 |
| 7 |
| 4 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| |
|---|
| 3 |
| 5 |
| 7 |
| 4 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Speicherverwaltung

Automation Systems Group

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 5 | 4 | 6 | 3 | 7 | 4 | 7 | 3 | 3 | 5 | 5 | 3 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

P

| |
|---|
| 1 |
| 3 |
| 5 |
| 7 |

Speicher

Pages sind nach der Reihenfolge des letzten Zugriffs geordnet;
das am längsten nicht benutzte Frame ist ganz unten

Zusammenfassung

Automation Systems Group

- Betriebssysteme verwalten
 - Prozesse
 - Speicher
 - Dateisystem
 - Eingabe und Ausgabe
- Prozessmanagement
 - Prozessmanagement Datenstrukturen
 - Zustandsübergänge
 - Prozess
 - Ressourcen und thread(s) of execution
 - Thread
 - Registersatz und thread-spezifischen Daten (lokale Variable, Stack)
 - Scheduling

Zusammenfassung

Automation Systems Group

- Speichermanagement
 - Speicherhierarchie
 - Caches
 - Typen von Caches
 - Hauptspeicher
 - virtuelle Speicherverwaltung
 - virtuelle Speicherverwaltung
 - Segmentation
 - Paging
 - Adressumrechnungen
 - page faults
 - page replace algorithms