5 Virtueller Speicher

5.1 Grundsätzliche Vorgehensweise

5.2 Seitenersetzung

5.3 Arbeitsmengenmodell

5.4 Der Ersetzungsalgorithmus wsclock

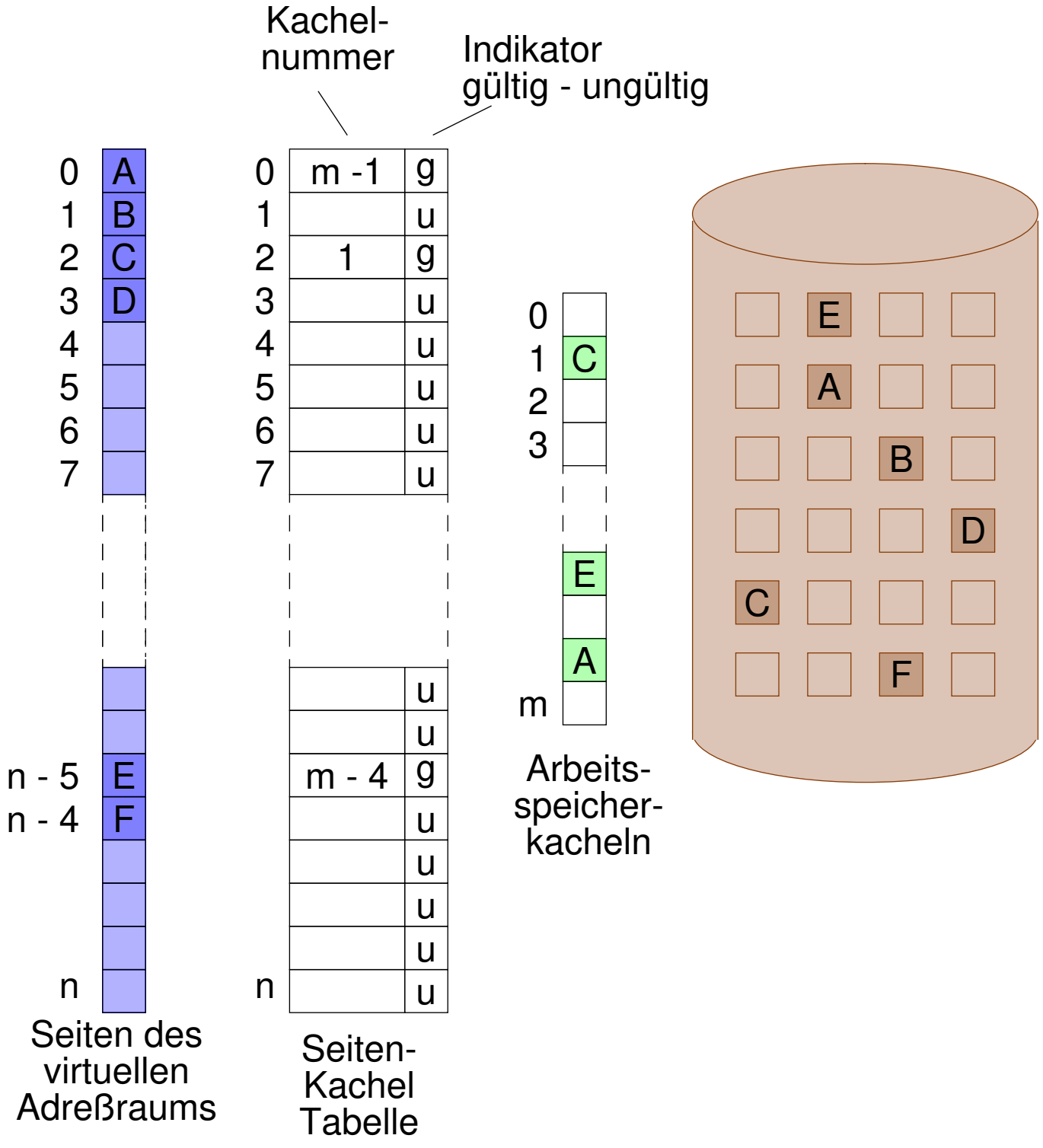
sp1/96/folien/3: 18.12.1996

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

5 Virtueller Speicher

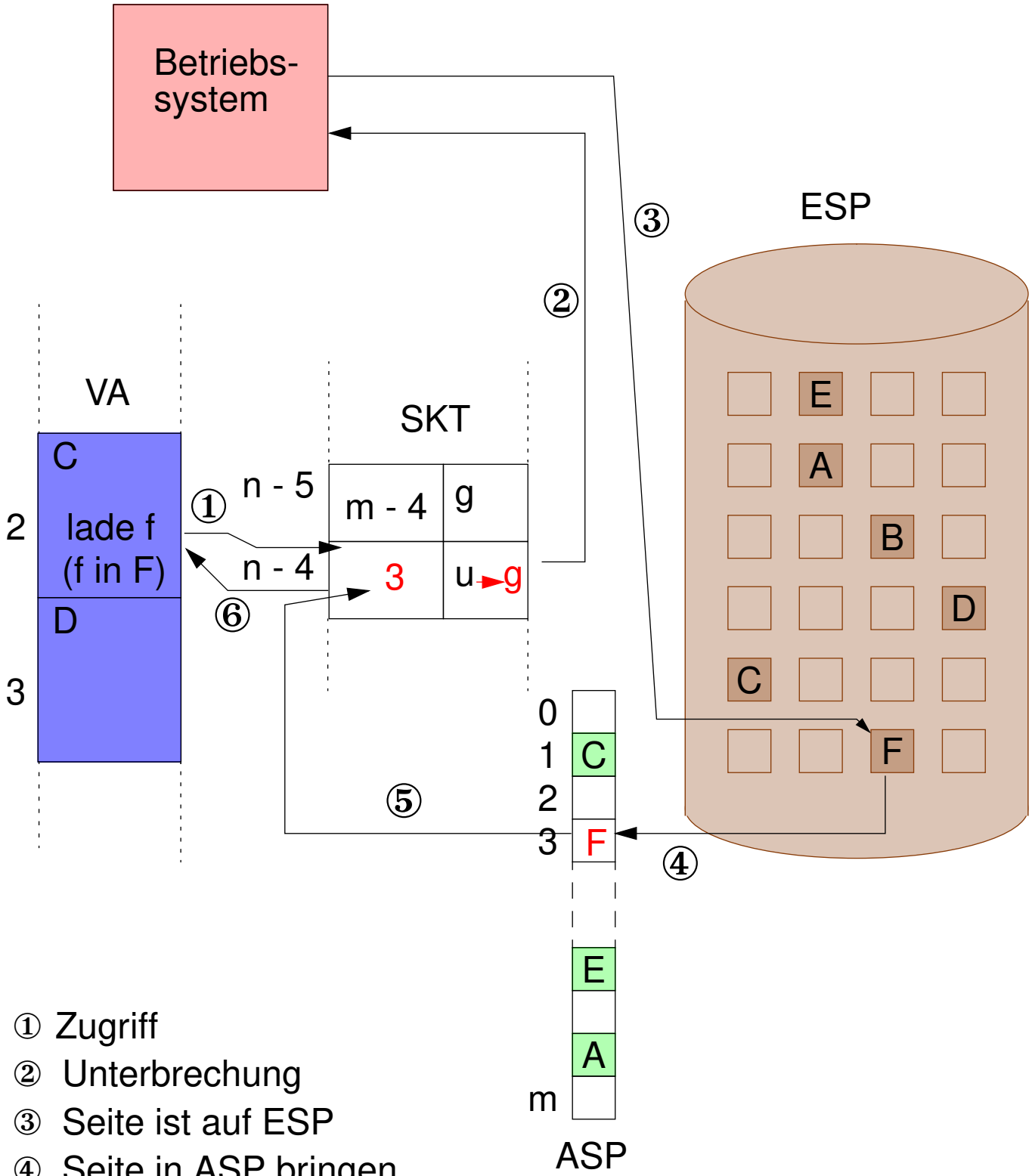
5.1 Grundsätzliche Vorgehensweise

- Gestreute Speicherung in Primär- und Sekundärspeicher



Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

❑ Reaktion auf “Seitenfehler”



① Zugriff
② Unterbrechung
③ Seite ist auf ESP
④ Seite in ASP bringen
⑤ SKT modifizieren
⑥ Befehlsausführung wiederholen

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

5.2 Seitenersetzung

- Bezeichnungen
  - Menge der Nummern von Seiten, die ein Prozeß ansprechen kann:

    $N = \{1, 2, ..., n\}$

  - Menge der Nummern der Kacheln, die den Arbeitsspeicher ausmachen:

    $M = \{1, 2, ..., m\}$

  - Zugriffszeitpunkte

    1, 2, 3, ..., T

  - Seitenreferenzfolge

    $\mathfrak{w} = r_1 ... r_t ... r_T \in N^T$

    (Der i-te Speicherzugriff betrifft die Seite r_i.)

  - Menge der Speicherzustände

    $M_m = \{W \mid W \subseteq N \wedge |W| \leq m\}$

    (Die Elemente von werden benutzt um darzustellen, welche Seiten sich im Arbeitsspeicher befinden.)

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

- Vorwärtsabstand

$$d(x, t) = \min(\{T + 1 - t\} \cup \{i \mid 0 < i \leq T - t \wedge (r_{t+i} = x)\})$$

Zeitintervall bis zum nächsten Zeitpunkt, zu dem die Seite x wieder angesprochen wird.

- Rückwärtsabstand

$$b(x, t) = \min(\{t\} \cup \{i \mid 0 < i < t \wedge (r_{t-i} = x)\})$$

Zeitintervall, seit dem die Seite nicht mehr angesprochen wurde.

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

- Seitenersetzungsstrategien
  - First-In, First-Out: Ersetzt wird die älteste Seite

Zugriff auf Seite		1	2	3	4	1	2	5	1	2	3	4	5
Speicher-zustand	Kachel 1	**1**	1	1	**4**	4	4	**5**	5	5	5	5	5
	Kachel 2		**2**	2	2	**1**	1	1	1	**3**	3	3	
	Kachel 3			**3**	3	3	**2**	2	2	2	2	**4**	4
Kontroll-zustand	y1	λ	λ	1	2	3	4	1	1	1	2	5	5
	y2	λ	1	2	3	4	1	2	2	2	5	3	3
	y3	1	2	3	4	1	2	5	5	5	3	4	4

9 Einlagerungen

Zugriff auf Seite		1	2	3	4	1	2	5	1	2	3	4	5
Speicher-zustand	Kachel 1	**1**	1	1	1	1	1	**5**	5	5	5	**4**	4
	Kachel 2		**2**	2	2	2	2	2	**1**	1	1	1	**5**
	Kachel 3			**3**	3	3	3	3	3	**2**	2	2	2
	Kachel 4				**4**	4	4	4	4	4	**3**	3	3
Kontroll-zustand	y1	λ	λ	λ	1	1	1	2	3	4	5	1	2
	y2	λ	λ	1	2	2	2	3	4	5	1	2	3
	y3	λ	1	2	3	3	3	4	5	1	2	3	4
	y4	1	2	3	4	4	4	5	1	2	3	4	5

10 Einlagerungen

© 18.12.1996, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

- B_o: Ersetzt wird eine Seite mit größtem Vorwärtsabstand

Zugriff auf Seite		1	2	3	4	1	2	5	1	2	3	4	5
Speicher-zustand	Kachel 1	**1**	1	1	1	1	1	1	1	1	1	1	1
	Kachel 2		**2**	2	2	2	2	2	2	2	**3**	**4**	4
	Kachel 3			**3**	**4**	4	4	**5**	5	5	5	5	5
Kontroll-zustand	y1	λ	λ	3	4	4	4	5	1	2	3	4	5
	y2	λ	2	2	2	1	2	2	5	1	1	1	4
	y3	1	1	1	1	2	1	1	2	5	5	5	1

7 Einlagerungen

Zugriff auf Seite		1	2	3	4	1	2	5	1	2	3	4	5
Speicher-zustand	Kachel 1	**1**	1	1	1	1	1	1	1	1	1	1	1
	Kachel 2		**2**	2	2	2	2	2	2	2	2	2	2
	Kachel 3			**3**	3	3	3	3	3	3	3	**4**	4
	Kachel 4				**4**	4	4	**5**	5	5	5	5	5
Kontroll-zustand	y1	λ	λ	λ	4	4	4	5	1	2	3	4	5
	y2	λ	λ	3	3	3	3	3	5	1	2	2	4
	y3	λ	2	2	2	1	2	2	3	5	1	1	2
	y4	1	1	1	1	2	1	1	2	3	5	5	1

6 Einlagerungen

© 18.12.1996, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

- Least Recently Used: Ersetzt wird eine Seite mit größtem Rückwärtsabstand

Zugriff auf Seite		1	2	3	4	1	2	5	1	2	3	4	5
Speicher-zustand	Kachel 1	**1**	1	1	**4**	4	4	**5**	5	5	**3**	3	3
	Kachel 2		**2**	2	2	**1**	1	1	1	1	1	**4**	4
	Kachel 3			**3**	3	3	**2**	2	2	2	2	2	**5**
Kontroll-zustand	y1	λ	λ	1	2	3	4	1	2	5	1	2	3
	y2	λ	1	2	3	4	1	2	5	1	2	3	4
	y3	1	2	3	4	1	2	5	1	2	3	4	5

10 Einlagerungen

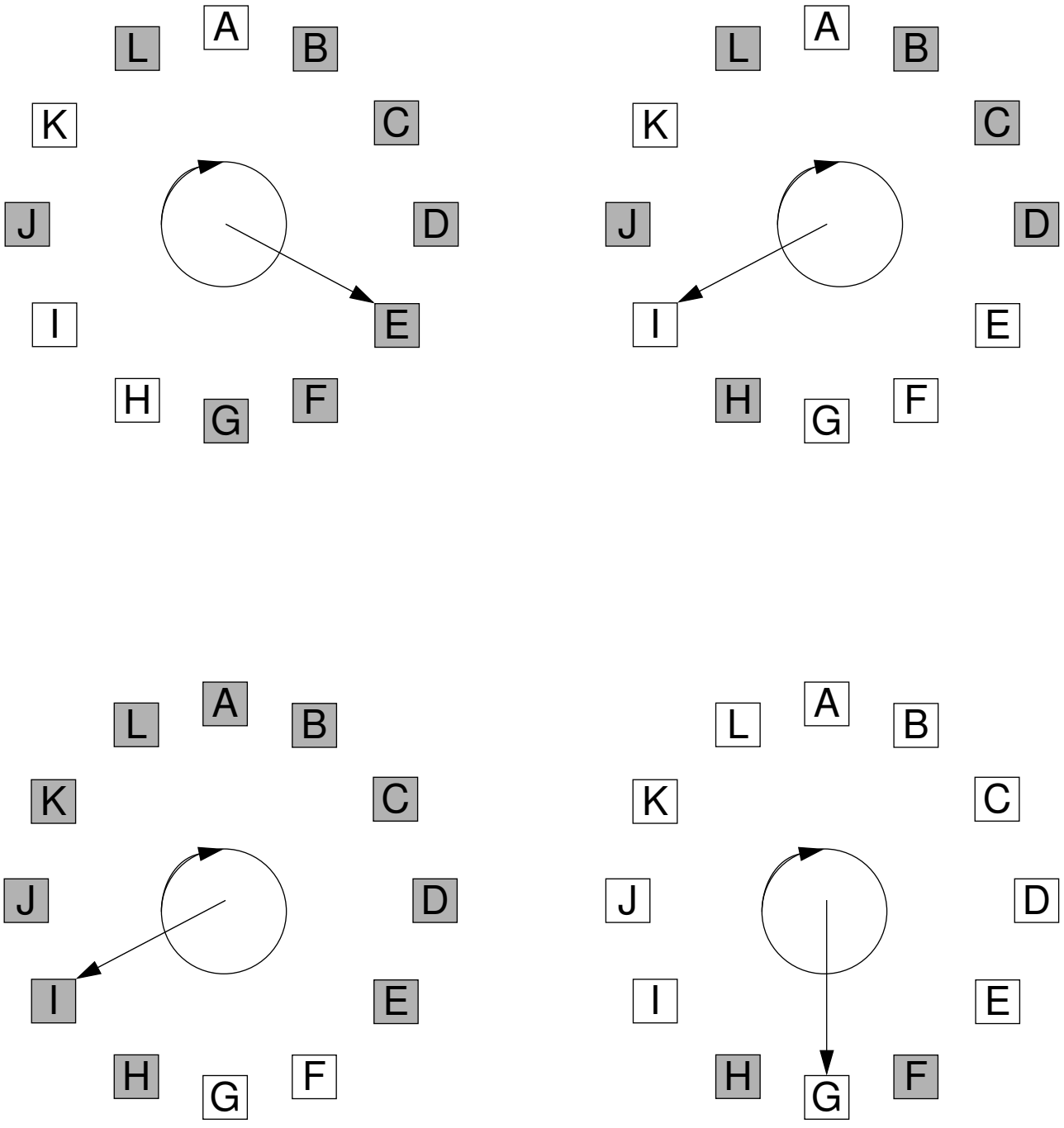
Zugriff auf Seite		1	2	3	4	1	2	5	1	2	3	4	5
Speicher-zustand	Kachel 1	**1**	1	1	1	1	1	1	1	1	1	1	**5**
	Kachel 2		**2**	2	2	2	2	2	2	2	2	2	2
	Kachel 3			**3**	3	3	3	**5**	5	5	5	**4**	4
	Kachel 4				**4**	4	4	4	4	4	**3**	3	3
Kontroll-zustand	y1	λ	λ	λ	1	2	3	4	4	4	5	1	2
	y2	λ	λ	1	2	3	4	1	2	5	1	2	3
	y3	λ	1	2	3	4	1	2	5	1	2	3	4
	y4	1	2	3	4	1	2	5	1	2	3	4	5

8 Einlagerungen

© 18.12.1996, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

- Second Chance



- Referenzindikator gesetzt
- Referenzindikator nicht gesetzt

© 18.12.1996, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

Zugriff auf Seite		1	2	3	4	1	2	5	1	2	3	4	5
Speicher-zustand	Kachel 1	**1**	1	1	**4**	4	4	**5**	5	5	5	5	5
	Kachel 2		**2**	2	2	**1**	1	1	1	**3**	3	3	
	Kachel 3			**3**	3	3	**2**	2	2	2	2	**4**	4
Kontroll-zustand	y1	1	1	**1**	1	1	**1**	1	1	1	0	**0**	1
	y2	**0**	1	1	**0**	1	1	**0**	**1**	**1**	1	1	**1**
	y3	0	**0**	1	0	**0**	1	0	0	1	**0**	1	1

9 Einlagerungen

Zugriff auf Seite		1	2	3	4	1	2	5	1	2	3	4	5
Speicher-zustand	Kachel 1	**1**	1	1	1	1	1	**5**	5	5	5	**4**	4
	Kachel 2		**2**	2	2	2	2	**1**	1	1	1	**5**	
	Kachel 3			**3**	3	3	3	3	**2**	2	2	2	
	Kachel 4				**4**	4	4	4	4	**3**	3	3	
Kontroll-zustand	y1	1	1	1	**1**	**1**	**1**	1	1	1	**1**	1	1
	y2	**0**	1	1	1	1	1	**0**	1	1	1	**0**	1
	y3	0	**0**	1	1	1	1	0	**0**	1	1	0	**0**
	y4	0	0	**0**	1	1	1	0	0	**0**	1	0	0

10 Einlagerungen

© 18.12.1996, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

5.3 Das Arbeitsmengenmodell

- Definitionen
  - Arbeitsmenge

$$W(t, T) = \{i \mid \exists j(\max(1, 1 - t + T) \leq j \leq T \wedge (i = r_{t-T+j}))\}$$

$$= \{i \mid b(i, t+1) \leq T\}$$

  - Arbeitsmengengröße

$$w(t, T) = |W(t, T)|$$

  - mittlere Arbeitsmengengröße

$$s_k(T) = \frac{1}{k} \sum_{t=1}^{k} w(t, T)$$

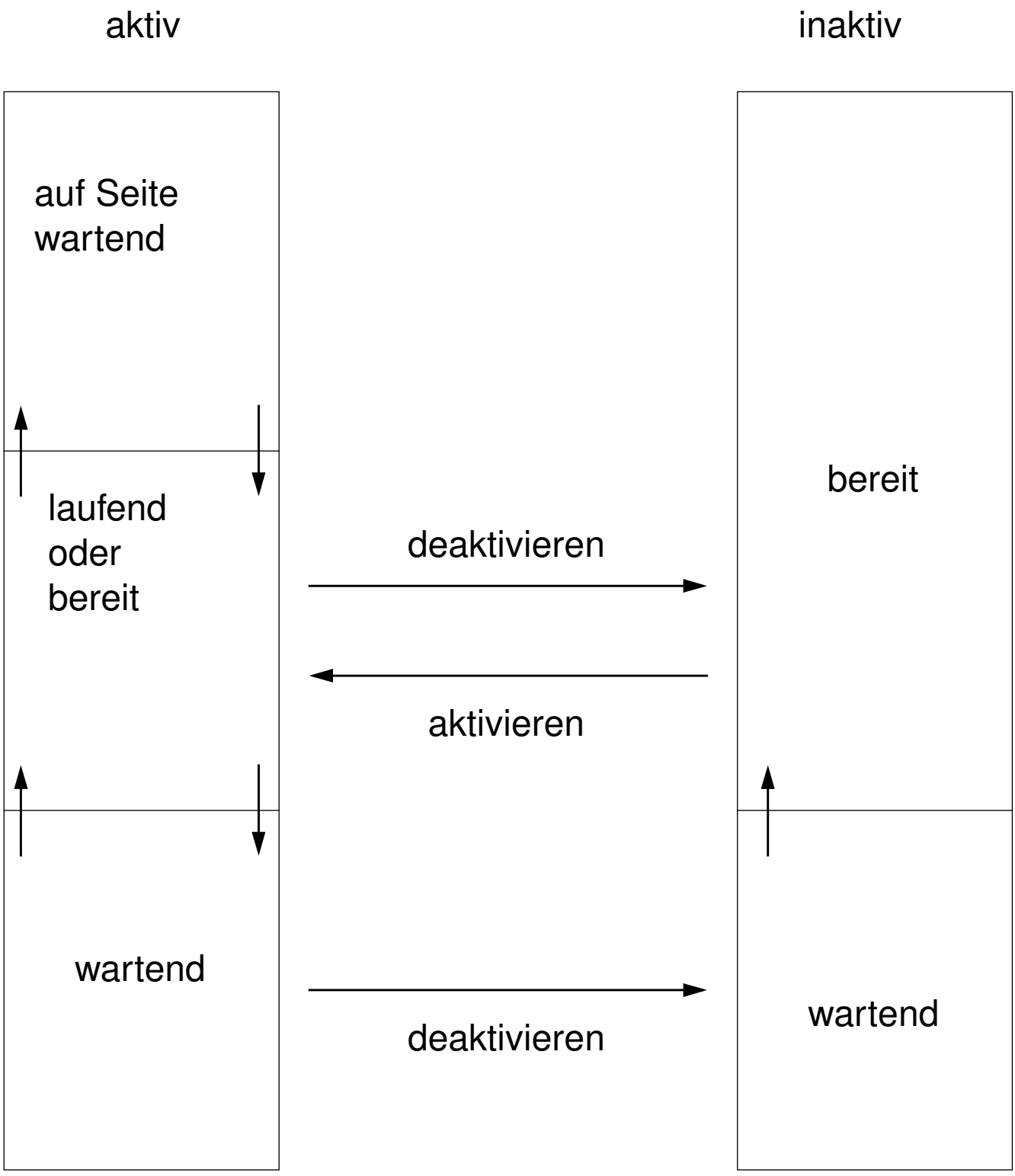
  - Ersetzt werden Seiten, die nicht mehr zur Arbeitsmenge gehören.
  - mittlere Seitenaustauschrate

$$n_k(T) = \frac{1}{k} \left| \{t \mid r_{t+1} \notin W(t, T) \wedge 0 \leq t < k\} \right|$$

© 18.12.1996, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

- Einführung von “Superzuständen” zur Vermeidung von Thrashing

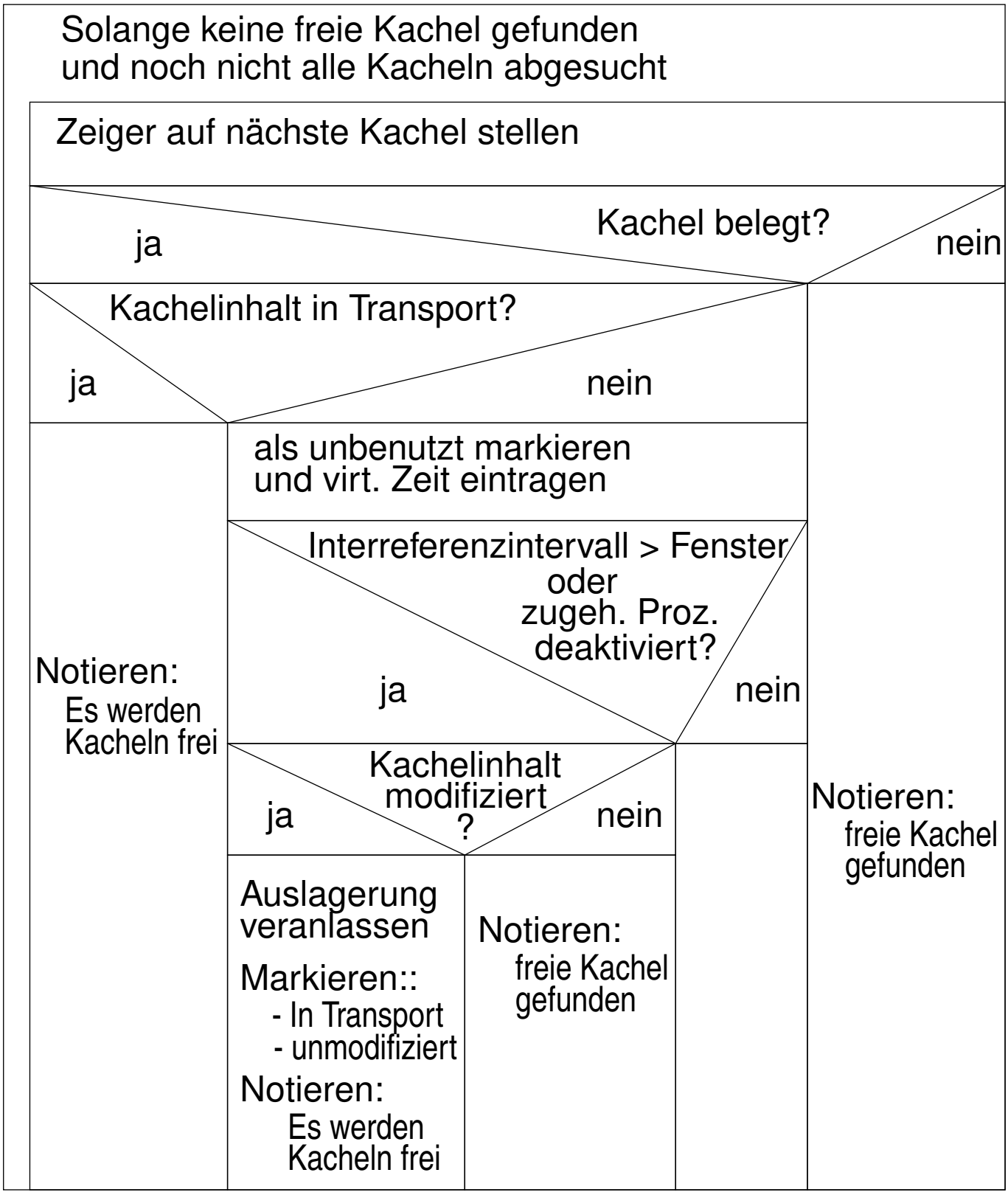


Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

© 18.12.1996, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

5.4 Der Algorithmus wsclock



© 18.12.1996, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

```
class pagedescriptor
  { int modified;
      // Modifikationsindikator
    int used;
      // Referenzindikator
    int present;
      // Praesenzindikator
    .
    .
    .
  public:
    pagedescriptor()
      { modified = FALSE;
        used = FALSE;
        present = FALSE;
      };
    void set_present()
      { present = TRUE; };
    void set_used()
      { used = TRUE; };
    void set_modified()
      { modified = TRUE; };
  friend class framedeskriptor;
  friend class frametable;
  friend class process;
  };
```


Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

```
class process
  { int active;
      // Diese Komponente hat genau dann den Wert TRUE,
      // wenn sich der Prozess im Superzustand aktiv
      // befindet.
    int my_virtual_time;
      // Virtuelle Zeit, die angibt, wieviel CPU-Zeit
      // der Prozess bereits verbraucht hat.
    pagedescriptor pagetable[NUMBER_OF_PAGES];
  public:
    process()
      { active = TRUE;
        my_virtual_time = 0;
      };
    int is_active()
      { return active; };
    void set_active()
      { active = TRUE; };
    void set_inactive()
      { active = FALSE; };
    int is_present(int process_id)
      { return pagetable[process_id].present; };
    void set_present(int process_id)
      { pagetable[process_id].set_present(); };
    void set_used(int process_id)
      { pagetable[process_id].set_used(); };
    void set_modified(int process_id)
      { pagetable[process_id].set_modified(); };
    int get_virtual_time()
      { return my_virtual_time; };
    void increment_virtual_time()
      { my_virtual_time++; };
  friend class frametable;
  };
```


Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

```
class framedeskriptor
  { int process_id;     // Nummer des Prozesses, der
                        // hier eine Seite gelagert hat.
    int pagenumber;     // Nummer der hier gelagerten
                        // Seite.
    int virtual_time;
    int in_transport;
    int in_use;
  public:
    void initialize(int f_process_id,
                    int f_pagenumber,
                    int f_vt)
      { process_id = f_process_id;
        pagenumber = f_pagenumber;
        virtual_time = f_vt;
        in_transport = FALSE;
        in_use = TRUE;
      };

  friend class frametable;
  };
```


Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

```
class frametable
  { framedeskriptor frame[NUMBER_OF_FRAMES];
  public:
    void initialize(int i, framedeskriptor &newframe)
     { frame[i] = newframe;
     };
    int is_in_use(int i)
     { return frame[i].in_use; };
    void set_unused(int i)
     { processtable[frame[i].process_id].
         pagetable[frame[i].pagenumber].used = FALSE;
     };
    int frame_is_used(int i)
     { return processtable[frame[i].process_id].
                 pagetable[frame[i].pagenumber].used; };
    int in_transport(int i)
     { return frame[i].in_transport; };
    void set_in_transport(int i)
     { frame[i].in_transport = TRUE; };
    void reset_in_transport(int i)
     { frame[i].in_transport = FALSE;
       frame[i].in_use = FALSE;
     };
    void update_virtual_time(int i)
     { frame[i].virtual_time
          = processtable
              [frame[i].process_id].my_virtual_time; };
    int unreferenced_virtual_interval(int i)
     { return processtable
                 [frame[i].process_id].my_virtual_time
              - frame[i].virtual_time;
     };
    int active(int i)
     {return processtable
              [frame[i].process_id].active;
     };
```


Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

```
    int modified(int i)
     {return processtable[frame[i].process_id].
            pagetable[frame[i].pagenumber].modified;};
    void set_absent(int i)
     { processtable[frame[i].process_id].
              pagetable[frame[i].pagenumber].present
         = FALSE; };
    void set_empty(int i)
       { frame[i].in_use = FALSE; };
    void set_unmodified(int i)
       { processtable[frame[i].process_id].
              pagetable[frame[i].pagenumber].modified
          = FALSE; };
    int frametable::
          determine_free_frame(int *framenumber);
} GFT;    // global_frametable
  // Die Kacheltabelle enthaelt zu jeder
  // der m Arbeitsspeicherkacheln
  // - einen Verweis pagenumber auf das Seiten-
  //   register, das auf diese Kachel verweist,
  //   bzw. NIL, wenn die Kachel nicht belegt ist;
  // - in process-id die Identifikation des
  //   Prozesses, fuer den eine Seite in dieser
  //   Kachel gespeichert ist, bzw. NIL, wenn die
  //   Kachel nicht belegt ist;
  // - in in_transport die Angabe, ob fuer diese
  //   Kachel ein Ein- oder Austransfer laeuft;
  //   diese Groesse wird durch das Betriebssystem
  //   nach Abschluss des Seitentransfers wieder
  //   auf FALSE gesetzt;
  // - in virtual_time den virtuellen Zeitpunkt,
  //   zu dem der zugehoerige Prozess die Ein-
  //   lagerung veranlasst hat bzw. zuletzt ueber-
  //   prueft wurde, ob der Kachelinhalt aus-
  //   gelagert werden kann.
  // - in in_use die Angabe, ob die Kachel in Benut-
  //   zung ist, d. h. eine Seite enthaelt oder ein
  //   die Kachel betreffender Transfer laeuft.
```


Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

```
int searchpointer;
    // Suchzeiger, der festhaelt, welche Kachel
    // zuletzt darauf ueberprueft wurde, ob sie
    //  als Auslagerungskandidat in Frage kommt.

process processtable[NUMBER_OF_PROCESSES];

int number_of_active_processes;

void start_transfer_to_disk(int k)
{
   . . .
}
```


Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

```
int frametable::determine_free_frame(int *framenumber)
{ int frame_available, frame_found;
  int number_of_inspected_frames;
  frame_available = FALSE;
  frame_found = FALSE;
  number_of_inspected_frames = 0;
  do {
    searchpointer = (searchpointer+1)%NUMBER_OF_FRAMES;
    number_of_inspected_frames++;
    if (!GFT.is_in_use(searchpointer)) {
      frame_found = TRUE;
    } else {
      if (!GFT.in_transport(searchpointer)) {
        if (GFT.frame_is_used(searchpointer)) {
          GFT.update_virtual_time(searchpointer);
          GFT.set_unused(searchpointer);
        }
        if (GFT.unreferenced_virtual_interval
                                        (searchpointer)
              > WINDOW
            || !GFT.active(searchpointer)) {
          if (!GFT.modified(searchpointer)) {
            frame_found = TRUE;
            GFT.set_absent(searchpointer);
            GFT.set_empty(searchpointer);
          } else {
            start_transfer_to_disk(searchpointer);
            frame_available = TRUE;
            GFT.set_in_transport(searchpointer);
            GFT.set_unmodified(searchpointer);
          }
        }
      } else {
        frame_available = TRUE;
      }
    }
  } while (!frame_found
            && number_of_inspected_frames
                                != NUMBER_OF_FRAMES);
```


Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig

```
  if (frame_found) {
    *framenumber = searchpointer;
    return(1);
  } else {
    if (frame_available) {
      *framenumber = -1;
      return(0);
    } else {
      *framenumber = -1;
      return(-1);
    }
  }
}
```


© 18.12.1996, Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors nicht zulässig