

Wiederholung

- 1) Gegeben ist ein Computer System mit 32-Bit Adressen und 3-Level Paging. Wenn a, b und c die Größe der Felder für die entsprechenden Page-Table Einträge sind und d die Bit Länge des Offsets, wie viele Seiten kann man damit maximal adressieren?
- 2) A certain computer provides its users with a virtual-memory space of 2^{32} bytes. The virtual memory is implemented by paging and the page size is 4096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location. Please specify the used table(s) index values. Use single level and two-level paging.
- 3) Ermitteln Sie die Anzahl an Page Faults mit dem Second Chance Algorithmus für die Page Zugriffe für eine Speicher mit 4 Frames:
A B C D C D E B A A C D B

Bitte den Zeitpunkt wo eine Seite eine „Second Chance,“ bekommt zeigen.
- 4) Wie viele Prozesse werden generiert?

```
for i = 1 to 5
{
    pid = fork();
    if !pid
    {
        fork();
        exit();
    }
}
```

- 5) In einer Waschanlage mit einer Waschstraße gibt es 10 Warteplätze. Ein Auto fährt weiter, wenn kein Warteplatz verfügbar ist. (Wichtig: Es wartet nicht).

Wenn kein Auto da ist wartet die Waschanlage auf Autos (explizit). Die Autos warten auf die Waschanlage dass sie frei wird.

Die folgende Initialisierung ist vorgegeben:

```
int WP = 10           //10 Warteplätze vor der Waschstraße, nicht mehr
semaphore A = 0       //Auto (beim Waschen), Anlage wartet dass ein Auto kommt
semaphore WA = 0      //Waschanlage, nur ein Auto kann gewaschen werden
mutex M = 1           //Mutex
```

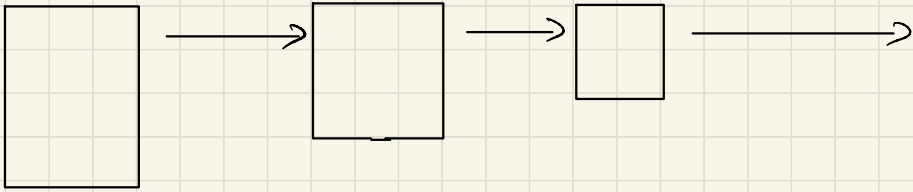
Vervollständigen Sie mit Pseudocode die Prozedur, die die korrekte Arbeit von der Waschanlage beschreibt.

Vervollständigen Sie mit Pseudocode die Prozedur, die das korrekte Verhalten von einem Auto beschreibt.

Die Semaphore und den Mutex kann man mit P() (oder wait()) und V() (oder signal()) Aufrufe bearbeiten.

```
Waschanlage()
{
    while(true)
    {
        //hier vervollständigen
    }
}
Auto ()
{
    while(true)
    {
        //hier vervollständigen
    }
}
```

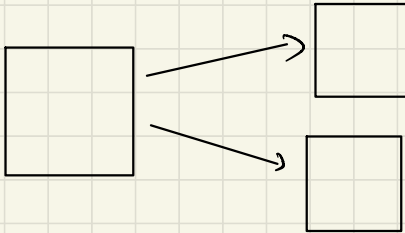
- 1) Gegeben ist ein Computer System mit 32-Bit Adressen und 3-Level Paging. Wenn a, b und c die Größe der Felder für die entsprechenden Page-Table Einträge sind und d die Bit Länge des Offsets, wie viele Seiten kann man damit maximal adressieren?



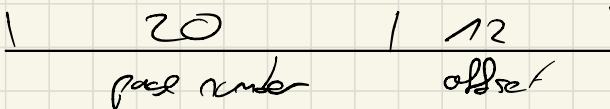
$$2^a \cdot 2^b \cdot 2^c = 2^{a+b+c} = 2^{32-d}$$

- 2) A certain computer provides its users with a virtual-memory space of 2^{32} bytes. The virtual memory is implemented by paging and the page size is 4096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location. Please specify the used table(s) index values. Use single level and two-level paging.

Single level paging

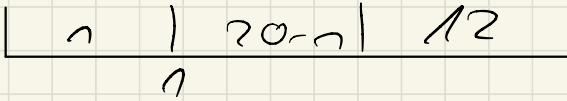
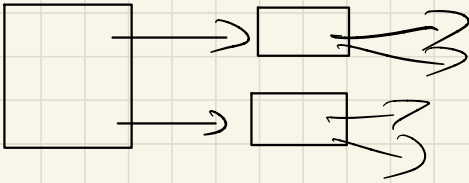


$$4096 = 4 \cdot 1024 = 4 \cdot 2^{10} = 2^{12}$$



1 1 1 2 3 4 5 6 \Rightarrow Table index: 0x11123

0001 0010 0011 0010 0011 | 0100 0101 0110



$n \in [1; 20]$

eg.: $n = 10$

$\Rightarrow 11123456$

0001 0001 0001 0010 0011 | 0100 0101 0110

0x094

0x123

Index für outer
case table

Index für inner
case table

- 3) Ermitteln Sie die Anzahl an Page Faults mit dem Second Chance Algorithmus für die Page Zugriffe für eine Speicher mit 4 Frames:
A B C D C D E B A C D B

Bitte der Zeitpunkt wo eine Seite eine „Second Chance“, bekommt zeigen.

Second Chance
bekommen

	A	B	C	D	C	D	E	B	A	C	D	B
Frame 0	A ₁	A ₁	A ₁	A ₁	A ₁	A ₁	E ₁	E ₁	E ₁	E ₁	E ₁	B ₁
Frame 1		B ₁	B ₁	B ₁	B ₁	B ₁	B ₁	B ₁	B ₁	B ₁	B ₁	D ₁
Frame 2			C ₁	C ₁	C ₁	C ₁	C ₁	C ₁	A ₁	A ₁	A ₁	A ₁
Frame 3				D ₁	D ₁	D ₁	D ₁	D ₁	D ₁	D ₁	C ₁	C ₁
Page Fault	X	X	X	X			X	X		X	X	X

9

- 4) Wie viele Prozesse werden generiert?

```
for i = 1 to 5
{
    pid = fork();
    if !pid
    {
        fork();
        exit();
    }
}
```

$$2 \cdot 5 = 10$$

(+ 1 Anfangsprozess)

- 5) In einer Waschanlage mit einer Waschstraße gibt es 10 Warteplätze. Ein Auto fährt weiter, wenn kein Warteplatz verfügbar ist. (Wichtig: Es wartet nicht).
Wenn kein Auto da ist wartet die Waschanlage auf Autos (explizit). Die Autos warten auf die Waschanlage dass sie frei wird.

Die folgende Initialisierung ist vorgegeben:

```
int WP = 10      //10 Warteplätze vor der Waschstraße, nicht mehr
semaphore A = 0  //Auto (beim Waschen), Anlage wartet dass ein Auto kommt
semaphore WA = 0 //Waschanlage, nur ein Auto kann gewaschen werden
mutex M = 1      //Mutex
```

Vervollständigen Sie mit Pseudocode die Prozedur, die die korrekte Arbeit von der Waschanlage beschreibt.

Vervollständigen Sie mit Pseudocode die Prozedur, die das korrekte Verhalten von einem Auto beschreibt.

Die Semaphore und den Mutex kann man mit P() (oder wait()) und V() (oder signal()) Aufrufe bearbeiten.

Waschanlage ()
{
 while (true)

 {
 P(A)
 P(M)
 WP++
 V(M)
 V(WA)

 }
}

Auto ()

{
 while (true)

 {
 P(M)
 if (WP)
 WP--
 V(M)
 V(A)
 else
 V(M)

 }
}

}