



## Aufgabe 1:

### Aufgabe 1.1: Grundlagen von Betriebssystemen

- a)

Betriebsmittelverwalter: Betriebsmittel (Zeit und Speicher) von Prozessen werden verwaltet, Probleme und Konflikte sollen behandelt werden.

Virtuelle Maschine: Hardwaredetails werden vor dem Benutzer verborgen. Dies sorgt für mehr hardwareunabhängigkeit.

- b)

Sicht 1:

(1) Zuordnung von Betriebsmitteln zu einzelnen Prozessen

(2) Behandlung von Problemen und Konflikten bei Inanspruchnahme von Betriebsmitteln, u.a. Behandlung von Verklemmungs Verklemmungs- und Engpass Engpass-Situationen

Sicht 2:

(1) Zugrunde liegende Rechnerarchitektur vor dem Nutzer verbergen

(2) Details des Mehrbenutzerbetriebs verstecken

- c)

monolithisch

schnellerer Zugriff

unübersichtlich unstrukturiert

hierarchisch

strukturierter

sicherer

nicht so gute performance

- d)

Netzwerk hohe Verteilung-Transparenz

Verteilt keine/geringe Verteilungs-Transparenz



## Aufgabe 1.2: Prozesse und Threads

- a)

Ein Programm ist eine Folge von Anweisungen und hat eine bestimmte Ziel-Pragmatik. Oftmals in einer bestimmten Programmiersprache geschrieben und in für einen Computer verständliche Befehle übersetzt.

Als Prozess wird ein Programm bezeichnet, das aktuell auf dem Betriebssystem ausgeführt wird. Besitzt einen eigenen Prozessadressraum der u.a. Programmcode, Konstanten und prozessspezifische Variablen enthält.

Die eigentliche Ausführung eines Prozess wird in kleineren Mini-Prozesse, s.g. Threads statt. Alle Threads eines Prozess teilen sich den gleichen Prozessadressraum, besitzen jedoch wegen ihrer Nebenläufigkeit einen eigenen Stack.

- b)

Zugriff auf selbe Betriebsmittel können zu komplikationen führen.

=> Deadlock

- c)

z.B Benutzeroberfläche aktualisieren

können auf den selben Speicher zugreifen und so einfacher kommunizieren

Ressourcen sparender

- d)

X:= BEREIT: Der Prozess besitzt alle benötigten Betriebsmittel, wartet jedoch auf die Freigabe eines Prozessorkerns.

Y:= LAUFEND: Der Prozess ist aktuell einem Prozessorkern zugeordnet und läuft auf diesem ab.

Z:= WARTEND: Der Prozess wurde unterbrochen und wartet auf eine nicht-Prozessorkern-Ressource.

a:= (NEW nach X) Starten des Prozess & Laden aller benötigten Betriebsmittel

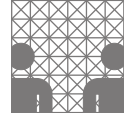
b:= (X nach Y) Zuweisung von Prozessorkern(en)

c:= (Y nach Z) Benötigung eines momentan nicht verfügbaren Betriebsmittels (I/O)

d:= (Z nach X) Zuteilung des Betriebsmittel

e:= (Y nach TERMINATED) Beenden des Prozess, Berechnung ist vollständig

f:= (Y nach X) Pausieren des Prozess



## Aufgabe 1.3: n-Adressmaschine

- a) 2-Adress Maschine: Leseaufträge: 15, Schreibaufträge: 10 Berechnungszeit: 25,5

```
MOVE >a1< >H1<  H1:=a1
ADD  >a2< >H1<  H1:=a2 + H1
MOVE >a3> >H2<  H2:=a3
DIV  >H1< >H2<  H2:=H1 / H2
MOVE >b2< >H3<  H3:=b2
SUB  >b1< >H3<  H3:=b1 - H3
MOVE >b3> >H4<  H4:=b3
DIV  >H3< >H4<  H4:=H3 / H4
ADD  >H2< >H4<  H4:=H2 + H4
MOVE >H4< >R<   R:=H4
```

- b) 1-Adress-Maschine: Leseaufträge: 9, Schreibaufträge: 4 Berechnungszeit: 13,65 Vertauscht!:

LOAD >b2<	AC:=b2	AC=b2
SUB >b1<	AC:=b2 - AC	AC=b1 - b2
DIV >b3<	AC:=AC / b3	AC=(b1 - b2) / b3
SAVE >H1<	H1:=AC	
LOAD >a1<	AC:=a1	AC=a1
ADD >a2<	AC:=AC + a2	AC=a1 + a2
DIV >a3<	AC:=AC / a3	AC=(a1 + a2) / a3
ADD >H1<	AC:=AC + H1	AC=((a1 + a2) / a3) + ((b1 - b2) / b3)
SAVE >R<	R:=AC	

- c) 0-Adress-Maschine: Leseaufträge: 6, Schreibaufträge: 1, Berechnungszeit: 7,6

```
PUSH >a1<    ;a1;
PUSH >a2<    ;a1;a2;
ADD         :a1 + a2;
PUSH >a3<    ;a1 + a2;a3;
DIV         ;(a1 + a2) / a3;
PUSH >b1<    ;(a1 + a2) / a3;b1;
PUSH >b2<    ;(a1 + a2) / a3;b1;b2;
SUB         ;(a1 + a2) / a3;b1 - b2;
PUSH >b3<    ;(a1 + a2) / a3;b1 - b2;b3;
DIV         ;(a1 + a2) / a3;(b1 - b2) / b3;
ADD         ;((a1 + a2) / a3) + ((b1 - b2) / b3);
POP >R<      ;
```