

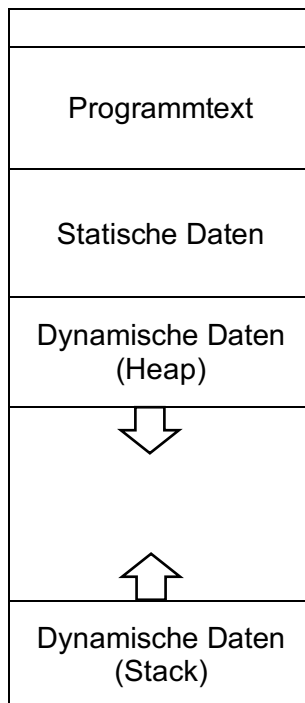
Tutor: Friedrich  
Tutorium: Dienstag 08-10 Uhr  
Raum FH 314

Gruppenmitglieder:  
Julius Basler  
Fabian Philipp Berner  
Jannik Leander Hyun-Ho Novak

### Aufgabe 1.1

a)

0x00000000



← Diese Adresse wird zwecks der NULL Adresse nicht für den Programmtext genutzt

Hohe Adresse

Quelle: 1

b)

- (a) Statische Daten
- (b) Programmtext
- (c) Statische Daten
- (d) Dynamische Daten (Stack)
- (e) Dynamische Daten (Stack)

Quelle: 2

Quellen:

- 1) Kao/Nordholz: Systemprogrammierung SS18; Vorlesungsfolien 1, S.12.,17.
- 2) Kao/Nordholz: Systemprogrammierung SS18; Vorlesungsfolien 1; S.13.

## Aufgabe 1.2

a)

Ein *Interrupt* (Unterbrechung) ist eine in die Hardware integrierte Möglichkeit, um die CPU nach Ausführung eines Befehles sofort in eine neue Prozedur springen zu lassen. Diese Prozedur dient der Untersuchung der Unterbrechung und veranlasst gegebenenfalls erforderliche Aktionen. <sup>1)</sup>

Bsp. 1: DMA

Die CPU initiiert einen direkten Speicherzugriff indem die dementsprechende Steuereinheit angesprochen wird. Diese liest nun eigenständig die Daten aus und informiert die CPU mittels *Interrupt*, dass das Einlesen vollendet wurde. <sup>1)</sup>

Bsp. 2: Druckauftrag

Die CPU initiiert einen Druckauftrag indem die dementsprechende Steuereinheit angesprochen wird. Es werden nun Daten über diese Steuereinheit gesendet, welche zwischendurch per *Interrupt* unterbrochen werden, damit die Einheiten in der Übertragungskette keine Daten verlieren. <sup>2)</sup>

b)

Systemaufrufe dienen der Trennung zwischen dem privilegierten (*system mode*) und unprivilegierten (*user mode*) Zustand der CPU. Dies verhindert, dass Benutzerprogramme im *user mode* bestimmte Instruktionen ausführen können, da diese z.B. im Normalfall nicht auf Daten eines anderen *tasks* zugreifen dürfen. Der *system mode* ist in der Regel für das Betriebssystem, jedoch können auch Benutzerprogramme einen Systemauftrag anfordern. Generell können *user mode* Instruktionen auch durch einen *Interrupt*, bei Fehlern oder durch explizite Instruktionen in den *system mode* gelangen, wobei die expliziten Aufrufe auf Gültigkeit geprüft werden. <sup>1)</sup>

Quellen:

1) Kao/Nordholz: Systemprogrammierung SS18; Vorlesungsfolien 1; S.26.-28,33,19.-20.

2) William Stallings: Operating Systems: Internals and Design Principles, Seventh Edition, Pearson Education Inc., 2012 S.14.