

### Aufgabe 1

Das Betriebssystem hat unter anderem die Aufgabe Prozesse zu verwalten. Prozesse können aber verschiedene Zustände besitzen:

#### Prozesszustände:

##### **bereit (ready):**

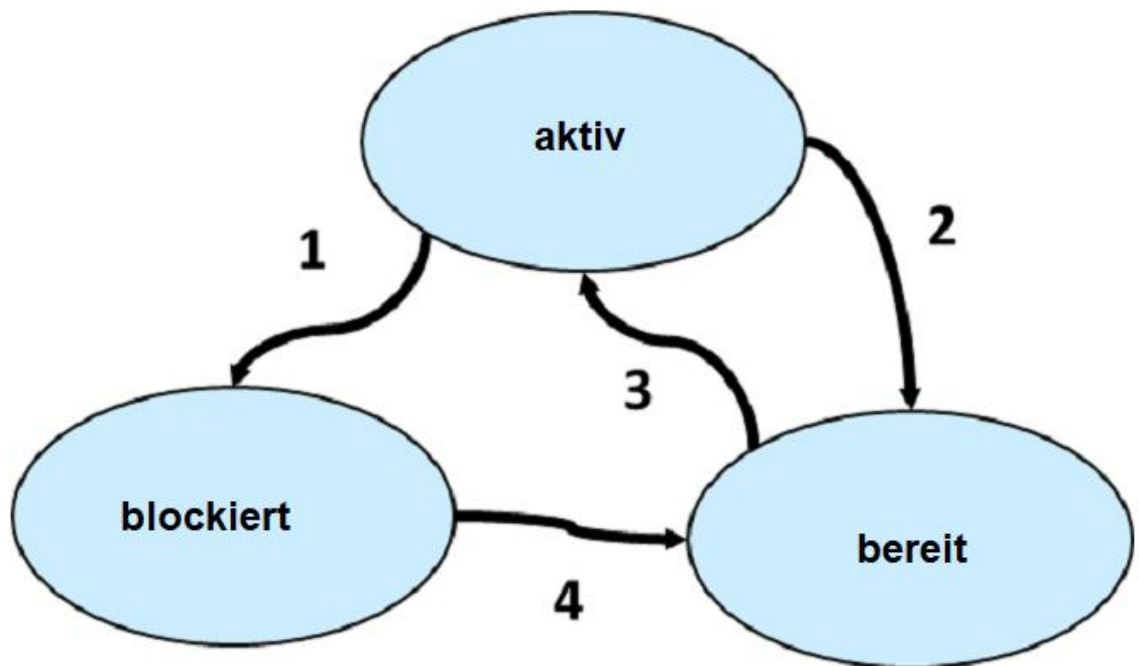
wartet auf die  
Zuteilung des  
Prozessors

##### **aktiv (running):**

besitzt den  
Prozessor und läuft  
ab

##### **blockiert**

**(blocked, asleep):**  
Prozessor wurde  
entzogen, wartet  
auf Betriebsmittel



#### Vier Übergänge sind zwischen diesen Zuständen möglich:

1. Prozessor blockiert: Notwendiges Betriebsmittel steht nicht zur Verfügung.
2. Scheduler entzieht den Prozessor
3. Scheduler teilt Prozessor zu
4. Benötigtes Betriebsmittel steht zur Verfügung

**Brückenkurs Informatik  
Tutorium  
Tag 5****Aufgabe 2**

Unter anderem sorgt das Betriebssystem für die Anbindung benötigte Geräte:

- Was verstehen Sie unter Gerätemanager?
- Welche Aufgaben hat die Geräteverwaltung?

**Aufgaben der Geräteverwaltung:**

- **Initialisierung von E/A Vorgängen (Gerätetreiber):**
  - Definition der Geräteeigenschaften gegenüber dem Betriebssystem
  - Initialisierung des Gerätes beim Systemstart
  - Aktivierung des Gerätes
  - Meldung von Gerätefehlern
  - Reaktion auf Interrupts des Gerätes
  - Umsetzung der E/A-Aufträge in spezifische Geräte-Informationen
  - Übergabe von Daten und Statusinformationen an den Nutzer
- **Überwachung von E/A-Vorgängen**
- **Terminierung von E/A-Vorgängen**
- **Effiziente Zuteilung der E/A-Geräte an Prozesse**
- **Vermeidung von Zugriffskonflikten**

### Aufgabe 3

Auf einer Rechenanlage werden viele Daten dauerhaft gespeichert und müssen entsprechend verwaltet werden:

- Was macht die Dateiverwaltung?
- Welche Operationen muß ein Betriebssystem sicherstellen?

## Dateiverwaltung



Drei Anforderungen an die langfristige Speicherung von Informationen:

1. Es muss möglich sein, sehr große Mengen von Informationen zu speichern.
2. Die Informationen müssen die Terminierung der Prozesse, die diese verwenden, überleben.
3. Es muss für mehrere Prozesse möglich sein, gleichzeitig auf die Informationen zuzugreifen.



- Informationen werden auf Platten und anderen externen Speichermedien in Einheiten, genannt Dateien, abgelegt
- Prozesse können diese lesen und ggf. beschreiben
- Dateien sind logische Betriebsmittel

Einführung in die Informatik, SS 2015  
Mohammad Abuoba

26

### Aufgabe 4

Öffnen Sie IrfanView. Drücken Sie nun die Tastenkombination STRG+ALT+ENTF und öffnen Sie den Task-Manager. Suchen Sie unter Prozesse „IrfanView“.

- Welche Prozesskennung hat die Instanz ihres Programms?
- Weshalb wird eine Prozesskennung benötigt?

Mehrere Prozesse mit selben Namen müssen eindeutig identifizierbar sein

- Führen Sie einen Rechtsklick aus und wählen sie „Task beenden“.
- Szenario: Ihr Windows Explorer hat sich aufgehängt. Wie gehen Sie vor?
- Taskmanager öffnen, Windows-Explorer aufklappen und mittels rechtsklick „Task beenden“.
- Nun „Datei“ – „Neuer Task Ausführen“ öffnen und „explorer.exe“ eingeben – Bestätigen
- Neuer Task ausführen kann jedes Programm oder Verknüpfung, welche im C:\Windows\System32 ohne Angabe der Dateierdung, gestartet werden.

### Aufgabe 5

Das Betriebssystem steuert und überwacht den geregelten Ablauf der Prozesse auf dem Rechner:

a) Definieren Sie den Begriff „Prozess“ in diesem Kontext!

b) Was ist der Unterschied zwischen „Prozess“ und „Programm“?

a)

-Ein Prozess(engl. Task) ist ein in Ausführung befindliches Programm  
-Das BS hat die Aufgabe alle Prozesse auf einen oder mehr Prozessoren abzubilden  
-Ein Betriebssystem erlaubt „Multitasking“, d.h.Im System befinden sich mehrere Prozesse gleichzeitig und werden von einem Prozessor (CPU) bedient. Der Prozessor schaltet dabei zwischen den einzelnen Prozessen hin und her, um die Illusion zu vermitteln, die Prozesse würden real gleichzeitig ablaufen.

-Ein Prozess besteht aus:

- Speicherabbild (coreimage), d.h. den Programmanweisungen und Daten, die im Hauptspeicher liegen,
- der Registerbelegung und
- Verwaltungsinformationen, z.B. der bisher verbrauchten CPU-Zeit

-Bei Verwaltung mehrerer Prozesse entscheidet der Prozess-Schedulerdes Betriebssystems über die Ausführungsreihenfolge und die Zuteilung des Prozessors

-Jeder Prozess besitzt eine Liste mit Betriebsmittelanforderungen sowie eine Liste der bereits zugeteilten Betriebsmittel

b) -Programm ist Handlungsvorschrift

-Prozess ist die Handlung selbst

-In jedem Prozess läuft ein Programm

### Aufgabe 6

Der Scheduler ist eine wichtige Komponente des Betriebssystems, die für einen geregelten Ablauf der Prozesse sorgt:

a) Was verstehen Sie unter Scheduler?

Prozesssteuerungsprogramm, welches die CPU-Ressourcen für Prozesse reglementiert (task-switcher).

# Brückenkurs Informatik

## Tutorium

### Tag 5

b) Welche Kriterien werden beim Scheduling-Verfahren berücksichtigt

**-Durchsatz:** Anzahl der Prozesse pro Zeiteinheit

**-Fairness:** „gerechte“ Zuteilung der Betriebsmittel, d.h. kein Prozess soll ausgehungert werden

**-Prozessorauslastung:** Maximale Auslastung nutzt allen Prozessen, da jeder Prozess den Prozessor benötigt

**-Antwortzeit:** Wichtig bei Dialogsystemen

**-Einhaltung von Fristen:** Notwendig bei Echtzeitsystemen

-Viele Scheduling-Algorithmen wurden entwickelt, um verschiedenen Anforderungen gerecht zu werden

c) Welche Scheduling Strategien kennen Sie? Nennen Sie min. 3 Verfahren und erläutern Sie diese kurz!

Strategien ohne Unterbrechungen	Strategien mit Unterbrechungen
<b>First ComeFirst Serve(FCFS):</b> -Planung nach Ankunftszeit -Prozesse werden in der Reihenfolge ihres Eintreffens bearbeitet -Antwortzeit hängt von der Prozessorauslastung ab, d.h. von der Anzahl der Prozesse	<b>RoundRobin:</b> -Jeder Prozess erhält den Prozessor für eine festgelegte Zeitspanne und wird danach wieder in eine Warteliste einsortiert -Warteschlange kann nach FCFS sortiert sein -Weitverbreitete Strategie -Einfache Implementierung -Problem: sinnvolle Zeitspanne wählen
<b>ShortestJob Next (SJN):</b> -Planung nach Laufzeit -Prozesse mit (geschätzt) kurzer Laufzeit werden bevorzugt -Wartezeit für kurze Prozesse sinkt -Antwortzeit hängt von der Länge des Prozesses und der Prozessorauslastung ab	<b>Planung nach Prioritäten:</b> -Jeder Prozess erhält eine bestimmte Priorität -Prozesse mit hoher Priorität werden bevorzugt -Problem bei Scheduling mit festgesetzten - Prioritäten Prozesse können ausgehungert werden Lösung: Aging-Priorität steigt mit der Wartezeit

## Aufgabe 7

Nennen Sie den Unterschied zwischen einem Compiler und einem Interpreter

**Compiler:** überführt das Programm in eine Folge von Maschinenbefehlen (Maschinencode), die der Prozessor des Rechners dann ausführen kann. (C,C++)

**Interpreter:** zur Laufzeit wird immer nur ein kleiner Ausschnitt betrachtet, analysiert und durch Maschinenbefehle ausgeführt (Python)

**Unterschied:** Ein kompiliertes Programm ist zu Laufzeiten vollständig in Maschinencode übersetzt worden. Während ein interpretiertes Programm zu Laufzeiten aufs Neue in Maschinenbefehle übersetzt wird.

### Aufgabe 8

Erstellen Sie mit dem Windows Aufgabenplaner ein automatisches Backup Ihrer wichtigen Dateien zu einem regelmäßigen Zeitpunkt. Nutzen Sie hierzu ein hinterlegtes Script.

#### 1) Batch-Datei (.bat oder .cmd) erstellen mit folgendem Inhalt (ggf. Powershell):

```
@echo off
```

```
set backup_folder=C:\Backup_Test\backup
```

```
echo ### Backing up "C:\Backup_Test\backup"
```

```
robocopy "C:\Backup_Test\ZuKopierenderOrdner" "%backup_folder%\NeuerOrdner" /MIR
```

```
echo Backup Complete!
```

```
@pause
```

#### 2.) Mit dem Windows-Aufgabenplaner das Script nun automatisieren...

Es kann auch ein Script mit einem anderen Copy Befehl genutzt werden.

Anstelle Aufgabenplaner können bei mac und linux cronjobs genutzt werden

<https://www.bennetrichter.de/anleitungen/cronjobs-linux/>

<https://ole.michelsen.dk/blog/schedule-jobs-with-crontab-on-mac-osx/>

---

Google Engine effektiv nutzen:

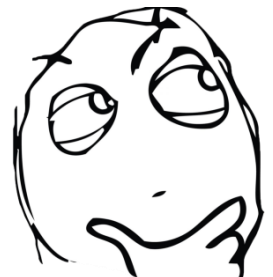
<https://www.lifehack.org/articles/technology/20-tips-use-google-search-efficiently.html>

HTW-Hochschulsport:

<https://hochschulsport.htw-berlin.de/>

Studierendenschaft der HTW:

<https://students-htw.de/>



## Aus Alt Mach Neu

*Softwareentwickler\*innen kommen oft in die Situation, dass sie Programme verbessern müssen, die alt sind, die irgendein\*e Praktikant\*in mal geschrieben hat oder die sie selber geschrieben haben, als sie noch nicht so viel Erfahrung hatten. Ein Programm nehmen und es verbessern heißt „Refactoring“. Dabei wird der Code eleganter gemacht, die Architektur der Software kann verbessert werden, das Programm kann auch ressourcenschonender gemacht werden. Je nachdem, welches Ziel man verfolgt, wird das Resultat anders aussehend – z.B. ist der schnellste Code nicht immer der eleganteste!*



*In der Aufgabe von Tag 3 hast du schon ein bisschen Refactoring gemacht, nämlich als du doppelten Code durch eine Funktion mit Parametern ersetzt hast.*

*In dieser Aufgabe übst du, wie du gegebenen Code eleganter machst. Dabei übst du Programmatic Thinking und fremden Code nachzuvollziehen.*

**1. Remixe das Projekt „Bananenspiel“** (Original von Flox1210):

<https://scratch.mit.edu/projects/220090548/>

**2. Spiele das Spiel!**

**3. Untersuche den Code:** Welche Figuren gibt es? Was machen diese? Welche Events lösen welchen Code aus?

Tipp: Mach dir Notizen bzw. male ein Diagramm

Tipp: Welche verschiedenen Schwierigkeitsstufen soll es geben, und wie unterscheiden sich diese?

**4. Recherchiere im Scratch-Wiki unter <https://de.scratch-wiki.info/wiki/Klonen>, wie die Klonen-Funktion bei Scratch funktioniert.**

**5. Recherchiere im Scratch-Wiki unter <https://de.scratch-wiki.info/wiki/Senden>, wie die Senden-Funktion bei Scratch funktioniert.**

**6. Mach dir einen Plan, was du alles refactoren könntest, um den Code möglichst elegant zu machen.** Also: Wie kannst du doppelten Code vermeiden? Wie kannst du Parameter und Variablen, Conditionals, Klone und Funktionen geschickt nutzen?



**Brückenkurs Informatik**
  
**Tutorium**
  
**Tag 5**

Tipp: Könnte man das Management der Schwierigkeitsstufen vereinfachen? Z.B. mit einer Variable?

Tipp: Schachtle Schleifen und Conditionals geschickt, um doppelten Code zu vermeiden. Wann ist doppelter Code für die Funktionalität des Programmes notwendig?



Hinweis: Bei Scratch wird Code konkurrent ausgeführt, wenn beider Code auf dasselbe Ereignis reagiert. Wenn dabei noch Timer verwendet werden, ist doppelter Code manchmal notwendig.

Beispiel:  
 Hier wird der Conditional, der getestet ob Bananas2 berührt wird, wiederholt, weil der Timer im rechten Skript sonst alles andere blockieren würde, was aber „gleichzeitig“ bzw konkurrent passieren muss. Sonst hakt das Spiel!

Tipp: Gibt es Code der nie ausgeführt wird?



7. **Refactore dein Spiel, Schritt-für-Schritt.** Z.B. Ersetze erst doppelten Code in einer Figur durch eine Funktion. Implementiere dabei keine Funktionen, die das Original-Spiel nicht auch hat/ haben soll.

Tipp: Teste nach jedem Schritt, ob dein Programm funktioniert.

Lösung: <https://scratch.mit.edu/projects/435664085/>

8. **Füge dein fertiges Spiel zur Gruppe unter**  
<https://scratch.mit.edu/studios/27659504/> **hinzu.**

9. **Schau dir refactored Spiele der anderen an und reflektiere.** Was haben sie anders gemacht als du? Was denkst du, ist die elegantere Lösung? Haben die anderen Lösungen auch Vorteile?

10. **Refactore dein Spiel erneut.**

11. Optional: Individualisiere das Spiel, wie es dir gefällt und verbessere die Funktionalität!