

Systemprogrammierung

Jan Fässler

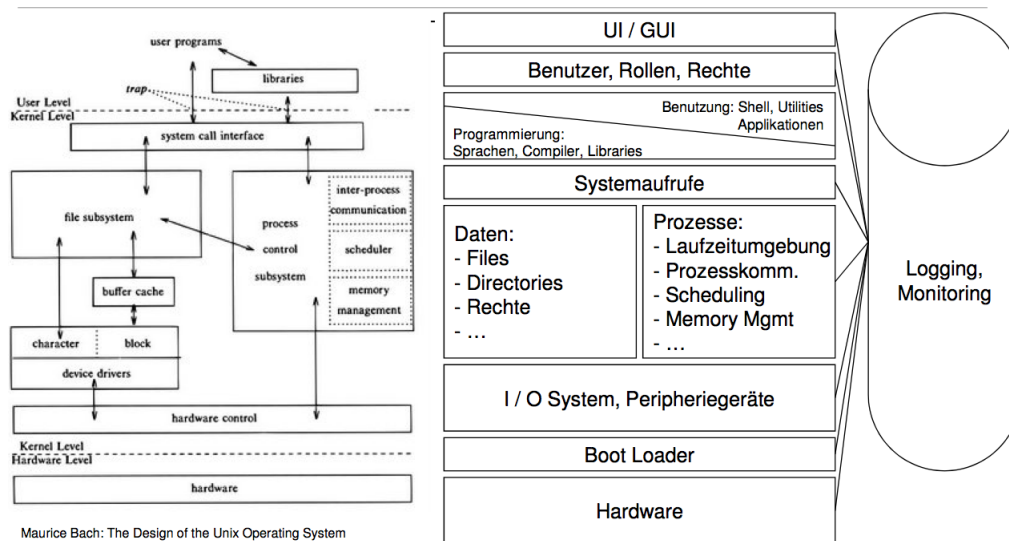
3. Semester (HS 2012)

Inhaltsverzeichnis

1	Unix/Linux	1
1.1	Aufbau	1
1.2	Prozesse	1
1.2.1	Steuerungssystem	1
1.2.2	Aufbau	1
1.2.3	Kernel und User Mode	2
1.2.4	Zustände	2
1.3	Memory	2
1.3.1	Virtual Memory	2
1.3.2	Swapping	3
1.3.3	Paged Memory	3
1.3.4	Fehlerzustände	3
2	System Call Schnittstelle	4
2.1	Prozess-System	4
2.2	Datei-System	4
2.3	Directory Handling	5
2.4	Speicherverwaltung	5
2.5	Weitere	5

1 Unix/Linux

1.1 Aufbau

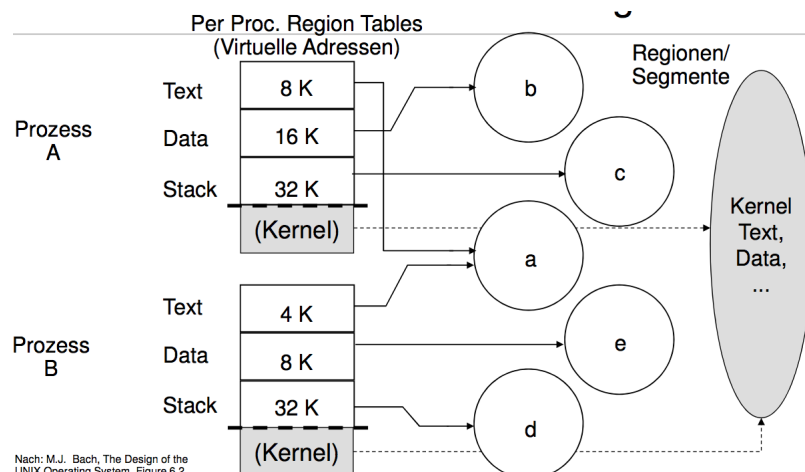


1.2 Prozesse

1.2.1 Steuerungssystem

- Prozesse kreieren & starten
- Prozesse schedulen, Warteschlangen, Ressourcenverbrauch
- Prozesse stoppen / unterbrechen / terminieren
- Prozess-Signalisierung und -kommunikation
- Faire Zuordnung von Hauptspeicher und anderen geteilten Ressourcen
- Ein-/Auslagerung von Prozessen bei vollem Speicher
- Prozesse und ihre Zustände anzeigen

1.2.2 Aufbau



1.2.3 Kernel und User Mode

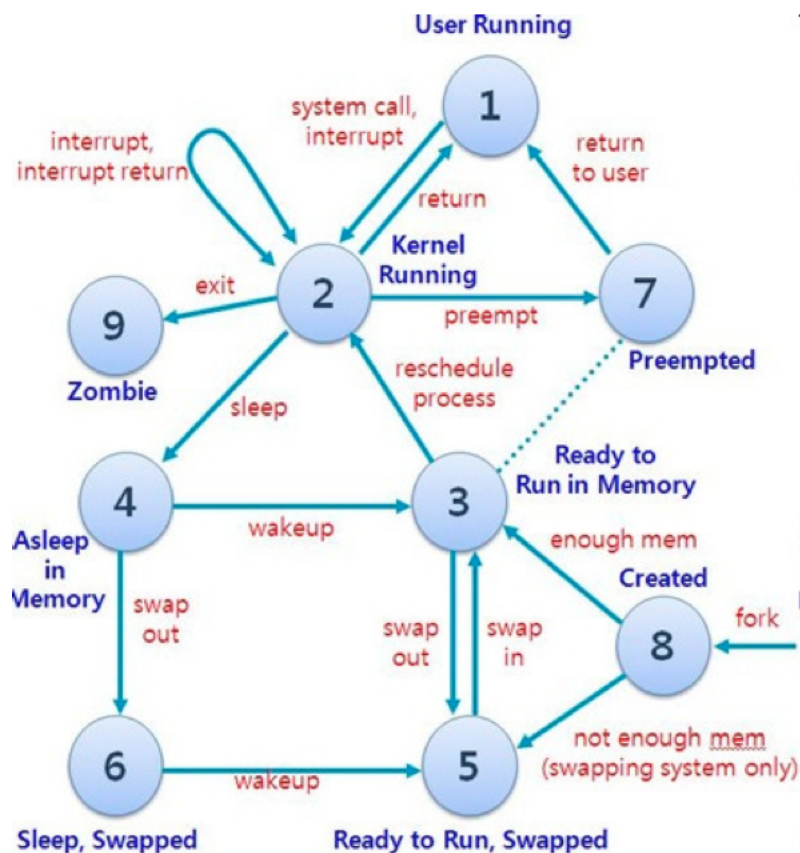
- Ein Prozess hat mindestens zwei Ausführungsmodi:

User Mode Es wird der normale Programmcode ausgeführt.

Kernel Mode Es werden Systemaufrufe ausgeführt oder Ausnahmen behandelt.

- Der Übergang erfolgt durch einen Systemaufruf durch das Programm, eine Ausnahmesituation oder durch asynchrone Events
- Beide Modi haben separate Segmente und sind voneinander abgesichert.

1.2.4 Zustände

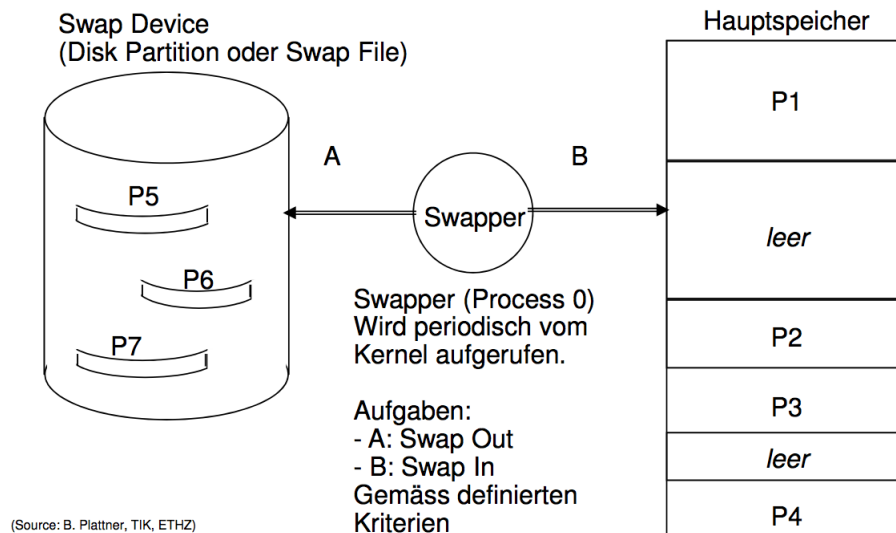


1.3 Memory

1.3.1 Virtual Memory

- Erweiterung des Hauptspeichers pro System (mehr Prozesse im System als Speicher verfügbar), oder pro Prozess (einzelner Prozess grösser als verfügbarer Hauptspeicher).
- Systematische Abstraktion für systemspezifische Overlay-Techniken.
- Organisation des Hauptspeichers in gleich grosse, einheitlich adressierbare Einheiten.

1.3.2 Swapping

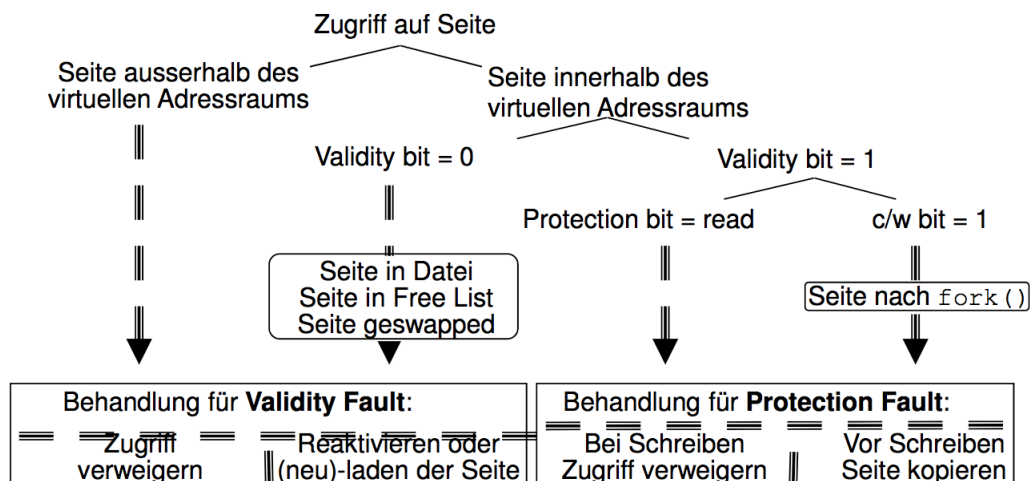


1.3.3 Paged Memory

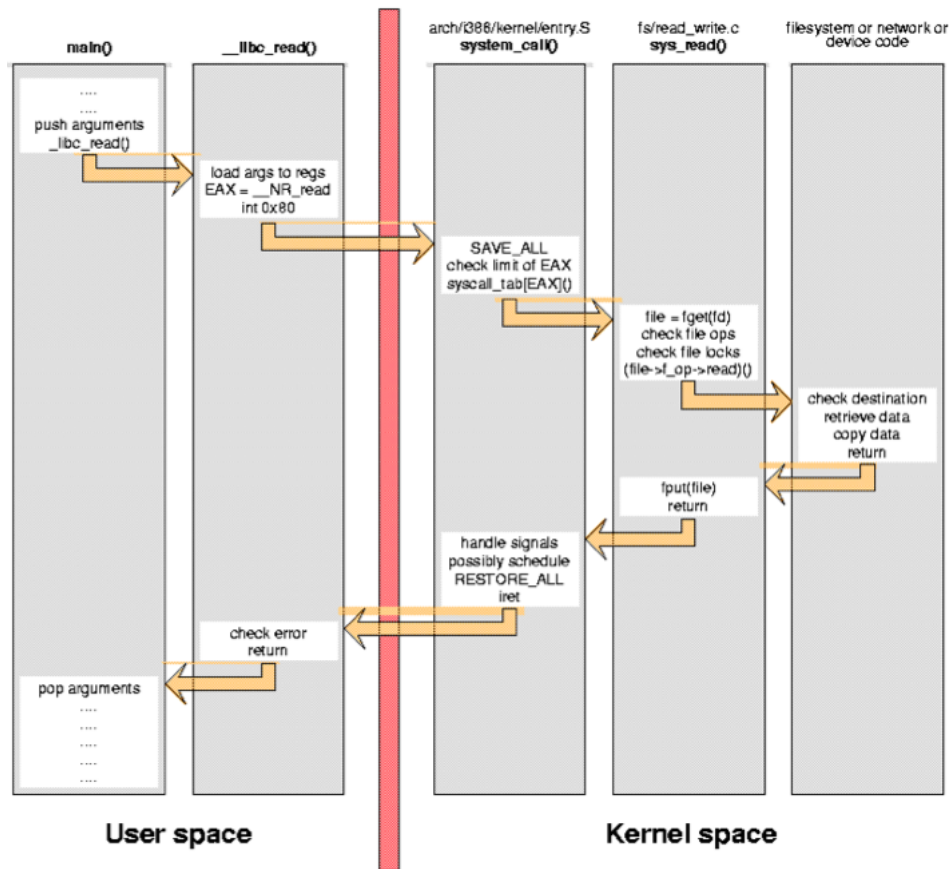
Design und Layout der Datenstrukturen für die seiten-orientierte Speicherverwaltung variieren zwischen Unix- und Linux-Varianten und sind zudem von den Hardware- Eigenschaften abhängig. Linux verwendet zum Beispiel eine 3-stufige Seitentabelle:

- Page Directory pro Prozess und für den BS-Kern
- Page Mid-level Directory (für 64 bit CPU-Architekturen) Paged Memory II
- Page Table, enthält Seitenbeschreibungen und Verweise auf den physische Speicherort

1.3.4 Fehlerzustände



2 System Call Schnittstelle



2.1 Prozess-System

fork() Erzeugung

exit() Beendigung

exec() Überlagerung des Prozesses

wait() Warten auf Prozesstermination (Kindprozesse)

sleep() Freiwilliges schlafen des Prozesses.

kill() Senden eines Signals (32 verschiedene)

signal() Signalbehandlung

2.2 Datei-System

creat() Anlegen einer Datei

mknod() Anlegen eines Ordners

open() Öffnen einer Datei

close() Schliessen einer Datei

unlink() Löschen einer Datei
read() Lesen aus einer Datei
write() Schreiben in eine Datei
lseek() Vorwärts-/Rückwärtsbewegung
ioctl() Kontrollieren der Eigenschaften
dup() Duplizieren eines Dateideskriptors
chown, chmod, umask Zugriffsrechte
chdir() Navigation im Dateisystem

2.3 Directory Handling

opendir() Öffnen eines Verzeichnisses
readdir() Lesen eines Verzeichnisses
writedir() Schreiben eines Verzeichnisses
closedir() Schliessen eines Verzeichnisses

2.4 Speicherverwaltung

malloc() Speicher allozieren
free() Speicher freigeben

2.5 Weitere

pipe() Basis-Interprozesskommunikation
socket() Interprozesskommunikation lokal oder u?ber Netzwerke