

Operativni sistemi

Diana Šantavec
diana.santavec@gmail.com

Istraživačka stanica Petnica

20.04.2023.



Sadržaj

- Pojam
- Učitavanje operativnog sistema
- Procesi
- Planeri procesa
- Zaštita memorije
- Fajl sistemi

Uvod

Šta je operativni sistem?

Uvod

- Program koji omogućava aplikacijama jednostavniji pristup hardveru
- Kontrolira izvršavanje aplikacija
- Olakšava pisanje programa visokog nivoa
- Omogućava nezavisnost programa od hardvera

Istorija

- Prvi računari su samo izvršavali dati program (ENIAC 1945)
- batch: učitao se više programa pa se izvrše
- 1970 - 1980 višekorisnički
- 1980 - 1990 prvi personalni (CP/M)
- ...

Učitavanje operativnog sistema

Šta se desi kada pritisnemo dugme?

Učitavanje operativnog sistema

- Prilikom pokretanja računara operativni sistem tek treba da se učitava iz neke trajne memorije
- BIOS-MBR
- UEFI-GPT

Firmware

- Kontrola niskog nivoa
- Na nekoj memoriji unutar uređaja
- BIOS i UEFI

Particija

- Logička sekcija diska sačinjena od kontinualnih sektora
- Partition table
 - Broj particija
 - Početak particije
 - Atributi particije
- MBR i GPT

BIOS-MBR

Kako zapravo radi?

BIOS

- BIOS (Basic Input/Output System)
- Sadrži rutine koje omogućavaju detekciju hardvera (monitor, miš, tastatura, disk, RAM,...)
- Učitava se sa čipa
- Testira hardver

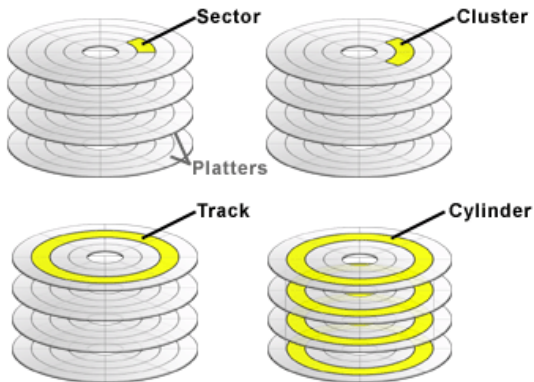
BIOS

- Real mode
- 16bit assembler

MBR (Master Boot Record)

- Boot sector (cilindar 0, glava 0, sektor 0)
- 512B
- Limit na 4 primarne particije do najviše 2TB
- Jedna kopija MBR-a

Dijagram hard diska



MBR (Master Boot Record)

- Svaki hard disk ga sadrži
- Postoje dve strukture:
 - Classic
 - Modern

MBR - classic

- Bootstrap Code (440B)
- Tabela partija (4x16B)
- Boot signature (0x55 0xAA)

MBR - modern

- Bootstrap Code (218B + 216B)
- Timestamp (6B)
- Disk signature (6B)
- Partition table (primarne particije) (4x16B)
- Boot signature (0x55 0xAA)

Kod u boot sector-u

```
mov ah, 0x0e
mov al, 'H'
int 0x10
mov al, 'e'
int 0x10
mov al, 'l'
int 0x10
mov al, 'l'
int 0x10
mov al, 'o'
int 0x10
jmp $
times 510 - ( $ - $$ ) db 0
dw 0xaa55
```

Kod u boot sector-u

```
SeaBIOS (version 1.16.2-debian-1.16.2-1)
```

```
iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+06FCB250+06F0B250 CA00
```

```
Booting from Hard Disk...  
Hello
```

■ Boot Sector Games

UEFI-GPT

Zašto onda imamo UEFI-GPT?

UEFI

- Standardizovan
- Nije ograničen na 16bita
- Nije vezan za neku arhitekturu (ne mora da koristi x86 set instrukcija)
- Bolje performance

GPT

- Najviše 128 particija
- Limit od 18exabytes
- Više kopija GUID tabela particija
 - Mogu se čuavati na početku i kraju

UEFI

- Safe boot
- Binarni potpis softvera za butovanje
- Emulacija ranijih BIOS firmware-a

Bootloader

- Nalazi se na određenoj particiji
- Koji operativni sistem, gde, odakle, parametri, . . .
- NTLDR, BOOTMGR, GRUB2, itd.

Učitavanje operativnog sistema

- Učitavanje fajl sistema
- Učitavanje konfiguracionih fajlova
- Lista operativnih sistema (ako ih ima više)
- Pokretanje odabranog

Pokretanje operativnog sistema

- Prvi proces u Linuksu - init
- Pokreće ostale procese
- systemd

Procesi

Šta je proces?

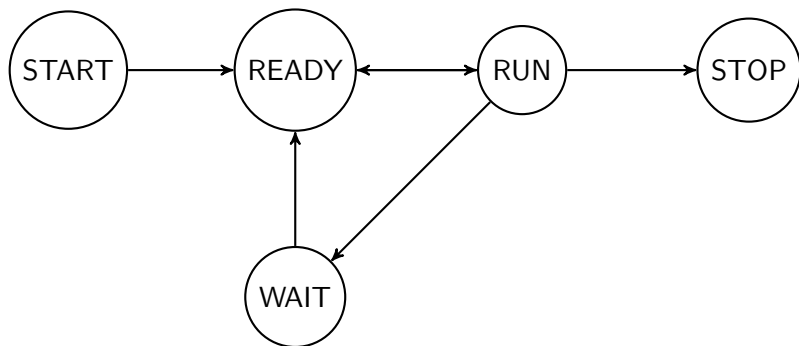
Pojam

- Program kada se izvršava
- Sadrži podatke o zauzetim ulazno/izlaznim uređajima, korisniku ,zauzetim fajlovima,...

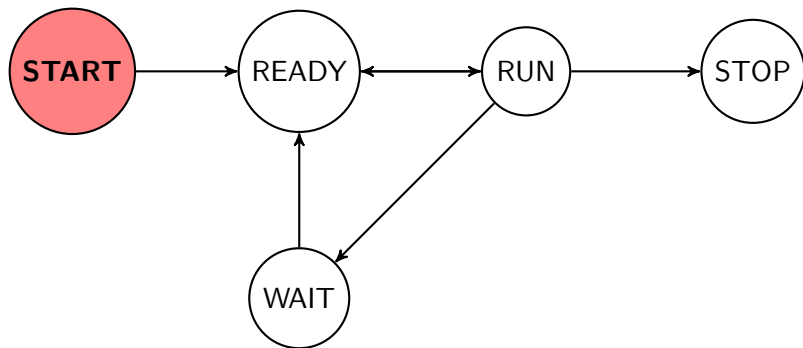
```
dianas@dianas-ThinkPad ~-> ls /proc/3123
arch_status      cpuset           limits           numa_maps        schedstat        task/
attr/            cwd@             loginuid         oom_adj          sessionid        timers_offsets
autogroup        environ          map_files/       oom_score         setgroups        timers
auxv             exe@             maps            oom_score_adj    smaps            timerslack_ns
cgroup          fd/             mem             pagemap          smaps_rollup     uid_map
clear_refs       fdinfo/         mountinfo        patch_state      stack            wchan
cmdline          gid_map          mounts           personality       stat
comm             io              mountstats      projid_map       statm
coredump_filter  ksm_merging_pages net/             root@            status
cpu_resctrl_groups ksm_stat        ns/             sched            syscall
dianas@dianas-ThinkPad ~-> █
```

Proces u memoriji

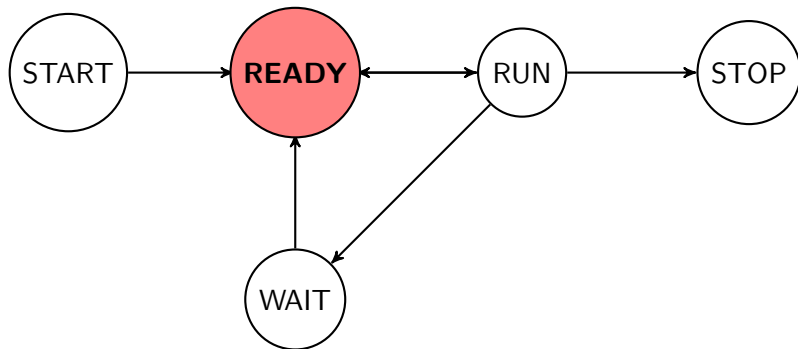
Promena stanja procesa



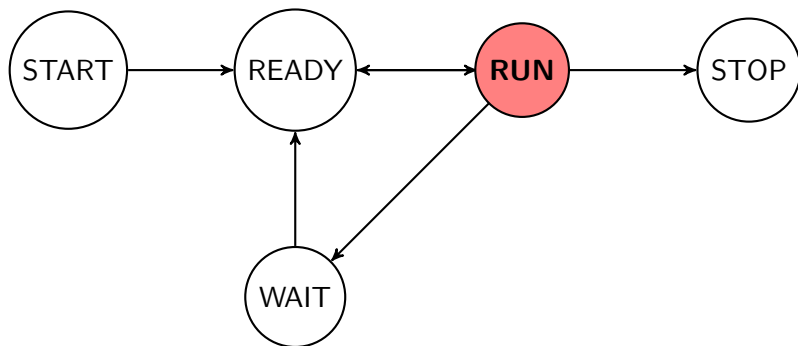
Promena stanja procesa



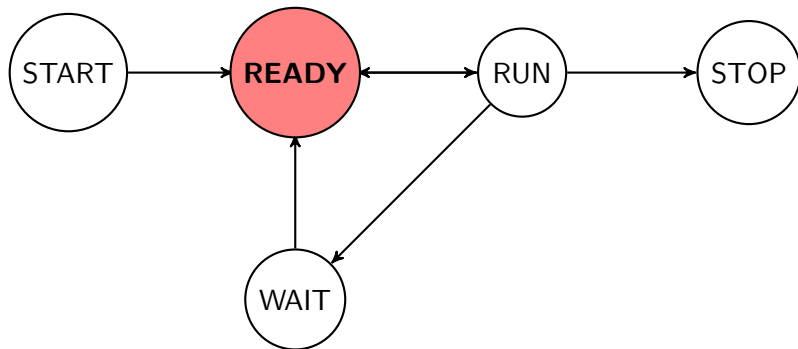
Promena stanja procesa



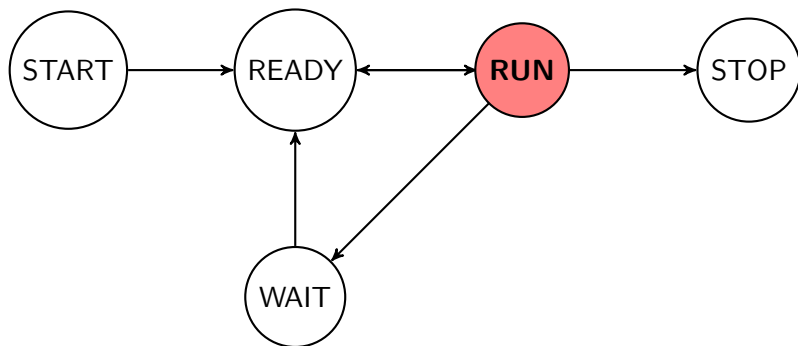
Promena stanja procesa



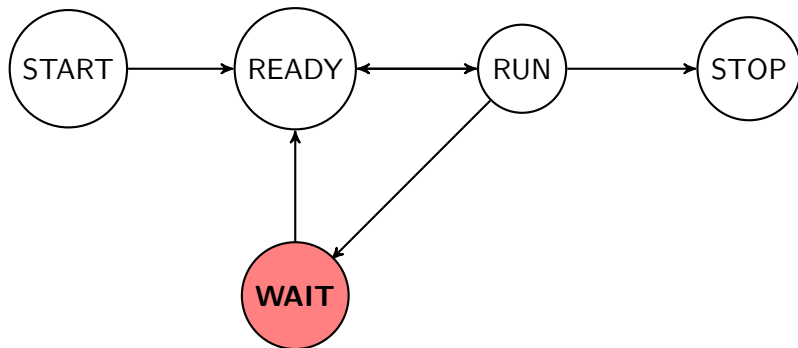
Promena stanja procesa



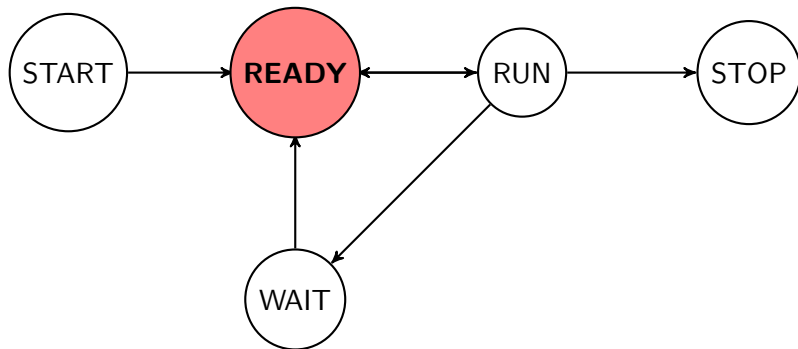
Promena stanja procesa



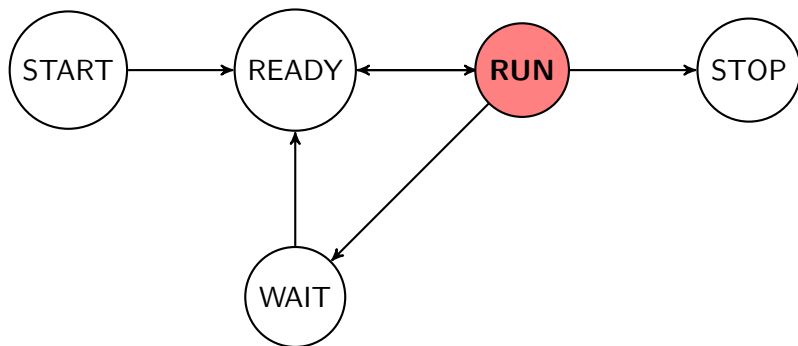
Promena stanja procesa



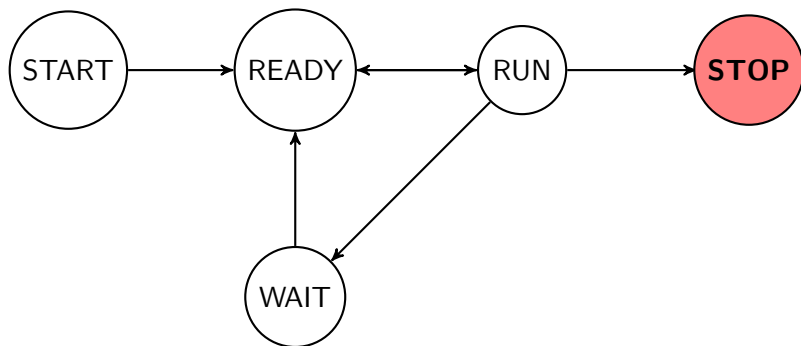
Promena stanja procesa



Promena stanja procesa



Promena stanja procesa



Fork

- Pravi kopiju (dete) originalnog procesa (roditelj)
- Gašenje/pucanje roditeljkog procesa prouzrokuje gašenje deteta procesa
- Dete proces zadržava i otvorene fajlove, ali su tokovi različiti

Niti (thread)

- Deo procesa
- Podela poslova na manje delove
- Paralelizacija procesa

Broj korisnika

- Singleuser
- Multiuser

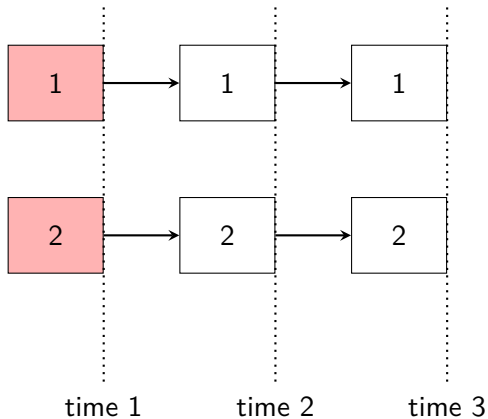
Izvršavanje procesa

- Sekvencijalno
- Time sharing
- Paralelno

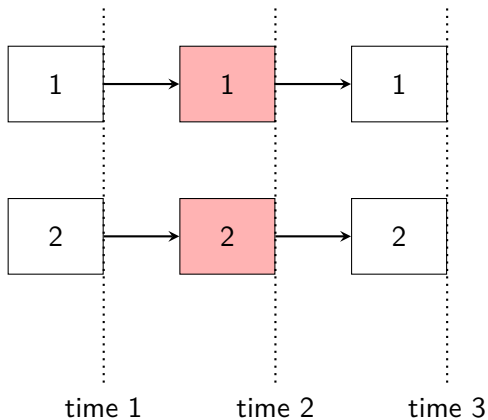
Sekvencijalno

Time sharing

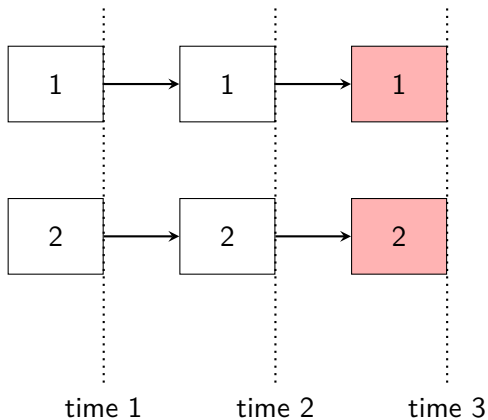
Parallel Execution at Time 1



Parallel Execution at Time 2



Parallel Execution at Time 3



Zastoj

Rešavanje zastoja

- Sprečavanje
- Dozvoliti da se desi, rešiti
- Ako se desi restartovati sistem (Windowd, Unix)

Planeri procesa

Kako operativni sistem smenjuje procese?

Pojam

- Programer ne mora da vodi računa da li će proces prepuštati resurse
- Omogućava "bolju" smenu procesa u zavisnosti od potreba operativnog sistema
- Omogućavaju efikasniju raspodelu resursa
- Cilj da se maksimizuje upotreba procesa i minimizuje vreme čekanja

Tipovi

- CPU
- non-preemptive
 - Proces ne može biti zaustavljen u toku izvršavanja
- preemptive
 - Planer procesa može prekinuti izvršavanje procesa

Bitna vremena

- Vreme izvršavanja (execution time)
- Vreme ulaska u spremno stanje (arrival time)
- Vreme završavanja (finish time)
- Vreme izvršavanja na procesoru (CPU execution time)

Tipovi procesa

- Većinu izvršavanja iziskuju procesorske resurse (CPU bound)
- Većinu izvršavanja provode čekajući signal (I/O bound)

Poznati planeri procesa

- First-come first-served
- Longest job first
- Shortest job first
- Round robin
- Prioritetni

U realnom vremenu

- Hard real time
- Soft real time

Upravljanje memorijom

Kako se kontroliše upotrebna RAM-a?

Pojam

- Vođenje računa o popunjavanju (prazni/slobodni)
- Dodela memorije
- Oslobađanje memorije

Adrese

- Simboličke adrese
 - Konstante, promenljive, labela instrukcija
- Relativne adrese
 - Kompajler simboličke prebacuje u relativne
- Fizičke adrese
 - Kada se program učitava u memoriju

Swap

- Korišćenje trajne memorije za nadoknadu RAM-a
- Više memorije
- Duže vreme za učitavanje programa (slanje na disk je sporije)

Paging

- Podela procesa na blokove
- Jednaki blokovi RAM-a
- Ne moraju biti uzastopni
- Demand Paging

Adrese

- Virtualne (protected mode)
 - generiše program
 - ne mora ceo program biti stalno u memoriji
- Fizičke
 - MMU

Alokacija memorije

- First fit
- Best fit
- Worst fit

Fragmentacija

- Interna fragmentacija
- Eksterna fragmentacija

Interna fragmentacija

- Pojavljuje se kadaje memorija izdeljena u fiksne blokove
- Procesu se dodeli više memorije nego što mu treba
- Ako se memorija ne deli u blokove, fragmentacija će biti izbegnuta

Eksterna fragmentacija

- Pojavljuje se kada su blokovi proizvoljne ili promenljive veličine
- Iako postoji dovoljno memorije, memorija je "razbacana"
- Rešenje je da se sva slobodna memorija pregrupiše u jedan blok
- Drugo rešenje je paging

Fajl sistem

Šta je zapravo fajl?

Fajl

- Informacije sačuvane u trajnoj memoriji
- Meta podaci (prava pristupa, modifikacija, vreme pravljenja, korisnik,...)

Fajl sistemi

Šta je fajl sistem?

Fajl sistemi

Kako pronalazimo fajlove na disku?

Fajl sistemi

- FAT32
- NTFS
- EXT4
- APFS

FAT32

- *File Allocation Table*
- File Allocation Table odvojena od podataka
- 4GB - 1B
- Particije do 2TB (sektori 512KB)

NTFS

- *New Technology File System*
- Ne zavisi od veličine sektora
- Alocira više kontinualnih sektora ($2n$) - volume
- Ograničenje na veličinu fajla je mnogo veće
- Master File Table sadrži podatke o volume-ima

Fajl

- Različiti tipovi fajlova
 - obični
 - direktorijum
 - specijalni
- Imaju različite strukture

Direktorijum

- Sadrži više fajlova
- Operacije
 - Kreiranje fajlova
 - Brisanje fajlova
 - Pretraga fajlova
 - Izlistavanje fajlova

Pristup fajlu

- Sekvencijalno
- Direktno/nasumično
- Preko indeksa

Sekvencijalno

Direknot/nasumično

Preko indeksa

- Indeksni fajl (logički ključevi na fizičke adrese)
- Brzina i efikasnost

Alokacija memorije za fajl

- Contiguous Allocation
 - Zauzeta memorija za fajl je kontinualna
- Linked Allocation
 - Sadrži pokazivače na različite blokove diska
- Indexed Allocation
 - Sadrži listu svih indeksa blokova fajla

Razvoj strukture direktorijuma

- Jedan nivo
- Dva nivoa (korisnik/direktorijum/fajl)
- Stablo
- Acikličan graf

Jedan nivo

- Višekorisnički operativni sistem
- Svaki korisnik ima svoj direktorijum

Dva nivoa

- Kao za jedan nivo ali korisnik može da pravi direktorijume u tom nivou

Stablo

- Nema ograničenja na dubinu
- Ne postoje linkovi

Acikličan graf

- Uvode se i linkovi

Operacije nad sistemom fajlova

- Otvaranje/zatvaranje fajlova
- Dodavanje fajlova
- Brisanje fajlova
- Premeštanje fajlova

Pouzdanost fajl sistema

- Loši blokovi
- Backup (sigurnosne kopije)

Dokle smo stigli?

- Pojam
- Učitavanje operativnog sistema
- Procesi
- Planeri procesa
- Zaštita memorije
- Fajl sistemi

Šta dalje?

- Virtualizacija
- Cloud
- Distribuirani sistemi
- Operativni sistemi koji se izvršavaju u realnom vremenu
- Embedded sistemi
- ...

HVALA NA PAŽNJI!

Pitanja?