



OPERATIVNI SISTEMI

II - Jezgro operativnog sistema

II - Jezgro operativnog sistema

S A D R Ž A J

2.1 Jezgro

2.2 Proces

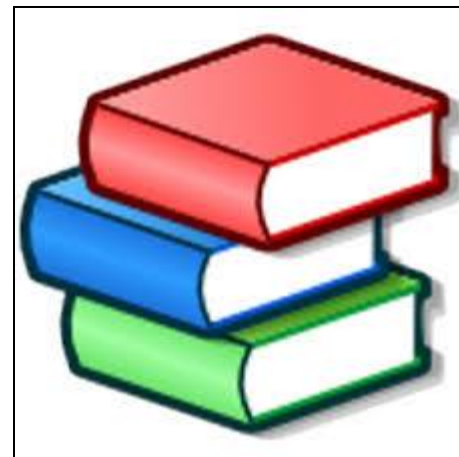
2.3 Predstavljanje procesa

2.4 Raspoređivanje procesa

2.5 Operacije nad procesima

2.6 Rutine za interprocesnu komunikaciju

2.7 Laki i teški procesi



2.1 Jezgro (*Nucleus* ili *Kernel*)

- predstavlja osnovni deo svakog operativnog sistema
- najbliže je hardveru računara
- izuzetak su Windows OS sa NT arhitekturom u osnovi, gde se ispod jezgra nalazi HAL sloj (HAL- *Hardware Abstraction Level*) koji omogućava OS-u da vidi različiti hardver na isti način
- jezgro ima zadatak da upravlja procesima
 - stvara okruženje u kojem mogu postojati procesi
 - određuje kada i na koje vreme će proces dobiti procesor
 - omogućava međusobnu komunikaciju između procesa
- jezgro nije potrebno ako se koriste programi napisani u jezicima koji mogu direktno pristupiti hardveru (mašinski jezik)
- funkcionisanje jezgra je uslovljeno postojanjem određenih komponenti na nivou hardvera

2.1 Jezgro (*Nucleus* ili *Kernel*)

Neophodne komponente hardvera

- **mehanizam prekida**
 - omogućava izvršavanje prekidne rutine koja obezbeđuje prelazak sa korisničkog na upravljački program
 - pri tome ovaj mehanizam čuva vrednost programskog brojača prekinutog korisničkog programa i pokreće upravljački program sa fiksne memorijske lokacije
- **zaštitni mehanizam adresiranja memorije**
 - onemogućava da jedan proces upiše svoje prateće podatke u deo memorije koji je dodeljen drugom procesu
 - time se čuva integritet procesa i podataka u radnoj memoriji
- **časovnik realnog vremena (*real-time clock*)**
 - vrši kontrolu i evidentiranje potrošnje resursa računara za sve aktivne procese
 - koristi se i za raspoređivanje i izvršavanje raznih poslova

2.1 Jezgro (*Nucleus* ili *Kernel*)

Neophodne komponente hardvera

- skup privilegovanih instrukcija
 - instrukcije koje su dostupne OS-u ali ne i korisničkim programima
 - omogućavaju OS-u da
 - maskira prekide
 - dodeli procesor drugom procesu
 - pristupi zaštićenim registrima u memoriji
 - izvršu U/I operaciju
 - zaustavi procesor
 - prilikom izvršavanja ovih instrukcija OS se nalazi u sistemskom režimu (*supervisor mode*)
 - korisnički programi privilegovane instrukcije ne mogu direktno izvršavati već isključivo uz pomoć sistemskih poziva

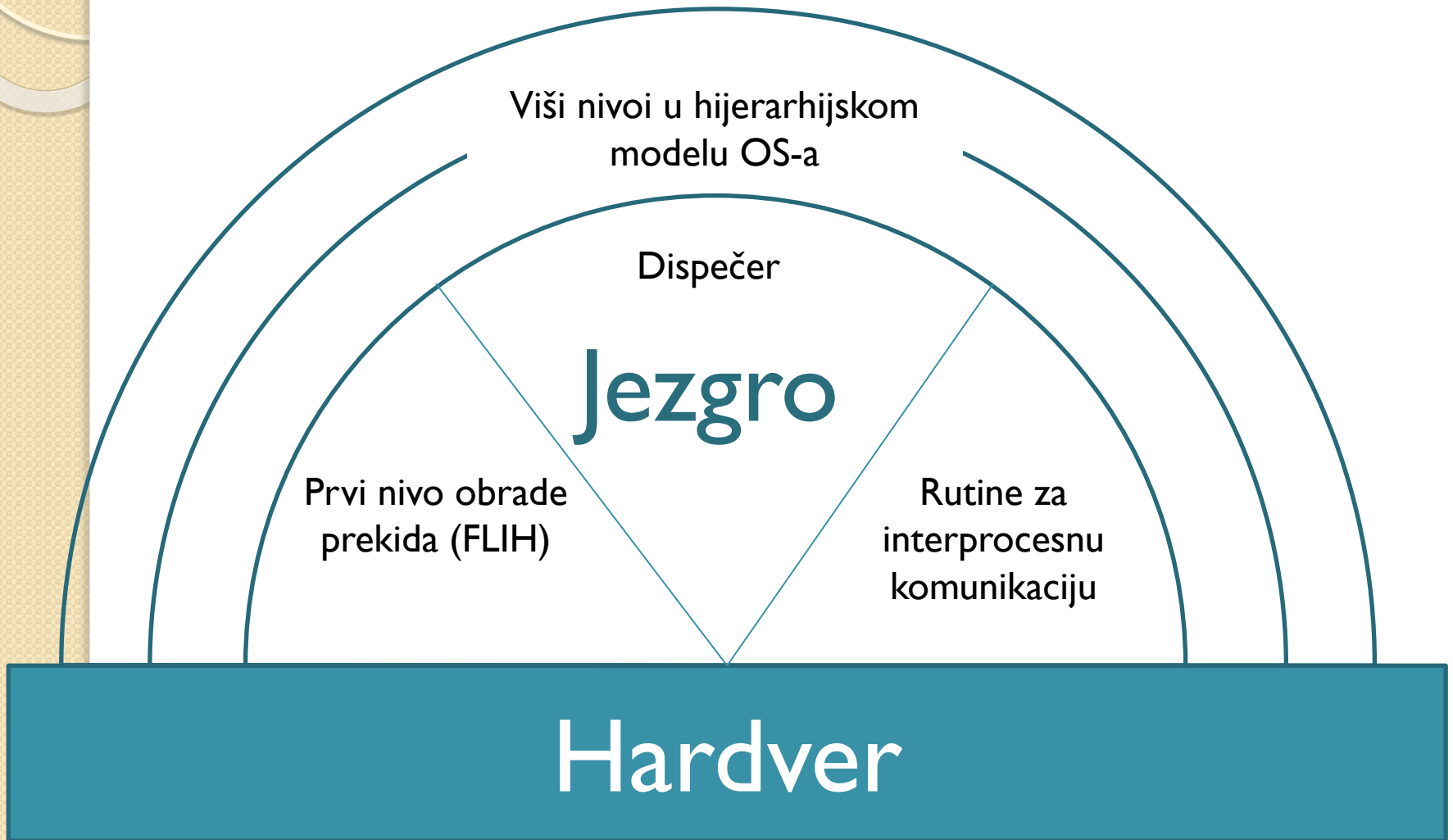
2.1 Jezgro (*Nucleus* ili *Kernel*)

Osnovni delovi jezgra teoretskog modela OS-a

- prvi nivo obrade prekida (*FLIH - First Level Interrupt Handler*)
 - čine ga rutine za određivanje izvora prekida i iniciranje servisa koji treba da se izvrše na osnovu tog prekida
- dispečer sistema (*dispatcher, low-level scheduler*)
 - dodeljuje procesor određenom procesu na osnovu nekog unapred definisanog algoritma za dodeljivanje
- rutine za interprocesnu komunikaciju
 - obezbeđuju da procesi koji se konkurentno (paraleno) izvršavaju mogu međusobno da komuniciraju putem
 - slanja poruke (*send message*)
 - semaforske tehnike
 - korišćenja imenovanih cevi (*named pipes*)
 - korišćenjem deljive memorije

2.1 Jezgro (*Nucleus* ili *Kernel*)

Osnovni delovi jezgra teoretskog modela OS-a



2.2 Procesi

- proces je program ili deo programa koji je u stanju izvršavanja, zajedno sa svojim resursima, poput registara i memorije
- kada se program zapisan u vidu datoteke učitava u memoriju, on postaje proces
- svaki korisnik koji izvršava određenu aktivnost će biti predstavljen sa posebnim procesom
- program koji bi se razdelio na dva dela koji se simultano izvode bi bio predstavljen sa dva procesa

2.2 Procesi

- svaki proces ima tri osnovna dela, odnosno sekcije
 - **programska ili tekstualna sekcija** koja se ne menja (*read-only*) i koja sadrži programski kôd
 - **stek sekcija** koja sadrži privremene podatke
 - parametri za procedure
 - povratne adrese
 - lokalne promenljive...
 - **sekcija podataka** koja sadrži globalne promenljive koje proces obrađuje
- radi uštede memorije, većina OS-a formiraće zajednički tekstualni segment za sve procese koji su nastali pokretanjem jednog programa, budući da je tekstualna sekcija nepromenjiva
- međutim, svaki od tih procesa imaće posebnu stek sekciju i sekciju podataka

2.2 Procesi

- osim memorijskih sekcija proces obuhvata
 - vrednost programskog brojača
 - vrednosti ostalih važnih registara procesora
 - U/I resurse koje eventualno koristi
 - datoteke
 - U/I uređaji...
- pored navedenog imena "procesi" za isti tip aktivnosti procesora koristi se takođe i naziv "poslovi" (*jobs*)

2.3 Predstavljanje procesa

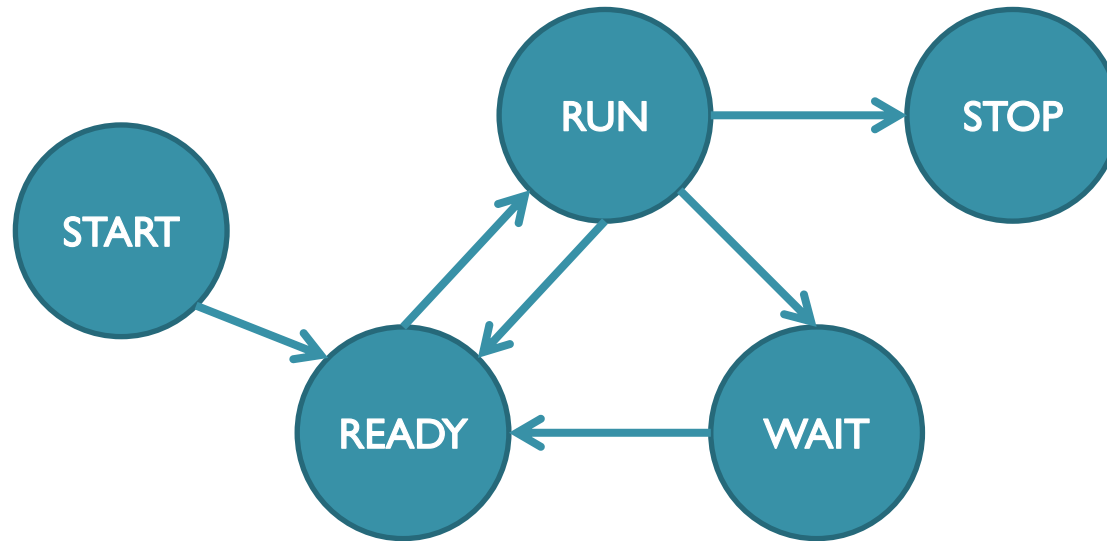
- da bi OS mogao da upravlja procesima potrebni su mu podaci o svakom pojedinačnom procesu
- ti podaci se čuvaju u delu radne memorije koji se naziva kontrolni blok procesa (PCB - *Process Control Block*)
- zahvaljujući njemu izvršavanje programa se može prekidati i nastavljati više puta
- kontrolni blok sadrži sledeće informacije o procesu
 - **jedinstveni identifikator procesa** (PID - *process ID*) koji uniformno definiše svaki proces
 - **konteks (okruženje) procesa** sadrži podatke koji se čuvaju prilikom oduzimanja procesora a koje generiše sam hardver
 - programski brojač, sadrži adresu sledeće instrukcije koje proces treba da izvrši
 - vrednosti registara i pokazivači na deo memorije koji se koristi
 - **konteks (okruženje) procesa** naziva se još i **hardverski kontrolni blok procesa**

2.3 Predstavljanje procesa

- kontrolni blok sadrži sledeće informacije o procesu (nastavak)
 - informacije o raspoređivanju odnosno prioritetu procesa
 - informacije o memoriji procesa
 - status U/I uređaja dodeljenih procesima
 - lista otvorenih datoteka
 - trenutno stanje procesa
 - START (*Created, New*) - trenutak formiranja procesa
 - WAIT - proces čeka na određeni resurs (šampač, npr.), rezultat nekog drugog procesa ili završetak neke U/I operacije
 - READY - proces ima sve resurse, čeka na procesor
 - RUN - proces se izvršava u procesoru
 - STOP - kraj izvršenja procesa
 - informacije za praćenje
 - utrošeno procesorsko vreme
 - vremenska ograničenja
 - brojevi korisničkih naloga
 - brojevi procesa...

2.3 Predstavljanje procesa

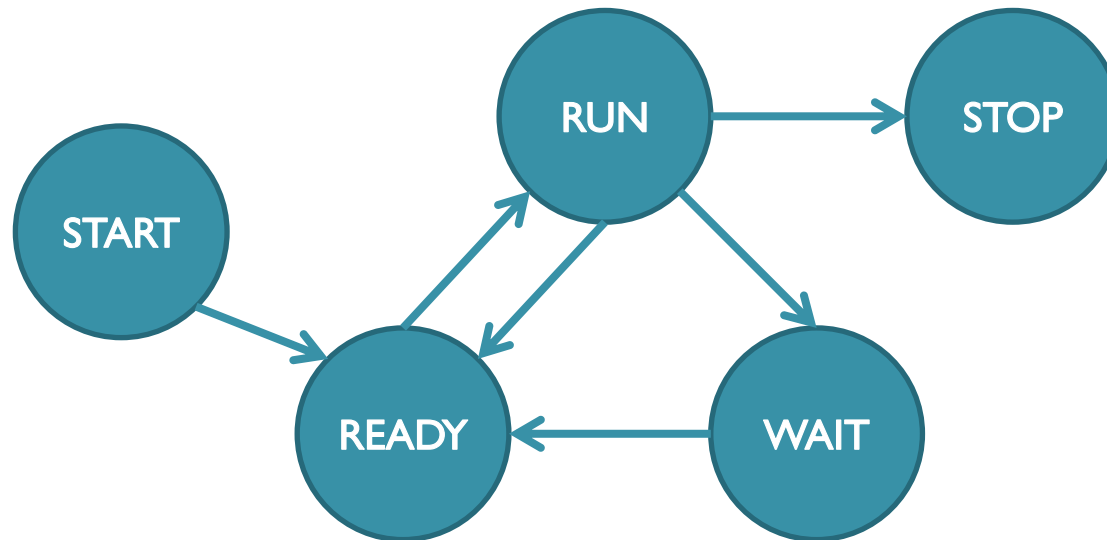
Dijagram stanja procesa



- svaki proces se mora nalaziti u nekom od prikazanih stanja
- tranzicije predstavljaju prelazak iz jednog stanja u drugo
 - Start-Ready - prebacivanje procesa u radnu memoriju
 - Ready-Run - dodeljivanje procesora procesu
 - Run-Ready - oduzimanje procesora procesu
 - usled isteka vremena koje mu je dodeljeno
 - ako naiđe proces višeg prioriteta

2.3 Predstavljanje procesa

Dijagram stanja procesa



- tranzicije predstavljaju prelazak iz jednog stanja u drugo
 - Run-Wait - oduzimanje procesora procesu ukoliko
 - je resurs potreban procesu za izvršenje zauzet
 - se čeka rezultat operacije koji izvršava neki drugi proces
 - proces čeka neki unapred programirani trenutak
 - Wait-Ready - proces je dobio resurs koji mu je neophodan da bi se izvršio te se vraća na kraj procesorskog reda
 - Run-Stop - proces završava rad (prirodno ili nasilno)

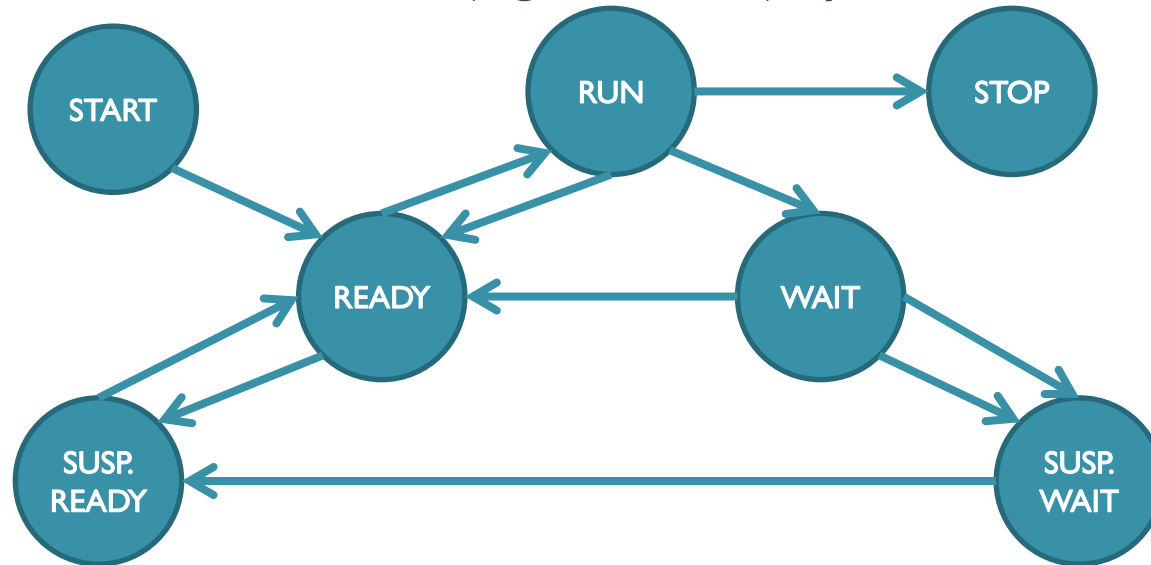
2.3 Predstavljanje procesa

Prošireni dijagram stanja procesa

- u nekim OS kao što je UNIX postoji mogućnost privremenog prekida izvršenja procesa
- polazi se od toga da korisnik koji inicira proces ima pravo da ga privremeno zaustavi (*suspend*), i da nakon nekog vremena nastavi njegovo izvršenje (*revive*)
- sam OS može u određenim situacijama suspendovati određene procese kako bi se izbegli određeni zastoji
- izvršenje procesa se može prekinuti samo u stanjima WAIT i READY
- na ovaj način formiraju se dva dodatna stanja
 - SUSPENDED READY - ako je proces suspendovan u stanju čekanja na procesor
 - SUSPENDED WAIT - ako je proces suspendovan u stanju čekanja na resurs

2.3 Predstavljanje procesa

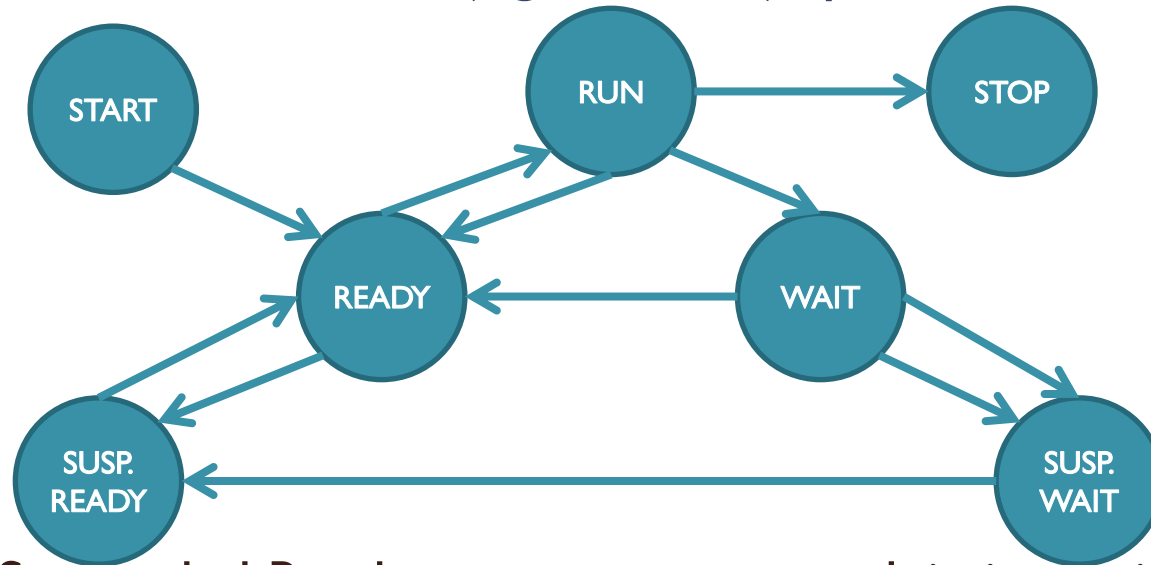
Prošireni dijagram stanja procesa



- na ovaj način se dobija prošireni dijagram stanja procesa sa sedam stanja i pet novih tranzicija
 - Ready-Suspended Ready - proces se suspenduje iz stanja READY iz sledećih razloga
 - na sistemu je suviše veliki broj procesa u stanju Ready,
 - izbegavanje zastoja
 - eksplicitno obustavljanje procesa od strane korisnika

2.3 Predstavljanje procesa

Prošireni dijagram stanja procesa



- Wait-Suspended Ready - proces se suspenduje iz stanja WAIT
- Suspended Wait-Suspended Ready - oslobođen je resurs za dalje izvršenje procesa, ali je proces i dalje suspendovan
- Suspended Ready-Ready - proces je odmrznut i dovodi se na kraj procesorskog reda, ova tranzicija je moguća samo na eksplicitan zahtev korisnika
- Suspended Wait-Wait - proces je odmrznut ali resurs za njegovo izvođenje nije oslobođen, ova tranzicija je moguća samo na eksplicitan zahtev korisnika