

Procesi

Računarski sistem koji ima samo jedan procesor može da izvršava samo jednu instrukciju u jednom trenutku. OS mora da svim korisnicima računarskog sistema i svim programima obezbedi pristup procesoru i ne sme dopustiti nikome da ostvari monopol nad njim. Da bi obavio ovaj zadatak, operativni sistem mora da ima određene informacije o svakom korisniku i svakom programu u računaru, u svakom trenutku.

Većina operativnih sistema upravlja procesima (*processes*) i zadacima (*tasks*). Proces predstavlja program ili deo programa u stanju izvršavanja, zajedno sa resursima koji su potrebni za rad programa. Jednostavno rečeno, program je pasivan objekat, to jest datoteka na disku. Kada se datoteka učita u memoriju, ona postaje proces tj aktivni objekat koji ima svoje resurse, poput registara i memorije. Sam operativni sistem je takođe sastavljen od niza procesa.

Svaki proces ima 3 fundamentalna memorijska dela, to jest sekcije:

- Programska ili tekstualna sekcija, koja se ne menja (*read only*) i koja sadrži programske kôde.
- Stek sekcija (*stack section*), koja sadrži privremene podatke (parametre za procedure, povratne adrese, lokalne promenljive)
- Sekcija podataka (*data section*), koja sadrži globalne promenljive

Osim memorijskih sekcija, proces obuhvata i vrednost programskog brojača (*program counter*) kao i vrednost ostalih važnih registara procesora. Proses obuhvata i ulazno-izlazne resurse koje eventualno koristi, kao što su datoteke i razne vrste ulazno-izlaznih uređaja.

Aktivnosti procesora se u literaturi nazivaju različitim imenima: procesi (*processes*), poslovi (*jobs*), zadaci (*tasks*). U teoriji operativnih sistema procesi i poslovi su relativno slični pojmovi. U operativnom sistemu UNIX, posao se definiše grupa procesa koja radi u pozadini ili je suspendovan.

Predstavljanje procesa

Svaki program se sastoji od niza instrukacija i statičkog je karaktera. Za razliku od programa, proces je dinamička struktura i najjednostavnije se može opisati kao program u stanju izvršenja tj kao niz aktivnosti koje su posledica izvršenja programa. Deo procesa su i podaci koji tu aktivnost opisuju, tj podaci koji su neophodni za upravljanje procesom. Ove podatke generiše i koristi operativni sistem, to jest dispeče, a u literaturi se pominju pod raznim imenima: **kontrolni blok procesa** (*Process Control Block, PCB*), **vektor stanja ili deskiptor procesa** (*process descriptor*).

Ako posmatramo višeprocesni operativni sistem sa dva aktivna procesa (P1 i P2). Proses P1 koji se izvršava, u jednom trenutku se dovodi u stanje čekanja na resurs. Nakon toga, nastavlja se izvršavanje procesa P2 koji posle određenog vremena takođe biva blokiran. U međuvremenu se oslobođa resurs neophodan za izvršenje procesa P1, pa proces P1 može nastaviti izvršenje. Da bi operativni sistem znao gde treba nastaviti izvršavanje, svakom procesu se dodeljuju prateće informacije tj jedinstveni kontrolni blok.

Kontrolni blok je deo radne memorije to jest memorijska struktura sa osnovnim informacijama o procesu, koje operativni sistem koristi za upravljanje tim procesom. Zahvaljujući kontrolnom bloku, izvršavanje programa se može prekinuti i nastavljati više puta. U informacije iz kontrolnog bloka spadaju:

- Ime ili jedinstveni identifikator procesa (PID)
- Kontekst (okruženje) procesa
- Prioritet procesa
- Informacije o memoriji procesa
- Lista otvorenih datoteka
- Status zauzetih ulazno-izlaznih resursa
- Trenutno stanje procesa

Uopšteno, kontekst procesa čine podaci koji se čuvaju prilikom oduzimanja procesora, a njih generiše sam hardver: programski brojač, vrednosti registara, pokazivači na deo memorije koji proces koristi. Deo kontrolnog bloka procesa u kome se čuva kontekst još se naziva i **hardverski kontrolni blok procesa** ili **hardverski deskriptor procesa** (*hardware process descriptor*).

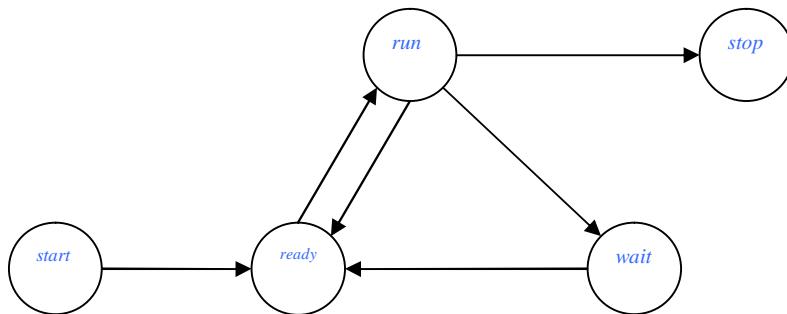
Dijagram stanja procesa

Proces se satoji od niza koraka koji slede jedan za drugim. Između dva koraka, proces može biti prekinut, a njegovo izvršavanje se može nastaviti u drugom trenutku, na istom ili drugom procesoru. Svi procesi koji uđu u računarski sistem prolaze kroz niz stanja tokom svog boravka u računaru. Jednostavno rečeno, stanje procesa (*process state*) opisuje ono što se udatom trenutku događa sa procesom. Provodenje procesa iz jednog stanja u drugo (*state transition*) obavlja operativni sistem.

Proces se može naći u nekoliko stanja (5 ili 7, zavisno od konačnog automata koji je korišćen za opisivanje stanja) a sledeća 3 su najznačajnija:

- Stanje izvršavanja (**RUN, RUNNING**) – Procesor izvršava instrukcije ovog procesa
- Stanje čekanja na procesor (**READY, RUNNABLE**) – Proces je dobio sve potrebne resurse osim procesora, spreman je za rad i čeka da mu se dodeli procesor
- Stanje čekanja na resurs (**WAIT, UNRUNNABLE**) – Proces čeka na neki događaj (npr da se izvrši štampanje), jer su za dalje izvršenje procesa potrebni resursi koji trenutno nisu na raspolaganju. Takođe, proces može ući u stanje WAIT ukoliko čeka završetak neke ulazno-izlazne operacije ili rezultat nekog drugog procesa.

Radi jednostavnosti izlaganja opisuje se konačni automat sa pet stanja. Dva dodatna pomoćna stanja vezana su za nastanak procesa, to jest trenutak formiranja procesa (START) i prestanak ili kraj izvršenja procesa (STOP). Proces u stanju STOP oslobađa sve resurse koje je zauzeo. Proces se u svakom trenutku mora naći u jednom od ovih stanja, a tranzicija iz jednog stanja u drugo vrši se zavisno od događaja vezanih za proces, resursa koje proces koristi i ostatka računarskog sistema.



Dijagram stanja procesa

Prilikom upravljanja procesima, dispečer prati i ispituje stanja u kontrolnim blokovima procesa. Npr, na osnovu toga, procesi se mogu dodati u red za čekanje na procesor. Dodavanje u red je simboličko-proces se fizički ne pomera s jedne lokacije na drugu, već se samo menjaju podaci u kontrolnim blokovima i pokazivači pomoću kojih se formiraju redovi, svaki proces počinje u stanju START, koje predstavlja vrstu pripreme, nakon čega se dovodi u stanje READY tj u stanje čekanja na dodelu procesora u procesorskom radu. Ovo je prva tranzicija u konačnom automatu.

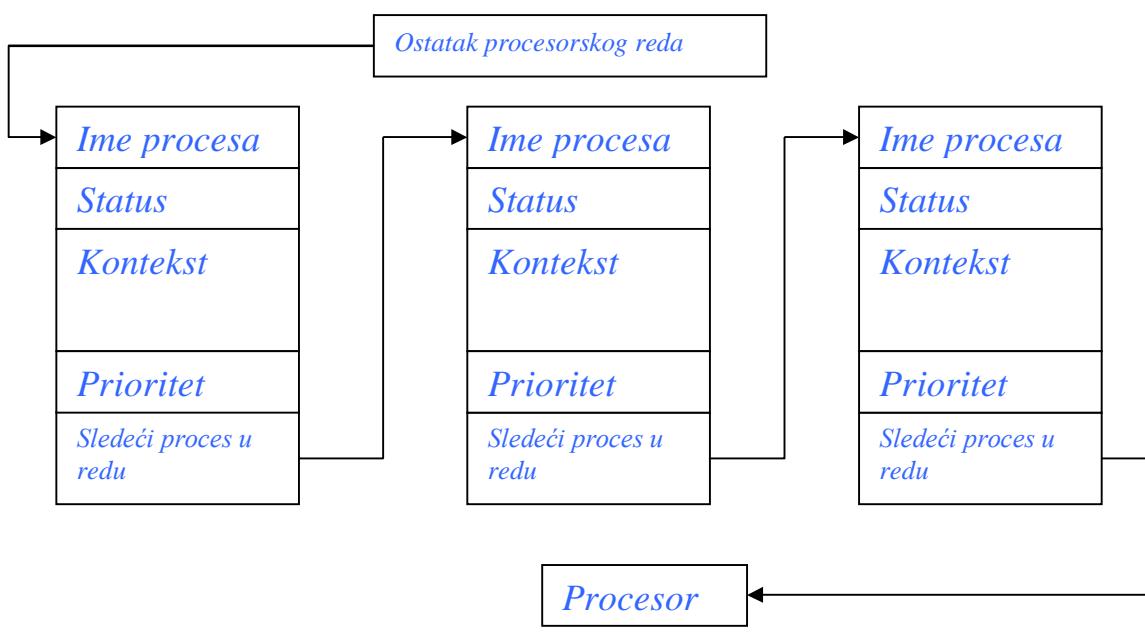
- **READY-RUN** – Dodela procesora procesu koji je došao na pošetak reda za dodelu procesa. Proces prelazi iz stanja čekanja na procesor u stanje izvršavanja kada procesor prekine izvršenje tekućeg procesa, tako da počinje izvršavanje novog procesa.
- **RUN-READY** – Oduzimanje procesora procesu nakon isticanja perioda u kojem mu je procesor dodeljen. Dodatno, ako je operativni sistem s preprežnjnjem (*pre-emptive*) procesor se može oduzeti i ukoliko najde procesor višeg prioriteta. Ova tranzicija je moguća samo u višeprocesnim operativnim sistemima.
- **RUN-WAIT** – oduzimanje procesora procesu ukoliko je resurs koji je potreban za izvršenje procesa zauzet. Ova tranzicija je moguća i u jednoprocesnim i u višeprocesnim operativnim sistemima (npr, spuler štampača ne može da nastavi štampanje dokumenta dok se u štampač ne

doda papir). Do ove tranzicije može doći i kada proces čeka rezultat operacije (izračunavanja) koju obavlja drugi proces.

- **WAIT-READY** – proces se vraća na kraj procesorskog reda nakon oslobađanja resursa koji je neophodan za rad. Tranzicija WAIT-RUN nije moguća u višeprocesnim operativnim sistemima. U takvim sistemima proces se uvek prvo dovodi u stanje READY.
- **RUN-STOP** – Proces završava rad (prirodno ili nasilno)

U stanju RUN se u jednom trenutku može naći najviše onoliko procesa koliko procesora ima u sistemu. Procesi koji se nalaze u stanju RUN jesu tekući procesi. U stanju RUN se proces ne može zadržati do svog završetka, zato što se u opštem slučaju posmatra višeprocesno okruženje u kome se procesor dodeljuje na određeno vreme, po nekom algoritmu. Na taj način se sprečava da jedan proces crpi resurse računara, dok drugi procesi čekaju u redu. Ukoliko operativni sistem omogućava pretpričanje, procesor se može oduzeti i u slučaju da nađe proces višeg prioriteta, čime se obezbeđuje blagovremeno izvršavanje kritičnih poslova.

Svi procesi koji se nalaze u stanju READY mogu se povezati u red na jednostavnoj listi. Ova linearne lista čini procesorski red u koji je uključen i tekući proces. Dispečer koristi procesorski red prilikom upravljanja procesima.



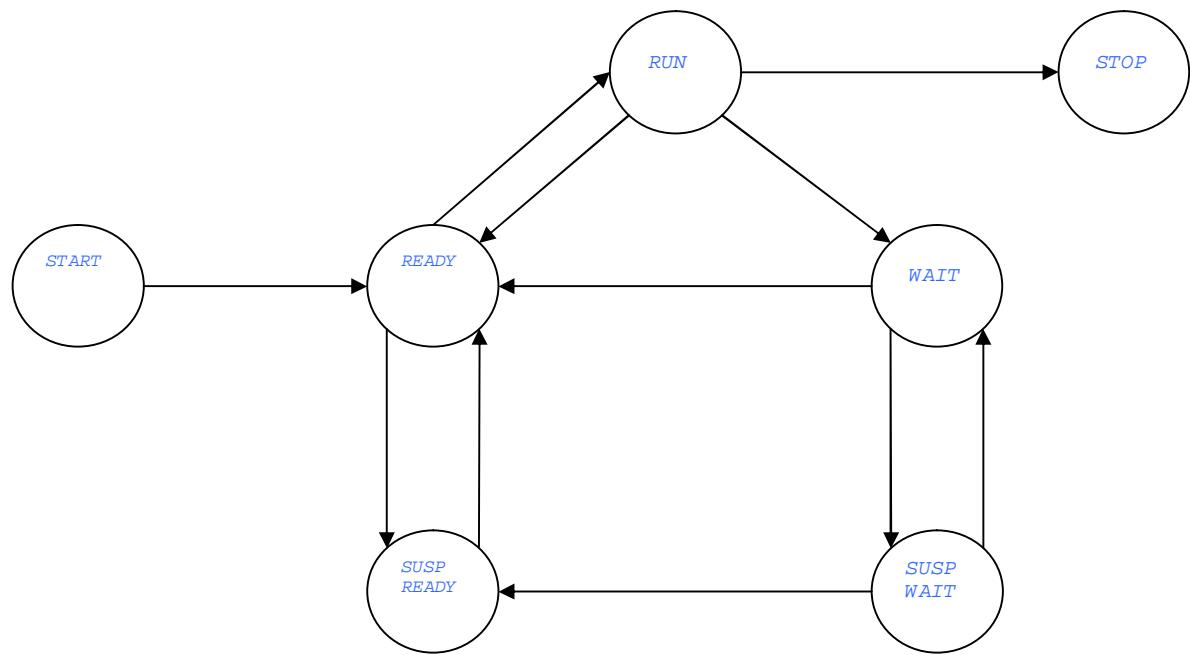
Prošireni dijagram stanja procesa

U nekim operativnim sistemima postoji mogućnost privremenog prekida izvršenja procesa. Proses čije je izvršenje privremeno prekinuto suspendovan je i prestaje da se takmiči za resurse. Ovaj model automata karakterističan je za operativne sisteme UNIX. Osnovna prepostavka je da korisnik koji inicira proces ima pravo da privremeno zaustavi njegovo izvršenje (*suspend*). Takođe, korisnik može u određenom momentu da nastavi izvršenje suspendovanog procesa (*revive*). Sam operativni sistem može, ukoliko postoji potreba, dovesti određen broj procesa u suspendovano stanje – tako se sprečavaju zastoji i pojava efekta zasićenja usled prevelike količine keširanih podataka. Suspendovanje procesa se može javiti i kao posledica *swap* prostora (prostor na disku za privremeno smeštanje neaktivnih procesa radi oslobađanja što veće količine RAM memorije za izvršenje drugih procesa). Izvršenje procesa se može prekinuti samo u stanjima WAIT i READY. Na taj način formiraju se dva dodatna stanja:

- **SUSPENDED-READY** – U ovo stanje proces dolazi ukoliko je suspendovan u stanju čekanja na procesor
- **SUSPENDED-WAIT** – U ovo stanje proces dolazi ukoliko je suspendovan u stanju čekanja na resurs

Nove tranzicije u proširenom dijagramu stanja:

- **READY-SUSPENDED READY** – Dovođenje procesa iz stanja READY u suspendovano stanje. Proces prolazi kroz ovu transakciju ako: (1) na sistemu postoji previše procesa u stanju READY, (2) treba izbeći zastoj (*deadlock*) ili (3) ako korisnik eksplisitno suspenduje proces
- **WAIT-SUSPENDED WAIT** – Dovođenje procesa u suspendovano stanje iz stanja WAIT
- **SUSPENDED WAIT – SUSPENDED READY** – Resurs neophodan za dalje izvršenje procesa je slobodan, ali proces je i dalje suspendovan
- **SUSPENDED READY – READY** – Proces je odmrznut, i dovodi se na kraj procesorskog reda. Ova tranzicija je moguća samo na eksplisitni zahtev korisnika
- **SUSPENDED WAIT – WAIT** – proces je odmrznut, ali resurs neophodan za njegovo izvršenje nije oslobođen. Ova tranzicija je moguća samo na eksplisitni zahtev korisnika



Prošireni dijagram stanja procesa