Procesi

Operativni sistemi 1

Institut za matematiku i informatiku
Prirodno-matematički fakultet, Kragujevac

doc. dr Miloš Ivanović
prof. dr Dragan Stojanović

Oktobar 2026. god.

O čemu će biti reči?

1. Definicija procesa
2. Stanja procesa
3. Opis procesa
4. Upravljanje procesom

Definicije procesa

Definicija (1)

Proces je program u izvršavanju.

Definicija (2)

Proces je primerak programa koji se izvršava u računaru.

Definicija (3)

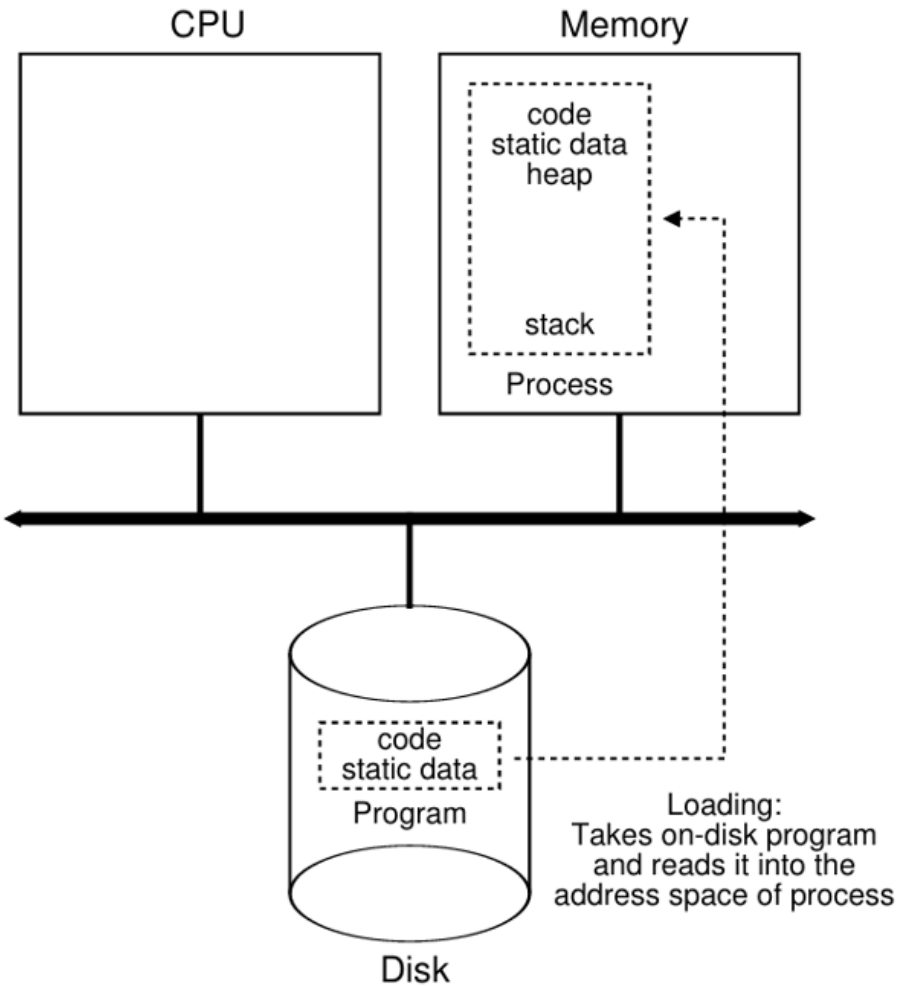
Proces je entitet koji se može dodeliti procesoru i u njemu izvršavati.

Definicija (4)

Proces je jednica aktivnosti koju karakteriše izvršavanje niza instrukcija, tekuće stanje i dodeljeni set sistemskih resursa.

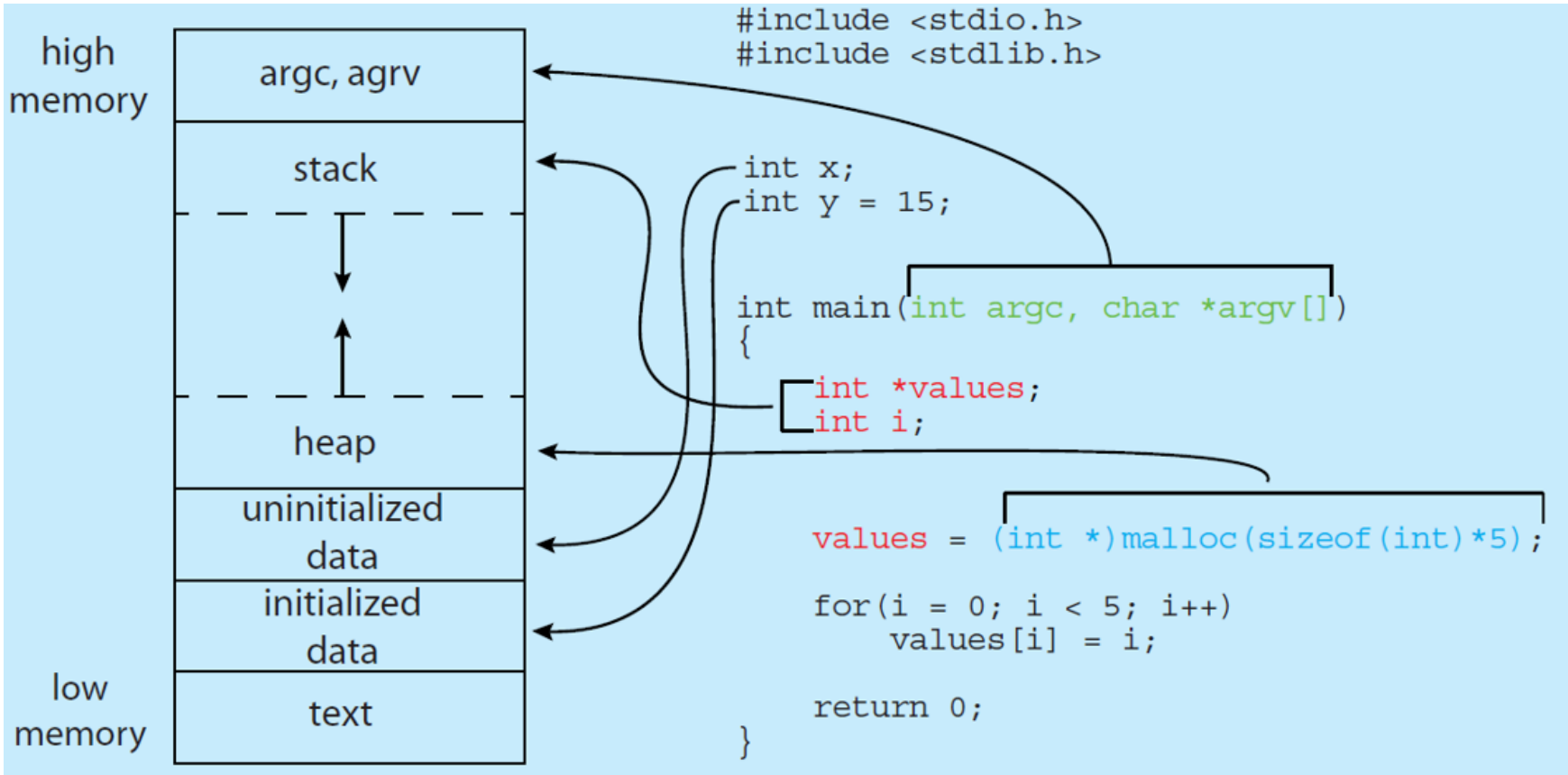
Program ⇒ Proces

Od programa do procesa



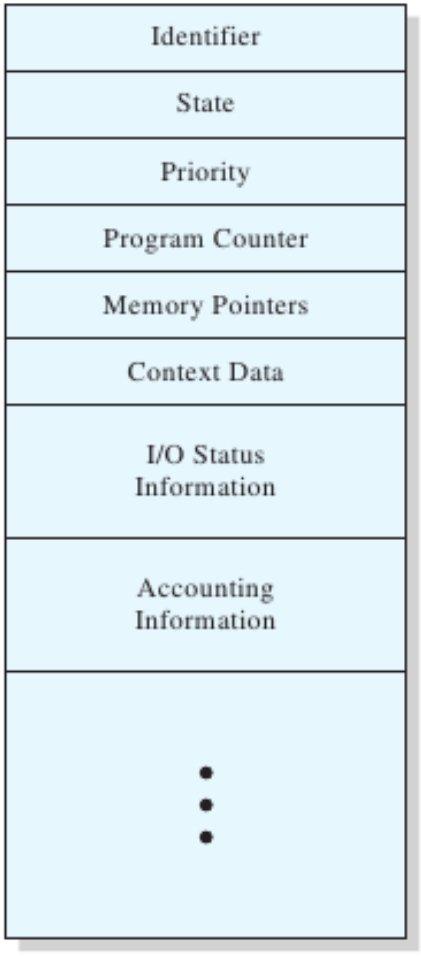
Memorijska slika procesa

Primer



Struktura upravljačkog bloka procesa

Pored kôda i skupa podataka, proces poseduje i upravljački blok



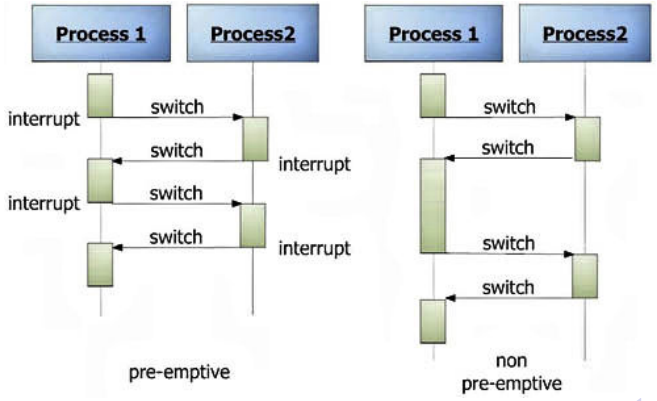
Stanja procesa

Trag (*Trace*) procesa

Ponašanje procesa može se predstaviti **sekvencom instrukcija** koje se izvršavaju za taj konkretan proces.

Dispečer (*Dispatcher*)

Dispečer je relativno mali prgram koji je deo OS-a i čiji je zadatak da prebacuje procesor sa procesa na proces. Dispečer može biti **neprekidni (non-preemptive)** i **prekidni (preemptive)**.

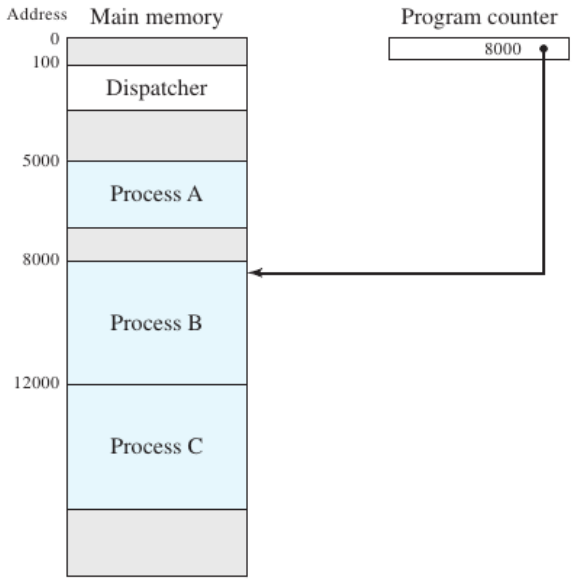


Stanja procesa

Primer

Pretpostavke

Sistem nema virtuelnu memoriju. Dispečer dozvoljava najviše 6 sukcesivnih instrukcija po procesu.



(a) Trace of Process A	(b) Trace of Process B	(c) Trace of Process C
5000	8000	12000
5001	8001	12001
5002	8002	12002
5003	8003	12003
5004		12004
5005		12005
5006		12006
5007		12007
5008		12008
5009		12009
5010		12010
5011		12011

5000 = Starting address of program of Process A
8000 = Starting address of program of Process B
12000 = Starting address of program of Process C

Stanja procesa

Primer

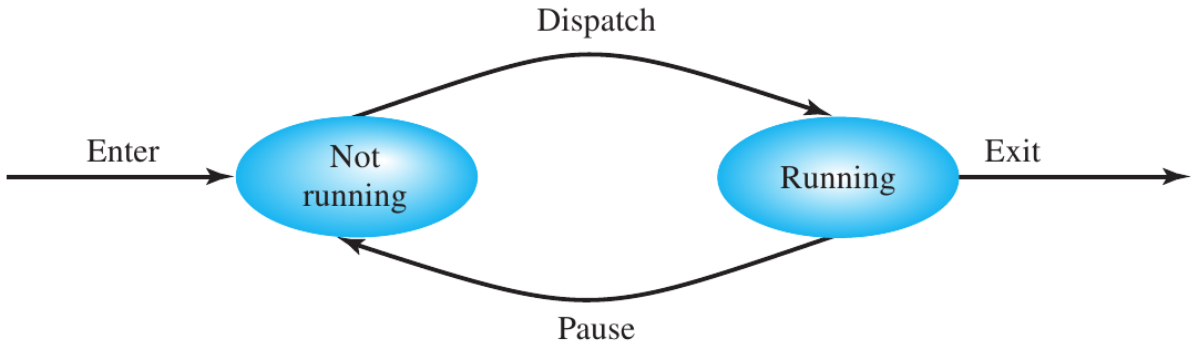
1	5000	27	12004
2	5001	28	12005
3	5002	----------	Timeout
4	5003	29	100
5	5004	30	101
6	5005	31	102
----------	Timeout	32	103
7	100	33	104
8	101	34	105
9	102	35	5006
10	103	36	5007
11	104	37	5008
12	105	38	5009
13	8000	39	5010
14	8001	40	5011
15	8002	----------	Timeout
16	8003	41	100
----------	I/O Request	42	101
17	100	43	102
18	101	44	103
19	102	45	104
20	103	46	105
21	104	47	12006
22	105	48	12007
23	12000	49	12008
24	12001	50	12009
25	12002	51	12010
26	12003	52	12011
		----------	Timeout

100 = Starting address of dispatcher program

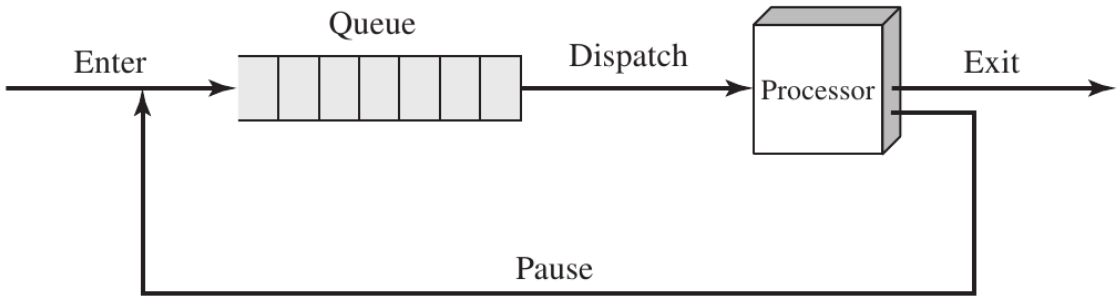
Shaded areas indicate execution of dispatcher process;
first and third columns count instruction cycles;
second and fourth columns show address of instruction being executed

Stanja procesa

Primer tranzicije stanja

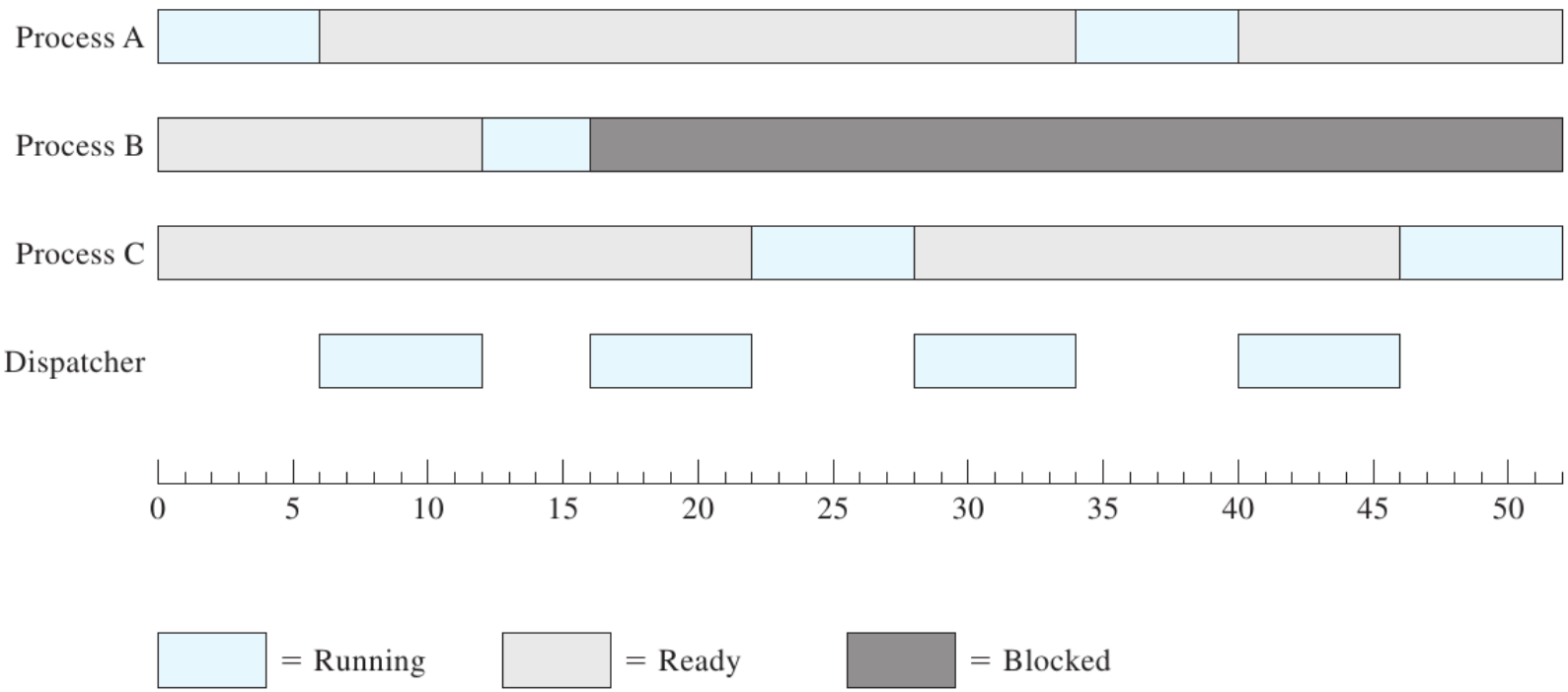


(a) State transition diagram



(b) Queuing diagram

Model procesa sa dva stanja



Kreiranje procesa

Razlozi za kreiranje procesa

1. **Novi paketni (batch) posao** - OS-u se podnosi zahtev za novi paketni posao
2. **Interaktivni login** - otvara se sesija za korisnika
3. **Kreira se od strane OS-a da bi obezbedio uslugu** - recimo print server
4. **Kreira se od strane postojećeg procesa** - *spawn*

Parent-child procesi

Kada jedan proces kreira drugi, stvara se odnos **roditelj-potomak** ili **parent-child** između njih. Uobičajeno je da oni komuniciraju.

Terminacija procesa

Razlozi za terminaciju procesa

1. Normalna terminacija
2. Prekoračenje vremenskog limita (*CPU time* ili *Wall Clock time*)
3. Nedostupnost memorije
4. Kršenje granica memorije
5. Greška zaštite
6. Aritmetička greška
7. Prekoračenje vremena (proces čekao predugo na neki resurs)
8. Otkaz U/I
9. Pogrešna instrukcija
10. Privilegovana instrukcija (proces pokušao da koristi kernel mode instrukciju)
11. Pogrešna upotreba podataka
12. Intervencija operatora ili OS-a
13. Terminacija procesa roditelja
14. Zahtev procesa roditelja

Model procesa sa pet stanja

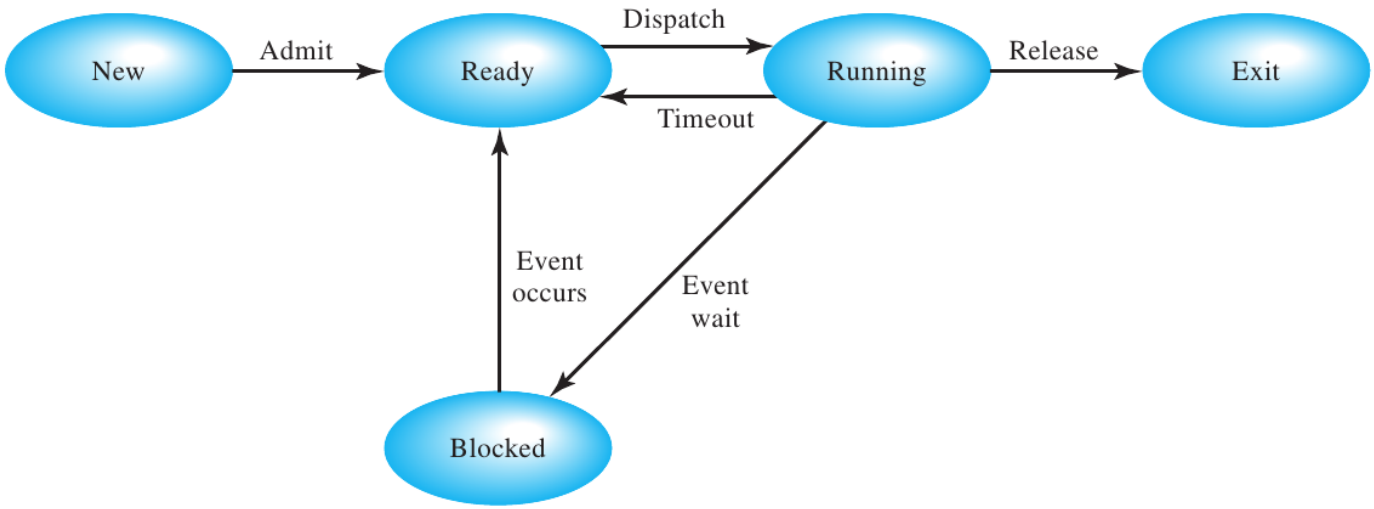
Pretpostavka: Koristi se *Round-robin* tehnika kružnog dodeljivanja

Problem

Ako postoji samo jedan red čekanja za sve procese, **dispečer bi svaki put morao da skenira taj red kako bi utvrdio koji proces nije blokiran i koji je u redu bio najduže**.

Rešenje

Stanje **Not-running** podeliti na dva stanja: **Ready** i **Blocked**.



Model procesa sa pet stanja

Značenje pojedinačnih stanja

1. **Running** - Proces koji se trenutno izvršava
2. **Ready** - Proces koji je spreman za izvršavanje kada dobije procesor na korišćenje
3. **Blocked** - Proces koji ne može da nastavi sa izvršavanjem dok se ne pojavi neki događaj, npr. kraj U/I operacije
4. **New** - Proces koji je upravo kreiran, ali ga OS još nije stavio na red spremnih. Obično su to procesi kojima je napravljen upravljački blok, ali još uvek nisu učitani u memoriju.
5. **Exit** - Proces koji je na normalan način prekinuo s radom ili ga je OS skinuo sa reda iz nekog drugog razloga.

Tranzicije stanja

$Noproc \rightarrow New$, $New \rightarrow Ready$, $Ready \rightarrow Running$, $Running \rightarrow Exit$, $Running \rightarrow Ready$, $Running \rightarrow Blocked$, $Blocked \rightarrow Ready$, $Ready \rightarrow Exit$, $Blocked \rightarrow Exit$

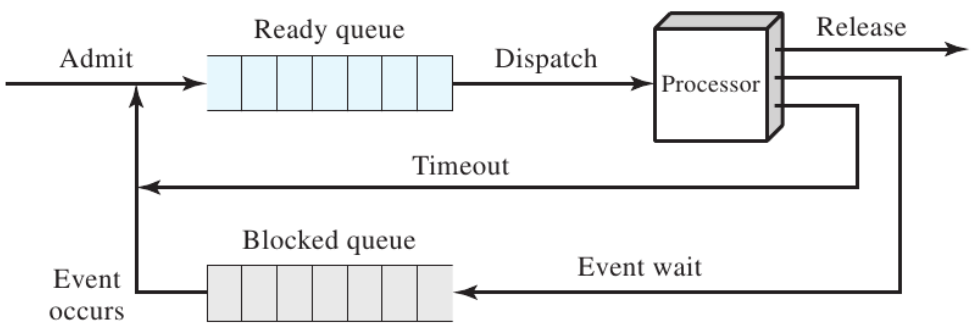
Da li je jedan red za blokirane procese dovoljan?

Problem

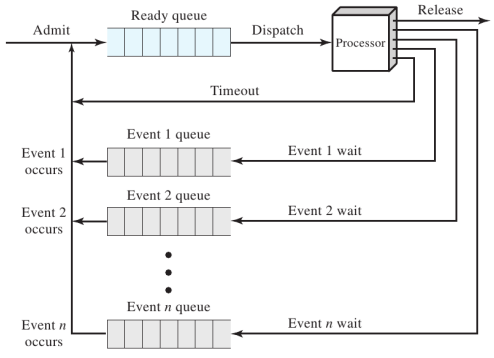
Kada svi procesi koji se nalaze u stanju **Blocked** čekaju u istom redu, u trenutku kada se pojavi bilo koji događaj (recimo U/I) dispečer mora da skenira ceo red kako bi utvrdio koji od blokiranih procesa čeka baš taj događaj.

Rešenje

Uvesti više redova blokiranih procesa za svaki tip događaja posebno.



(a) Single blocked queue



(b) Multiple blocked queues

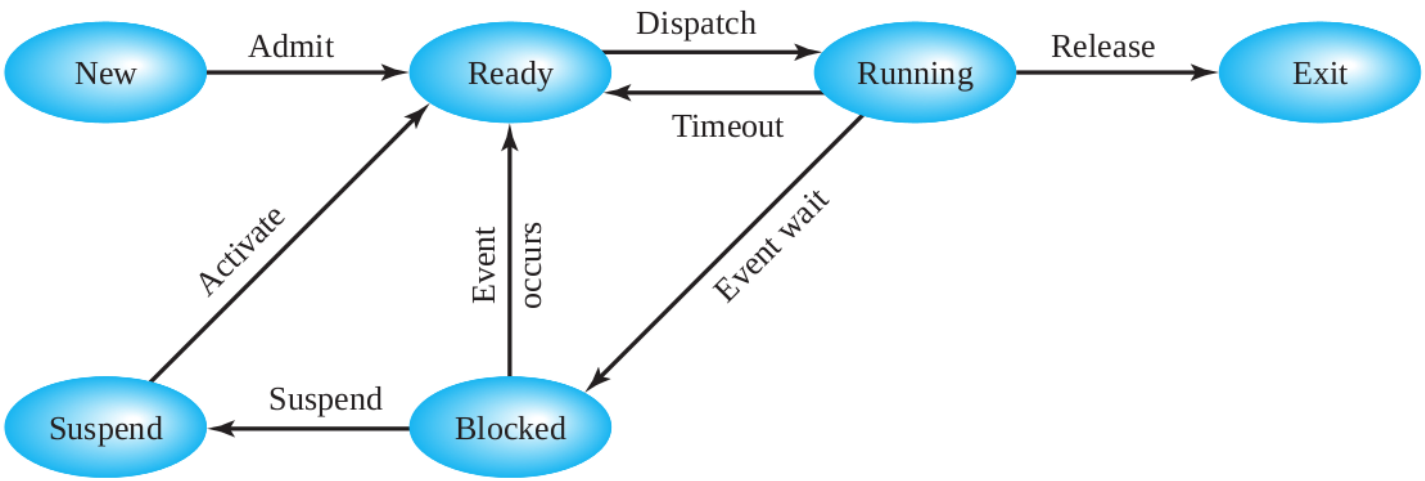
Suspendovani procesi

Problem

Ukoliko više procesa mora da čeka spori U/I, procesor u sistemu bez virtuelne memorije i dalje može veći deo vremena da provede besposlen.

Rešenje

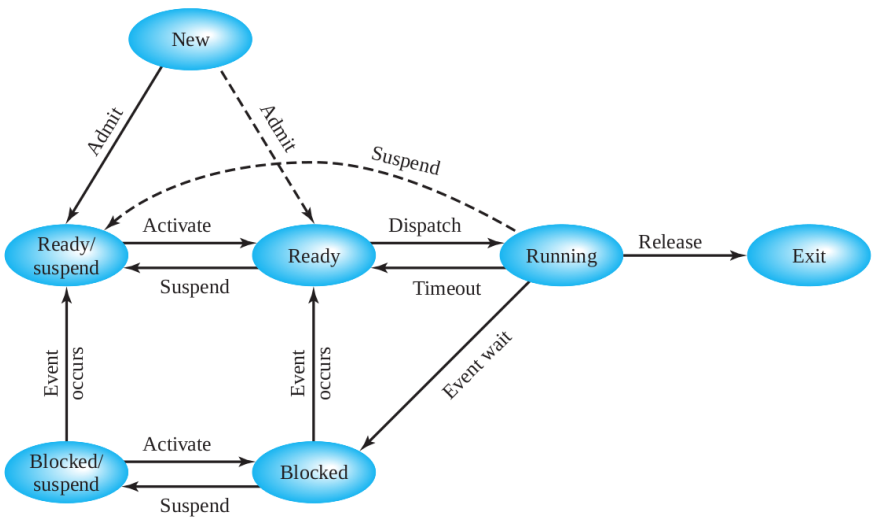
Ako nijedan proces u RAM-u nije u stanju **Ready**, OS prebacuje jedan od procesa na disk u red **Suspended** procesa. Tada OS može da preuzme u RAM drugi proces iz reda suspendovanih ili prima zahtev za novim procesom.



Model sa dva suspendovana stanja

Stanja procesa

1. **Ready** - Proces je u RAM-u i raspoloživ za izvršavanje
2. **Blocked** - Proces je u RAM-u i čeka na događaj
3. **Blocked/Suspended** - Proces je na disku i čeka na događaj
4. **Ready/Suspended** - Proces je na disku ali će biti raspoloživ za izvršavanje čim dođe u RAM



Ostali razlozi za suspenziju procesa

Generalizacija pojma suspenzije procesa

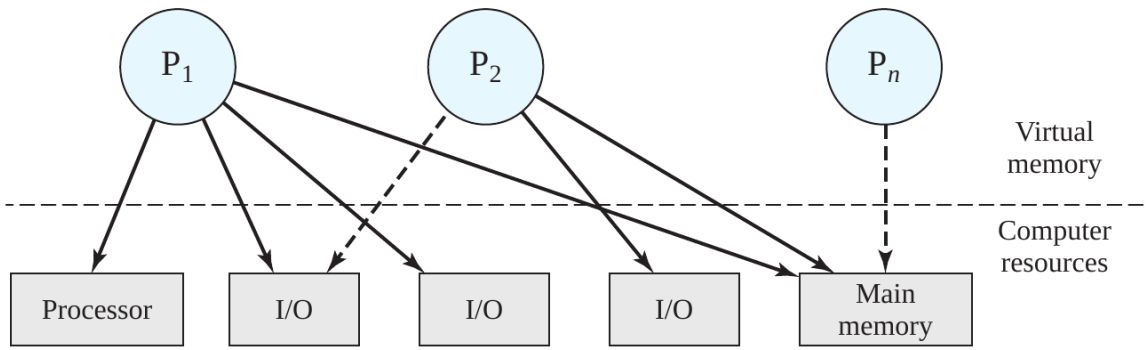
Suspenzija procesa se kao pojam može generalizovati i nastaje usled jednog od razloga navedenih dole.

1. **Razmenjivanje** - *Swapping, Paging*
2. **Drugi razlozi operativnog sistema** - recimo suspenzija procesa za koji se sumnja da je malicioznog karaktera
3. **Interaktivni zahtev korisnika** - na Linuxu `kill -STOP 945` `kill -CONT 945`
4. **Vremenska sinhronizacija** - Periodična suspenzija/nastavak rada
5. **Zahtev procesa roditelja**

Opis procesa

Uloga operativnog sistema

Operativni sistem možemo posmatrati kao entitet koji upravlja upotrebom sistemskih resursa od strane procesa.

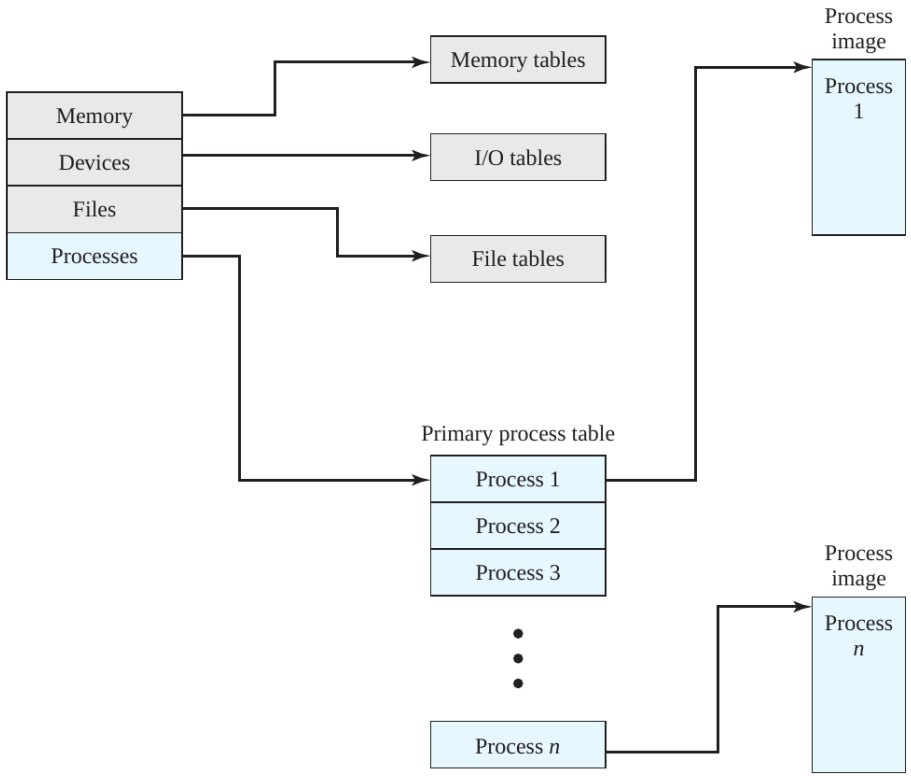


Slika: Na slici proces P_1 koristi jedini procesor, ima pod kontrolom 2 U/I resursa i bar jedan njegov deo je u RAM-u. Proces P_2 čeka U/I uređaj koji je zauzeo P_1. Proces P_n je razmenjen (*swap*-ovan) i prema tome suspendovan.

Upravljačke strukture operativnog sistema

Tabele OS-a

Opearativni sistem mora da održava informacije o svim resursima kojima upravlja. Za ovu svrhu se koriste se koriste **tabele operativnog sistema**.



Upravljačke strukture procesa

Elementi *Process Control Block* - PCB

Elementi slike procesa su: **korisnički podaci, korisnički program, sistemski stek i upravljački blok (PCB)**.

Identifikacija procesa

- Identifikator procesa
- Identifikator procesa roditelja
- Korisnički identifikator

Informacije o stanju procesa

- Registri vidljivi korisniku
- Upravljački i statusni registri (PSW, PC, ...)
- Pokazivači steka - jedan ili više dodeljenih sistemskih stekova

Upravljačke strukture procesa - nastavak

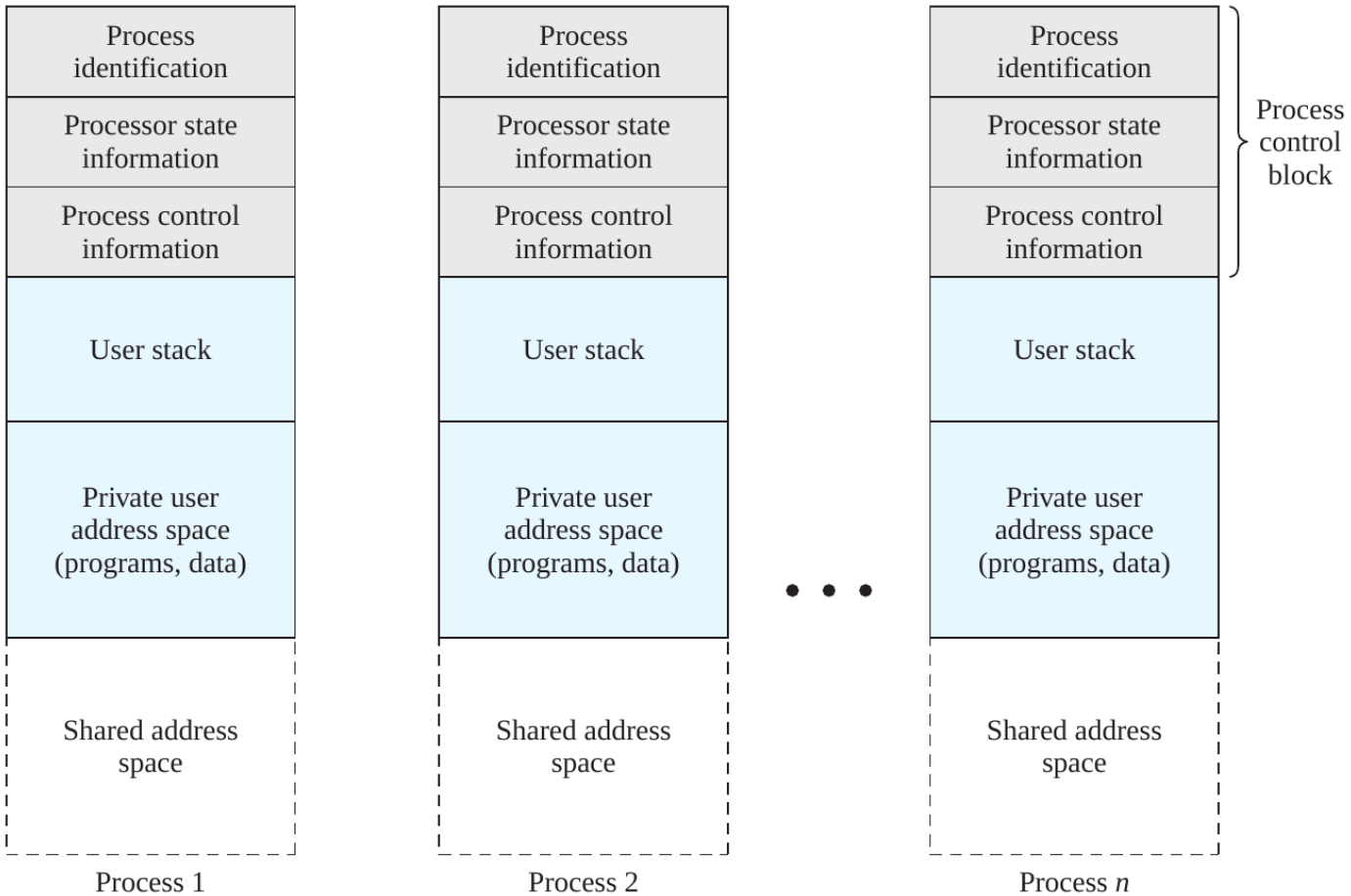
Elementi *Process Control Block* - PCB

Informacije za upravljanje procesom

- **Raspoređivanje i informacije o stanju**
  - Stanje procesa (*Ready*, *Blocked*, ...)
  - Prioritet
  - Informacije vezane za raspoređivanje - recimo broj do sada izvedenih instrukcija
  - Doagađaj - Identitet događaja koji proces čeka da bi nastavio s radom
- **Strukturiranje podataka** - recimo pointerska veza sa sledećim procesom u listi
- **Međuprocesna komunikacija** - markeri, signali i poruke
- **Privilegije procesa**
- **Upravljanje memorijom** - pokazivači na delove tabele stranica virtuelne memorije
- **Vlasništvo nad resursima i iskorišćenje** - npr. otvoreni fajlovi

Procesi u virtuelnoj memoriji, struktura liste procesa

Dijagram



Uobičajene funkcije kernela operativnog sistema

- **Upravljanje procesom**
  - kreiranje i prekid procesa
  - raspoređivanje i slanje procesa
  - komutiranje procesa
  - sinhronizacija i podrška komunikacije među procesima
  - upravljanje PCB-om
- **Upravljanje memorijom**
  - dodela adresnog prostora procesu
  - razmenjivanje (*swapping*)
  - upravljanje stranicama i segmentima
- **Upravljanje UI-jem**
  - upravljanje baferima
  - dodela UI kanala i uređaja procesima
- **Funkcije za podršku**
  - obrada prekida
  - izračunavanje
  - nadgledanje

Kreiranje procesa

1. Novom procesu dodeljuje se **jedinstveni identifikator**
2. Procesu se **dodeljuje prostor** - za elemente slike procesa
3. **Inicijalizuje se PCB** - ulgavnom nule, osim PC koji se setuje na početnu adresu tekst segmenta
4. Postavljaju se **odgovarajuće veze**- npr. veze u redu *Ready*
5. **Kreiranje ili proširivanje ostalih struktura** - npr. obračunske informacije

Komutiranje (zamena) procesa

Pitanje: Kada se komutira proces?

Tabela: Mehanizmi koji prekidaju izvršenje procesa

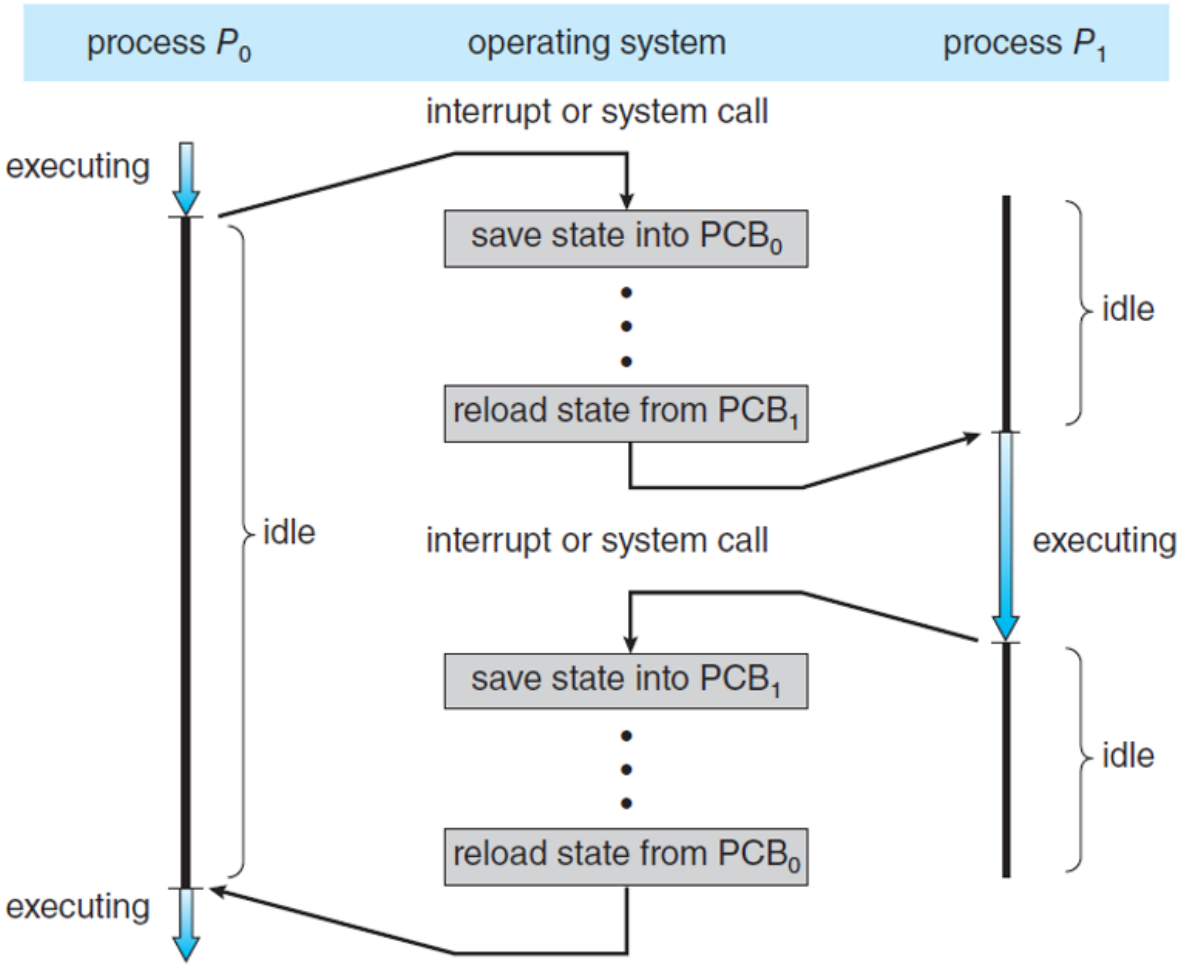
Mehanizam	**Uzrok**	**Upotreba**
Prekid (*Interrupt*)	Eksterni u odnosu na tekuću instrukciju	Reakcija na asinhroni spoljni događaj
Zamka (*Trap*)	Pridružen izvršenju tekuće instrukcije	Rukovanje greškom ili izuzetkom
Poziv supervizora	Eksplicitni zahtev	Poziv funkcije operativnog sistema

Primer

Odrediti kom tipu pripadaju sledeći mehanizmi: Prekid sistemskog sata, U/I prekid, zahev procesa za ptvaranjem fajla, deljenje nulom.

Komutiranje (zamena) procesa

Context switch



Komutiranje moda vs. promena stanja procesa

Komutiranje moda

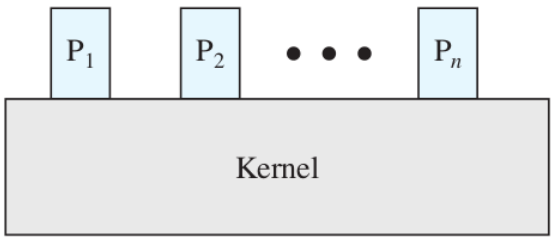
1. PC registar se setuje na početnu adresu rukovaoca prekidom (*interrupt handler*)
2. Procesor se prebacuje iz korisničkom moda u kernel mod, pošto rutina za obradu prekida može sadržati privilegovane instrukcije

Promena stanja procesa

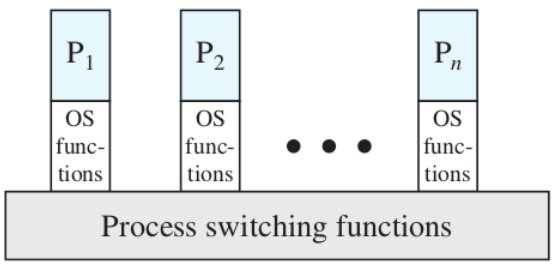
1. Sačuvati kontekst proceora, PC, PSW i druge registre
2. Ažurirati PCB procesa koji je trenutno u *Running* stanju. Šta se sve menja?
3. Pomeranje PCB-a u odgovarajući red (Ready, *Blocked on Event i*, ...)
4. Odabir nekog drugog procesa za izvršavanje
5. Ažuriranje PCB-a odabranog procesa
6. Ažuriranje memorijskih struktura
7. Učitati stanje procesora koje je bilo aktuelno kada je poslednji put radio

Izvršavanje funkcija operativnog sistema

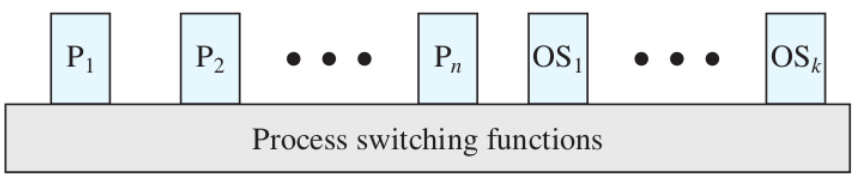
Gde se izvršava kôd operativnog sistema



(a) Separate kernel



(b) OS functions execute within user processes



(c) OS functions execute as separate processes

