

Operacinės sistemas

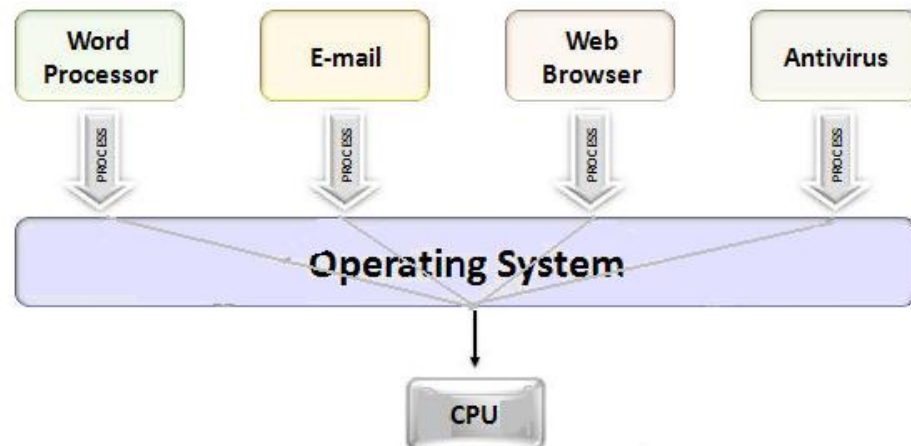
N. Sarafinienė
2013m.



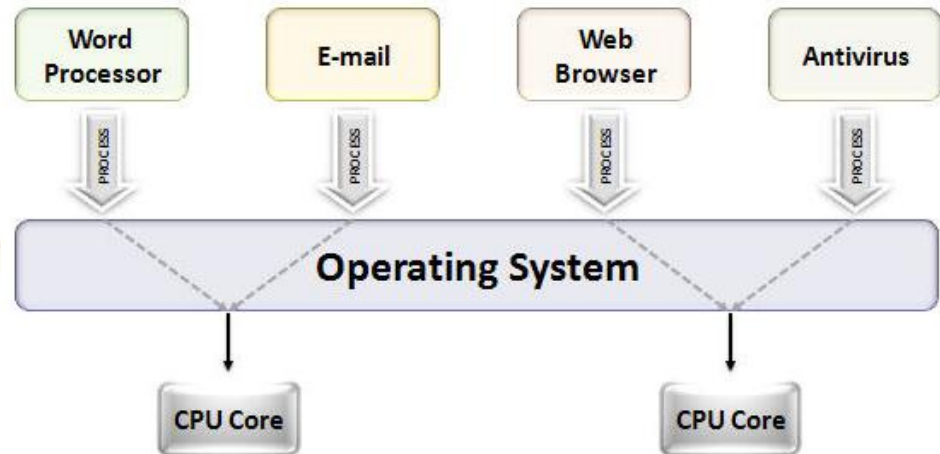
Kalbėsime

- Apibrėšime procesą
- Proceso kontekstą
- Proceso būvius
- Proceso vykdymo modas
- Persijungimus tarp procesų
- Operacinės sistemos vykdymą

Daugiaprogramis (Multiprogramming) darbo režimas



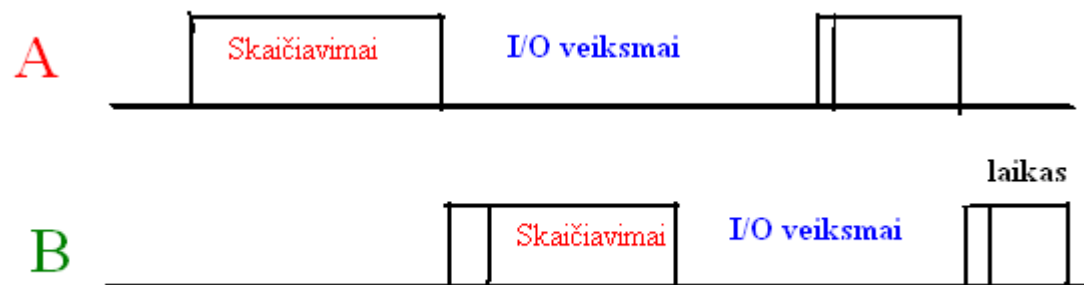
Daugiaprocesorinis darbo režimas



Daugiaprocesis (Multitasking) darbo režimas

- Būtų labai neefektyvu, jei vienas procesas naudotų procesorių nuo jo vykdymo pradžios iki pabaigos. Tai susiję:
 - Laukimo laikais (I/O) – CPU prastovos, ilgi vykdymo laikai
 - Pertraukčių apdorojimas
- Siekiant efektyviai padidinti sistemos pralaidumą dauguma modernių OS leidžia **daugeliui procesų būti vienu metu aktyviais** (pasiruošusiais vykdymui).
 - Procesorius priima juos apdorojimui paeiliui.
 - CPU greitis yra žymiai didesnis nei I/O įrenginių.
 - Gaunamas trumpesnis vykdymo laikas
 - Pertraukimų apdorojimas nesukelia problemų
- Toks darbo režimas vadinamas **daugiaprocesiu** (Multitasking) arba **daugiaprograminiu** (Multiprogramming) darbo režimu.
 - Persijungimas tarp procesų yra planuojamas bei valdomas procesų planuotojo (scheduler).
 - Kai kuriose operacinėse sistemose procesai turi patys “geranoriškai” grąžinti CPU po tam tikro laiko intervalo
 - Dauguma OS atima iš proceso CPU praėjus laiko kvantui.

Daugiaprocesis darbas



Daugiaprocesio darbo režimo OS keliami reikalavimai

- *Sugebėti “vienu metu” vykdyti daug procesų siekiant maksimizuoti procesoriaus panaudojimą bei užtikrinti priimtinus atsakymo laikus.*
- Vykdydama daug procesų vienu metu OS turi:
 - ☐ Skirti resursus procesams
 - ☐ Palaikyti komunikacijas tarp procesų
 - ☐ Palaikyti procesų sinchronizaciją
 - ☐ Leisti vartotojams kurti procesus

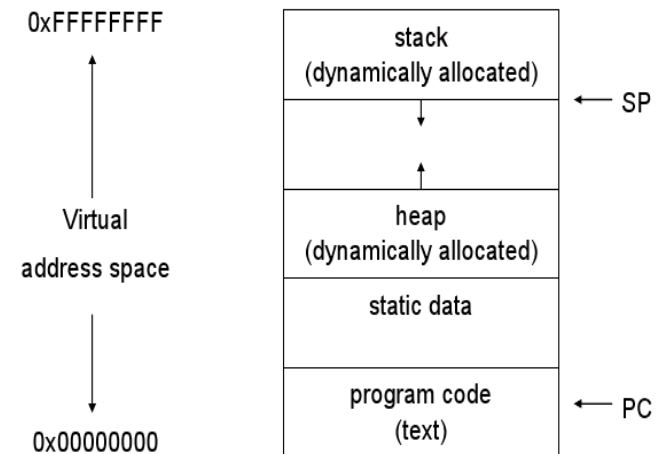
Procesas ir programa

- Procesas – tai programa jos vykdymo metu.
 - Programa – tai statinis failas (pasyvus elementas)
 - Procesas = vykdoma programa = programa + vykdymo būvis (aktyvus elementas).
- Procesas tai bazinis vienetas, kurio atžvilgiu OS skiria resursus (jo atžvilgiu gali būti vykdomas ir CPU laiko planavimas)
 - Procesai gauna vietą atmintyje, jiems skiriami įrenginiai ...
- Skirtingi procesai gali vykdyti tą pačią programą ir skirtingame etape
 - Pav.: vienu metu gali būt kompiliuojamos kelios programos `***.c`

Procesas

- Tai vykdoma programa
- Proceso vykdymui reikia šių minimalių resursų:
 - Atmintinės srities programos kodo bei duomenų saugojimui
 - Grupės CPU registrų proceso vykdymo palaikymui
- Procesas turi savo adresų sritį, kurioje yra:
 - Programos kodo sritis
 - Saugomas procesoriaus vykdomas kodas
 - Duomenų sritis
 - Saugomi duomenys
 - Steko sritis
 - Saugoma informacija susijusi su perėjimais prie procedūrų

Processes – Address Space



Pavyzdys

Programa hello.c

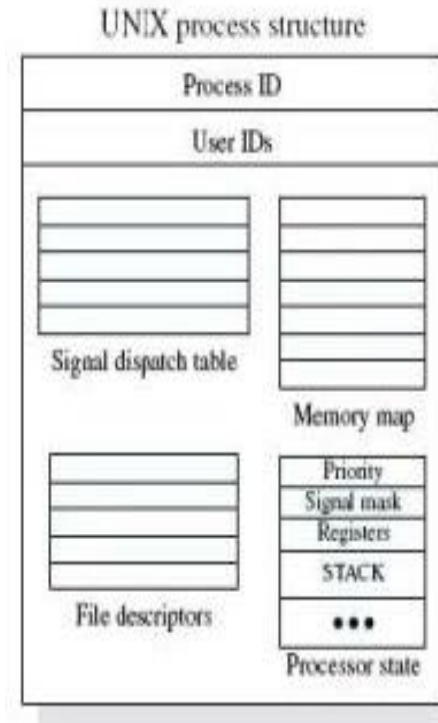
```
int a = 5; // Saugoma duomenų sekcijoje
int b[20]; // Saugoma duomenų sekcijoje
int main() { // Saugoma kodo sekcijoje
    int x; // Saugoma steke
    int *p =(int*) malloc(sizeof(int)); // heap
}
```

Kelių procesų vykdymas

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

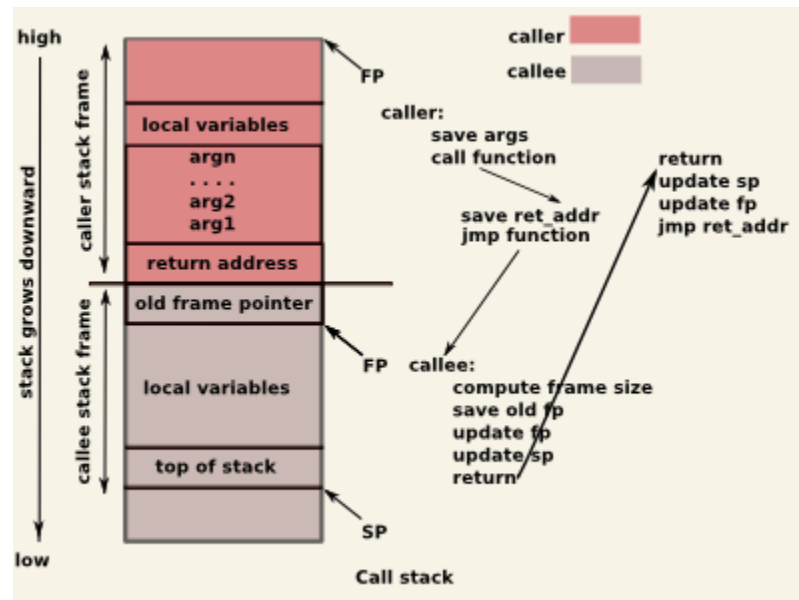
Proceso vykdymo priežiūra

- Procesas vykdo tam tikrą kodą.
 - OS privalo sekti programos vykdymo eigą (PC).
- OS turi sekti steko rodyklę.
- OS saugo proceso būvio informaciją **proceso kontrolės bloke** (PCB).
 - Proceso ID, PC,...
- Programos komandos, duomenys stekas & plėtimosi sritis yra procesui skirtose atmintinėse
- **Metaduomenys** yra PCB (tai yra branduolio duomenų struktūra)

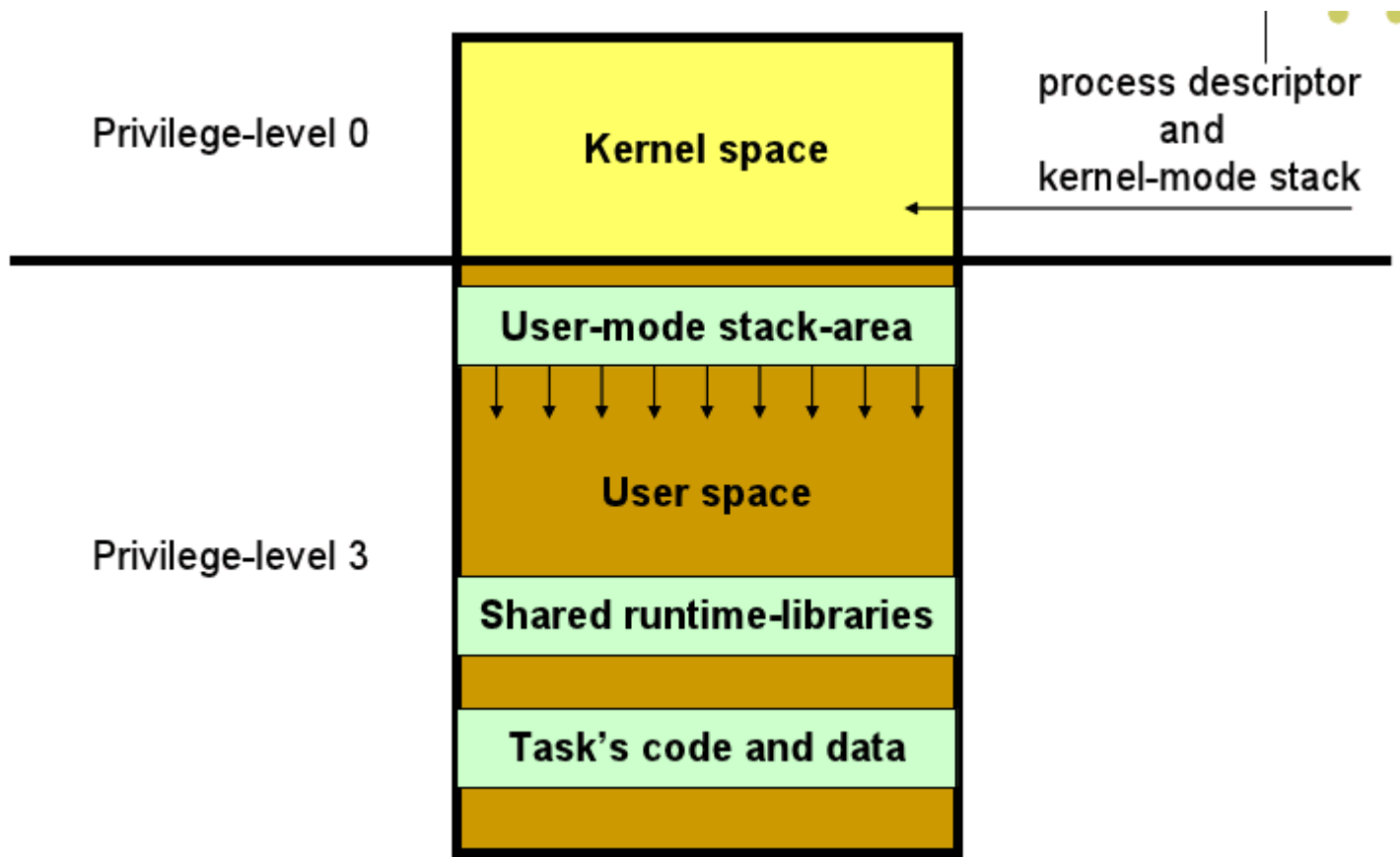


Steko paskirtis

- Atmintis, naudojama parametų perdavimui procedūroms.
-
- Atmintis naudojama priskiriant vietą lokaliems kintamiesiems
-
- Naudojama išsaugojimui grįžimo adresų.
-
- Naudojama išsaugojimui registrų turinių.



Virtuali processo erdvè



Proceso kontrolēs blokas (PCB)

proceso būvis	sekantis einantis prieš
Proceso ID (PID)	
programos skaitliukas	
registrai	
Atmintinės struktūra	
Atvertų failų lentelė	
Ir t.t.	

Sukuria ir valdo
operacinę
sistema.

PCB yra didelė
duomenų struktūra

Linux sistemoj tai
būtų *task_struct*
(apie 70 laukų)

Windows XP ---
EPROCESS --- apie 60
laukų

PCB

■ Proceso identifikacija

- identifikatoriai
 - skaitmeniniai
 - PID
 - PPID
 - UID (vartotojo ID)

■ Procesoriaus būvio informacija

- Prieinamų vartotojui registrų turiniai
- Kontrolės ir būvio registrai:
 - *PC (Program counter)*
 - *Būklės kodai*
 - *Būklės informacija* : (vykdymo moda, pertraukčių galimumas)
- Steko rodyklė

PCB

- **Proceso kontrolės informacija**

- **Planavimo ir būklės informacija**

- *Proceso būvis*: (pav., vykdomas, pasiruošęs, sustabdytas).
 - *Prioritetas*: (gali būt nusakomas keliomis reikšmėmis)
 - *Su planavimu surišta informacija*: (pav. kiek procesas laukė, kiek gavo cpu laiko,...).
 - *Laukiamas įvykis*

- **Surišimas su kitomis duomenų struktūromis**

- *Su to paties prioriteto procesais toje pačioje eilėje*
 - *Su procesais, kuriuos sieja T_V santykiai*
 - *Su procesais laukiančiais to paties įvykio*

PCB

- **Proceso kontrolės informacija**
 - Tarp-procesinės komunikacijos informacija
 - *Įvairios vėliavėlės, signalai, pranešimai*
 - Proceso privilegijos
 - *Privilegijos naudotis atmintine, tam tikromis komandomis ar paslaugomis.*
 - Atmintinės valdymas
 - *Nuorodos į segmentų arba puslapių lenteles.*
 - Resursų valdymas ir panaudojimas
 - *G.B nurodomi proceso kontroliuojami resursai – atverti failai. Resursų panaudojimas planavimo tikslais.*

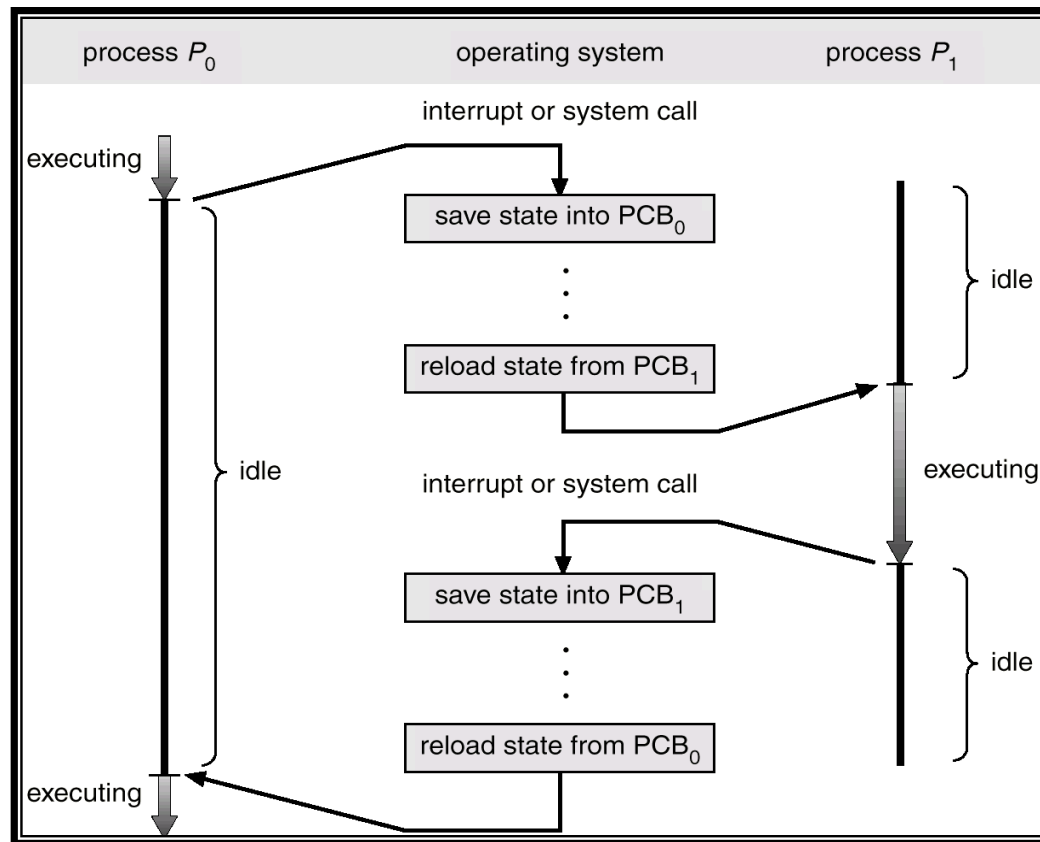
Procesų perjungimas

- Pertraukimas pagal laiką
 - *Baigėsi kvantas*
- I/O pertraukimas
- Klaida susijusi su kreipiniu į atmintinę
 - *Puslapio nėra pagrindinėje atmintinėje*
- Įvykus klaidai
 - *Vykdamas procesą gauta klaidos situacija (procesą baigt?)*

Proceso perjungimas

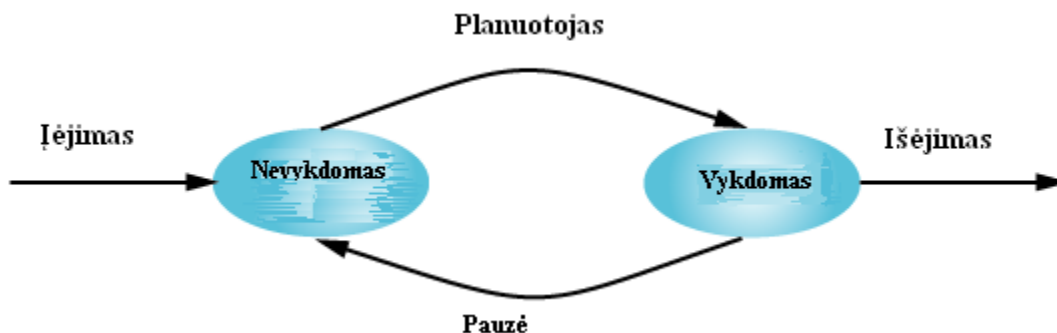
- Valdo operacinė sistema. Šis perėjimas vadinamas **konteksto perjungimu**
- Stabdomas vykdomas procesas ir pradedamas vykdyti vienas iš pasiruošusių vykdymui procesų.
 - *prieš tai vykdyto proceso kontekstas yra išsaugomas to proceso kontrolės bloke (PCB)*
 - *Vykdymui parenkamas naujas procesas*
 - *įkraunamas naujo proceso kontekstas (iš jo PCB bloko). (keičiamas jo būvis į vykdomą)*

Procesų perjungimas

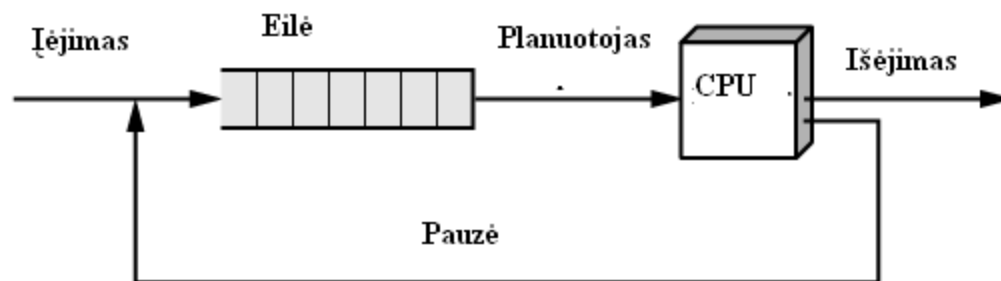


Dviejū būvių procesų diagrama

- Procesas gali būti viename iš dviejų būvių
 - Vykdomas
 - Nevykdomas



Procesų eilė

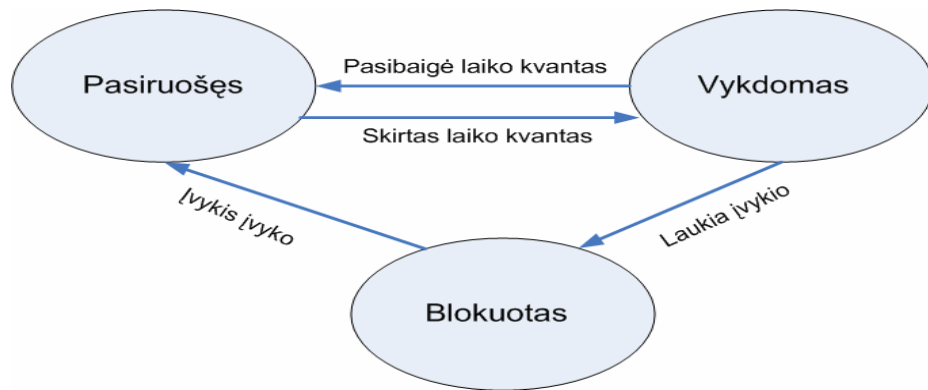


Pagrindiniai proceso būviai

3-jų būvių diagrama

Nevykdomas procesas :

- *Pasiruošęs vykdymui*
- *Blokuotas*



Proceso sukūrimas

- *Paketinio* aptarnavimo sistemose *atsiranda kai sistemai pateikiama nauja paketinė užduotis*
- *Interaktyviose* sistemose naujas procesas *atsiranda prisijungus naujam vartotojui.*
 - Naują procesą gali sukurti ir **operacinė sistema** tam tikros vartotojo užduoties atlikimui.
 - Naujus procesus gali sukurti jau egzistuojantys sistemos **procesai**

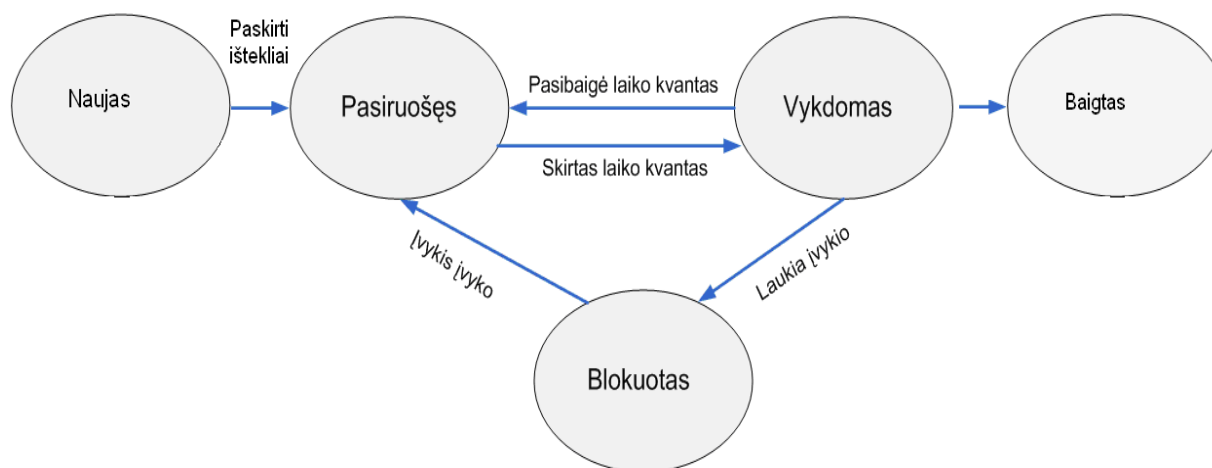
Proceso sukūrimas

- Inicializuojamas proceso kontrolės blokas
- Priskiriamas unikalus proceso identifikatorius. Sukuriami atitinkami ryšiai
 - *Pav. ryšiai į kitus naujus procesus*
- Sukuriamos kitos reikalingos duomenų struktūros
 - *Pav. apskaitos informacijai, planuotojui*

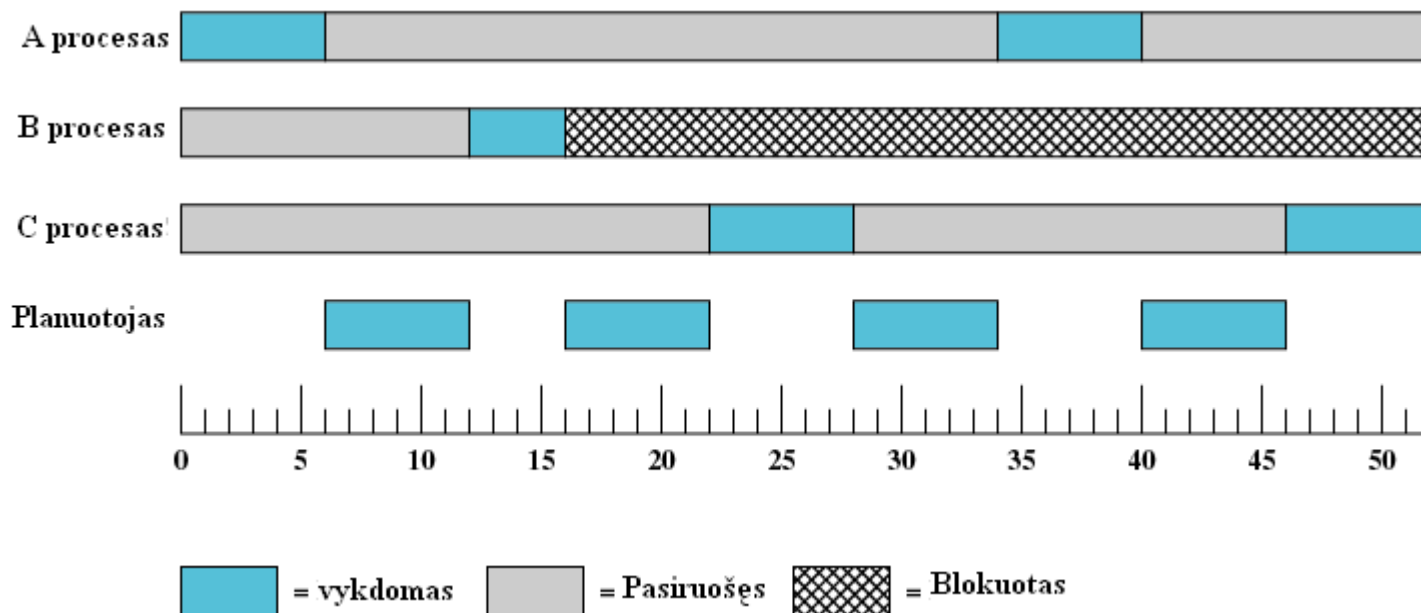
Proceso pabaiga

- Atsijungus vartotojui
- Normali veiksmų pabaiga.
- Viršyta laiko riba
- Neprieinama atmintinė
- Atmintinės ribų pažeidimas.
- Prieigos klaida
- Aritmetinė klaida , I/O klaida, netinkama komanda
- Privilegiuota komanda.
- Procesas gali būti užbaigiamas įsikišus operacinei sistemai, pavyzdžiui, iškilus mirties taško situacijai sistemoje.
- Procesas gali būti nutraukiamas ir tuo atveju, jei tą proceso nutraukimą inicijuoja to proceso „tėvo“ procesas.

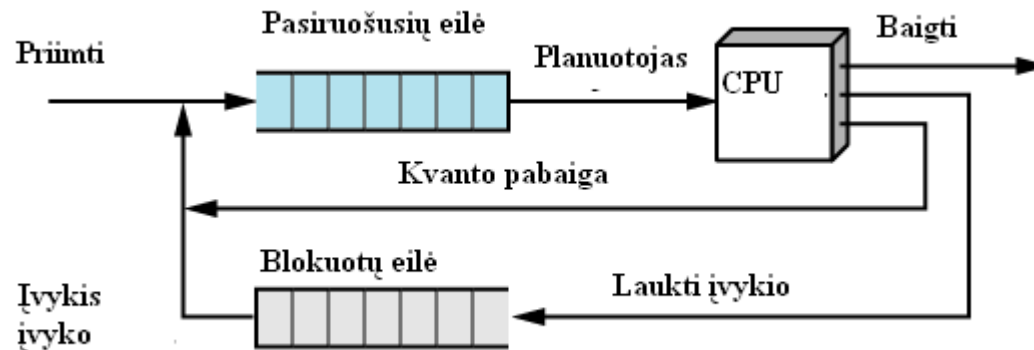
5-ių būvių diagrama



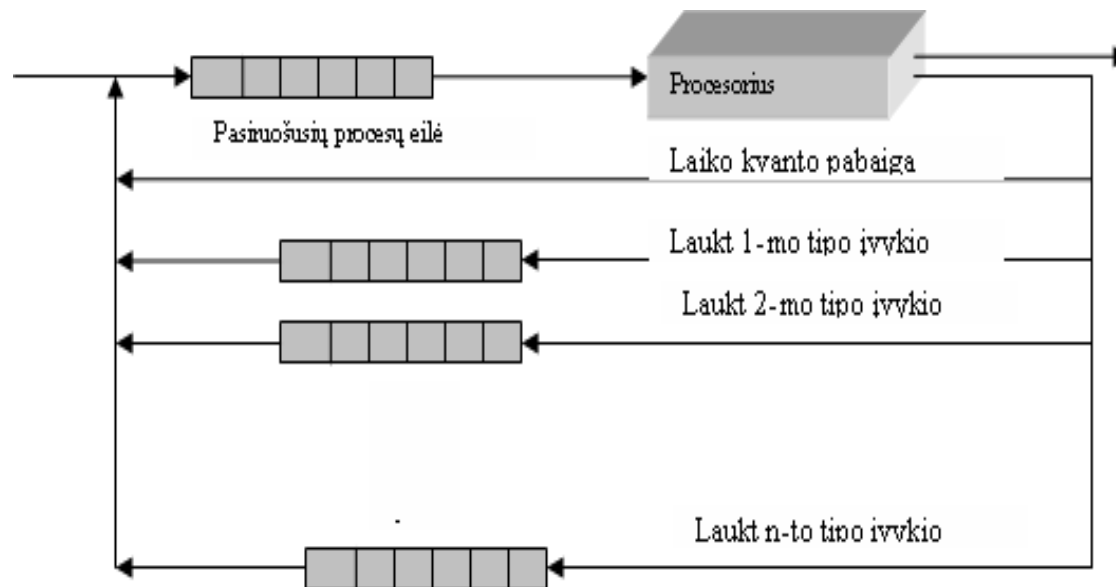
Procesų vykdymas



Eilių tvarkymas

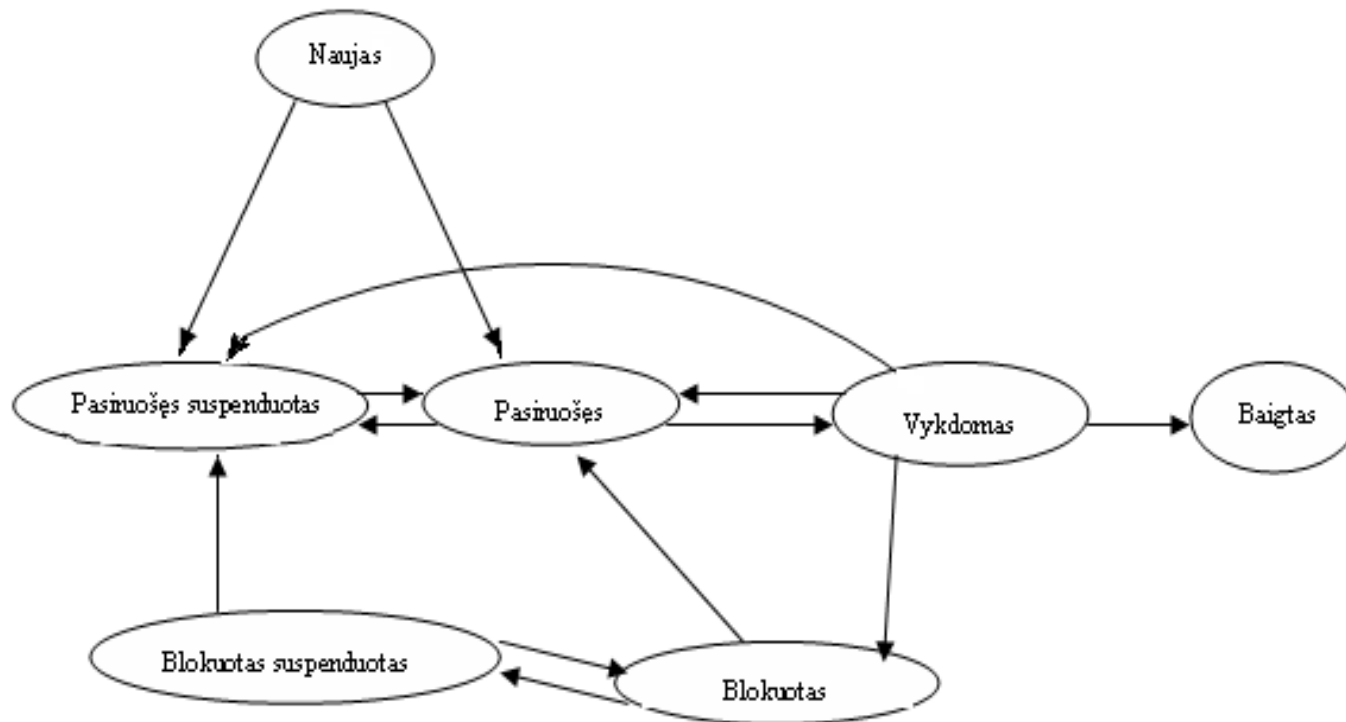


Eilių tvarkymas

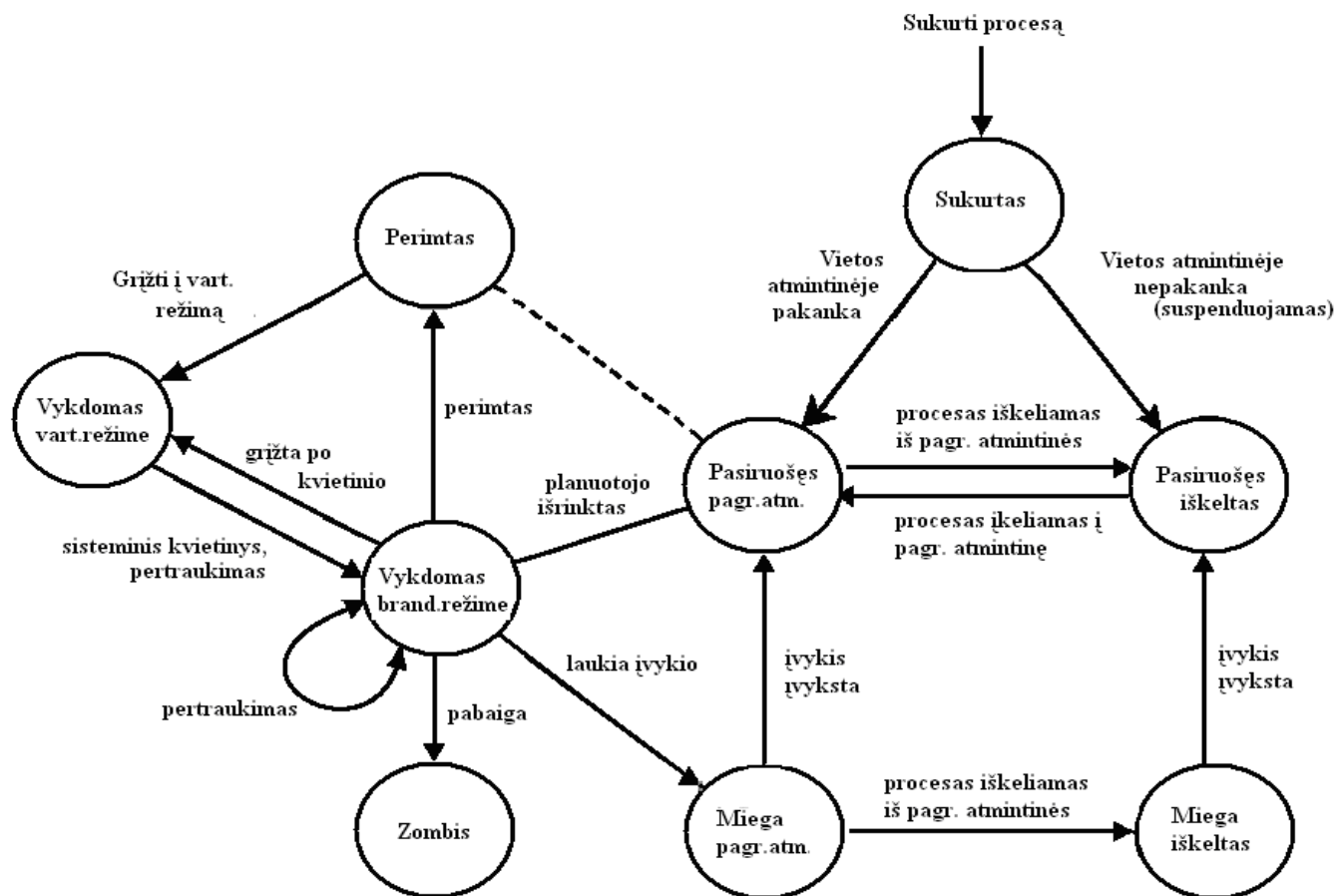


Suspenduoti procesai

- du papildomi būviai: „**Blokuotas-suspenduotas**“ ir „**Pasiruošęs-suspenduotas**“.
- Būvyje „**Blokuotas-suspenduotas**“ bus blokuoti procesai, kurie yra iškelti į diską
- Būvyje „**Pasiruošęs-suspenduotas**“ - procesai, esantys pasiruošusių procesų eilėje ir iškelti į diską.



UNIX SVR4 procesų būvių diagrama



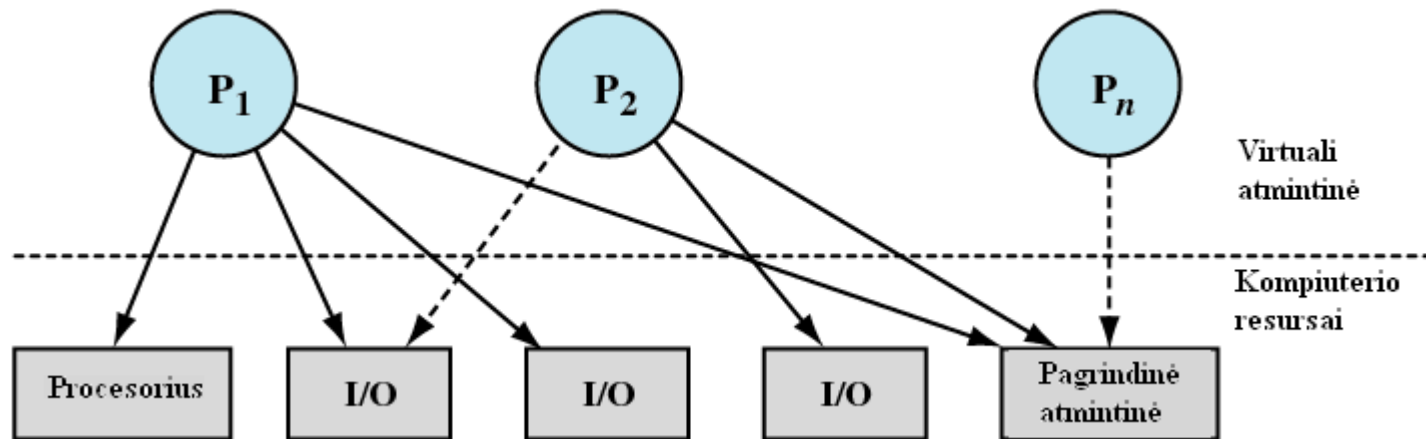
Proceso suspendavimo priežastys

- OS turi atlaisvinti pakankamai vietos pagrindinėje atmintinėje norėdama įkelti procesą, kuris yra pasiruošęs vykdymui.
- OS gali atidėti foninį ar paslaugos procesą, kuris kelia problemą (tuo išvengiant mirties taško).
- Vartotojas gali atidėti programos vykdymą derinimo tikslais ar norėdamas panaudoti kažkurį sistemos resursą.
- Procesas gali būti vykdomas periodiškai (pvz. apskaitos procesas) ir todėl jis gali būti atidedamas iki kito laiko momento.
- Tėvo procesas gali norėti atidėti vaiko proceso vykdymą, siekdamas jį patikrinti ar modifikuoti.

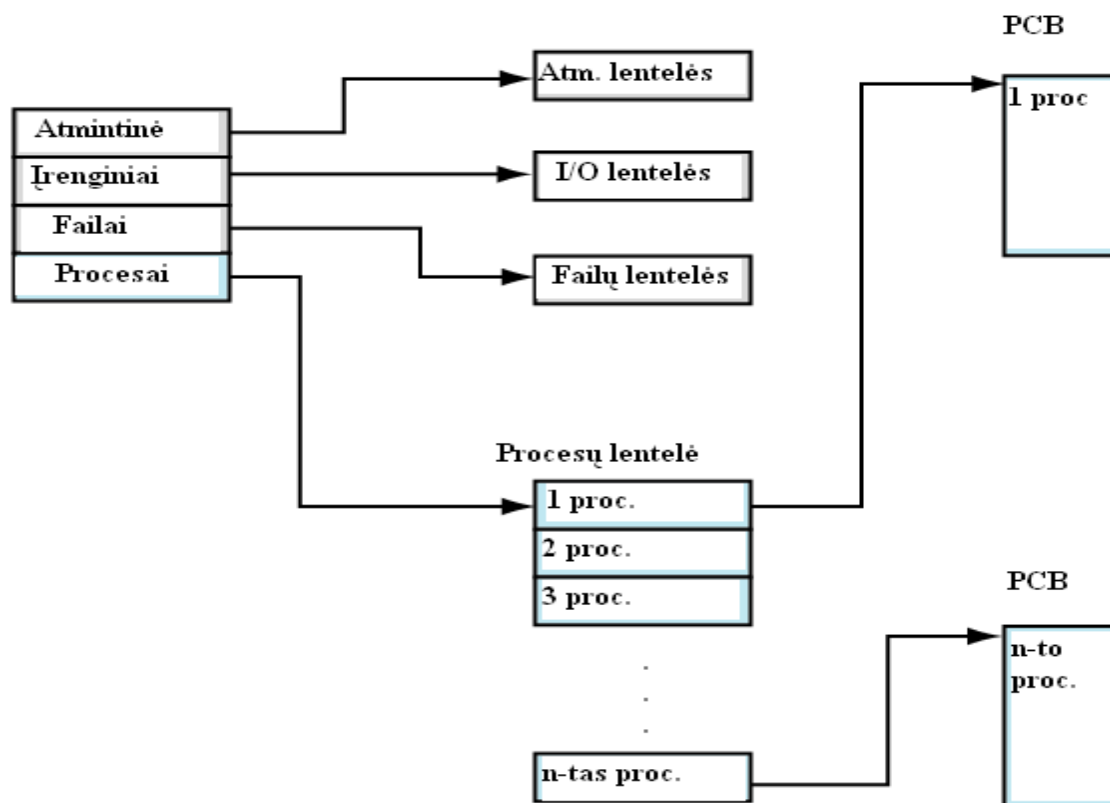
OS kontrolēs struktūros

- Jose saugoma informācija apie kiekvieno *proceso* ir kiekvieno *resurso* einamā būvī
- Sudaromos lentelēs apie kiekvienā OS valdomā objektā

Procesai ir resursai



OS palaikomos lentelės



Lentelės surištos su atmintine

■ Jose atsispindi:

- Informacija susijusi su **virtualios** atmintinės valdymu:
 - Pagrindinės atmintinės skyrimas procesams
 - Išorinės atmintinės skyrimas procesams
- **Prieigos** parametrai, susiję su **bendrai** naudojamomis atmintinės sritimis.
- Informacija apie laisvą vietą:
 - Pagrindinėje atmintinėje
 - Diske

I/O lentelės

- I/O įrenginiai:

- ☐ prieinami ar priskirti
- ☐ I/O operacijų statusas
- ☐ Buferiai pagrindinėj atmintinėj, kaip I/O perdavimų šaltinis ar paskirties taškas.

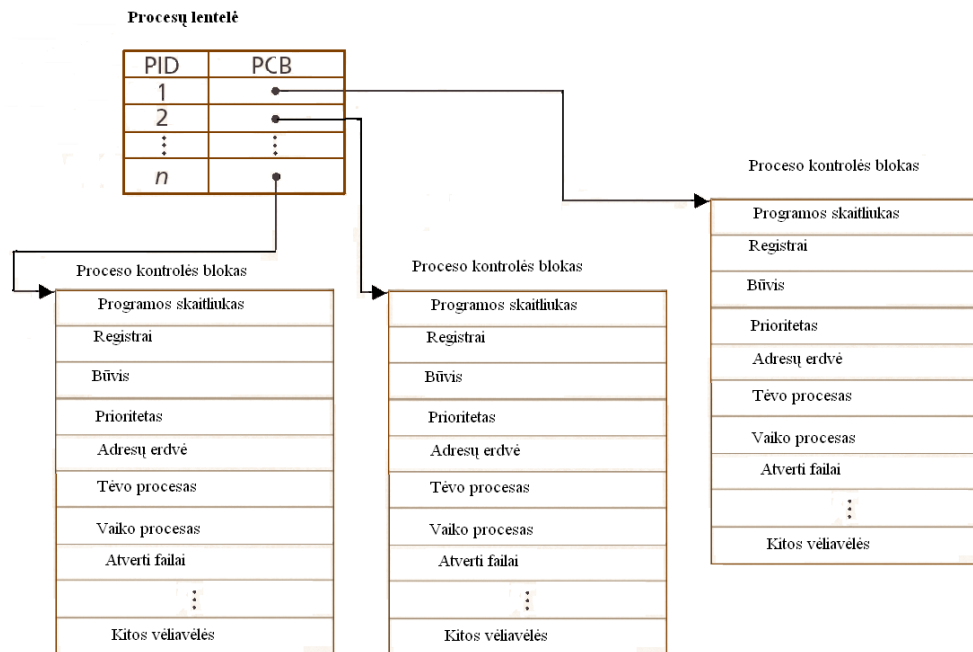


Failų lentelės

- Sistemos atvertų failų lentelė
 - Jų vieta išoriniuose įrenginiuose
- Procesų atvertų failų lentelės
 - Parametrai

Procesų lentelė

- Kokie procesai veikia
- Kur patalpintas proceso kontrolės blokas



Vykdymo modos (režimai)

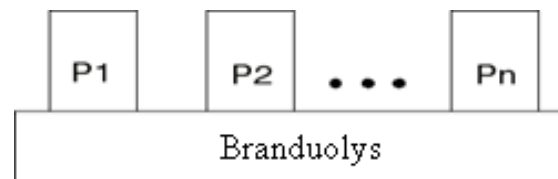
- Privilegiuota, dar vadinama sistemos moda, branduolio moda, ar kontrolės moda.
 - *Dirbant šiame režime galimi įvairūs veiksmai kontrolės registrų atžvilgiu, galimi veiksmai susiję su įrašais sisteminėse duomenų lentelėse bei kiti veiksmai su svarbia sistemai informacija.*
- Vartotojo moda.
- Vykdymo modos pasikeitimas vadinamas modos perjungimu (mode switch), kurio nereikėtų painioti su proceso perjungimu, pastarasis reiškia, kad CPU persijungia nuo vieno proceso prie kito

Operacinės sistemos veiksmų vykdymas

- Jei įsivaizduosime pačią operacinę sistemą kaip tam tikrų programų rinkinį, tai kyla klausimas, ar operacinė sistema yra procesas?
- Jei operacinė sistema taip pat yra procesas, tai kaip yra kontroliuojami jos veiksmai? Atsakymas į šį klausimą priklauso nuo operacinės sistemos projekto.
- Galimi keli sprendimo variantai:
 - *Neprocesinis branduolys*
 - *Operacinės sistemos veiksmai – vartotojiškuose procesuose*
 - *Operacinės sistemos veiksmai – atskiruose procesuose*

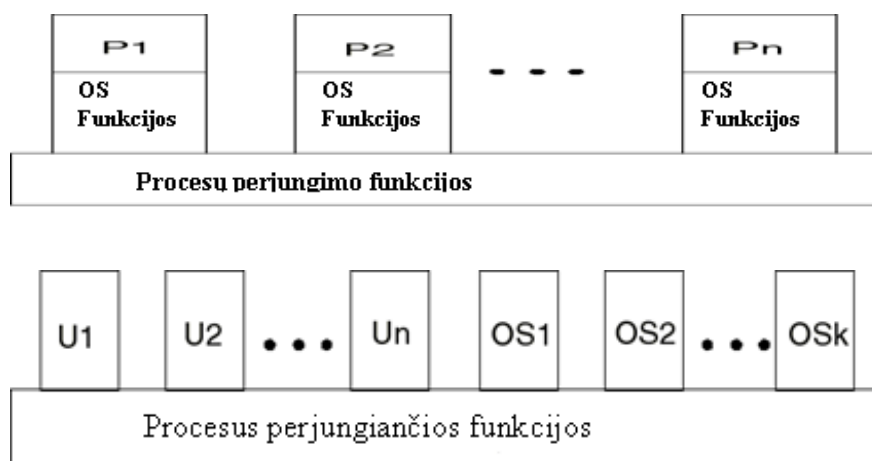
Neprocesinis branduolys

- Senesnis šios problemos variantas buvo susijęs su tokia realizacija, kai branduolys nebuvo laikomas paprastu procesu - neprocesinis branduolys
- Proceso sąvoka buvo taikoma tik vartotojų programoms. Operacinės sistemos kodas buvo vykdomas kaip atskiras vienetas privilegijuotame režime, ir jis niekad nebuvo vykdomas proceso viduje.



OS veiksmų vykdymas vartotojiškuose procesuose

- Šios realizacijos atveju virtualiai visas operacinės sistemos kodas yra vykdomas vartotojo proceso kontekste



Procesais pagrįstos operacinės sistemos

operacinė sistema tiesiog laikoma sisteminių procesų rinkiniu. Pagrindinės branduolio funkcijos yra realizuojamos kaip atskiri sisteminiai procesai

UNIX procesu būviai

Table 3.9 UNIX Process States

User Running	Executing in user mode.
Kernel Running	Executing in kernel mode.
Ready to Run, in Memory	Ready to run as soon as the kernel schedules it.
Asleep in Memory	Unable to execute until an event occurs; process is in main memory (a blocked state).
Ready to Run, Swapped	Process is ready to run, but the swapper must swap the process into main memory before the kernel can schedule it to execute.
Sleeping, Swapped	The process is awaiting an event and has been swapped to secondary storage (a blocked state).
Preempted	Process is returning from kernel to user mode, but the kernel preempts it and does a process switch to schedule another process.
Created	Process is newly created and not yet ready to run.
Zombie	Process no longer exists, but it leaves a record for its parent process to collect.

UNIX Proceso kontekstas

Table 3.10 UNIX Process Image

User-Level Context	
Process Text	Executable machine instructions of the program
Process Data	Data accessible by the program of this process
User Stack	Contains the arguments, local variables, and pointers for functions executing in user mode
Shared Memory	Memory shared with other processes, used for interprocess communication
Register Context	
Program Counter	Address of next instruction to be executed; may be in kernel or user memory space of this process
Processor Status Register	Contains the hardware status at the time of preemption; contents and format are hardware dependent
Stack Pointer	Points to the top of the kernel or user stack, depending on the mode of operation at the time of preemption
General-Purpose Registers	Hardware dependent
System-Level Context	
Process Table Entry	Defines state of a process; this information is always accessible to the operating system
U (user) Area	Process control information that needs to be accessed only in the context of the process
Per Process Region Table	Defines the mapping from virtual to physical addresses; also contains a permission field that indicates the type of access allowed the process: read-only, read-write, or read-execute
Kernel Stack	Contains the stack frame of kernel procedures as the process executes in kernel mode