

Operacinės sistemas

N. Sarafinienė
2013 m.



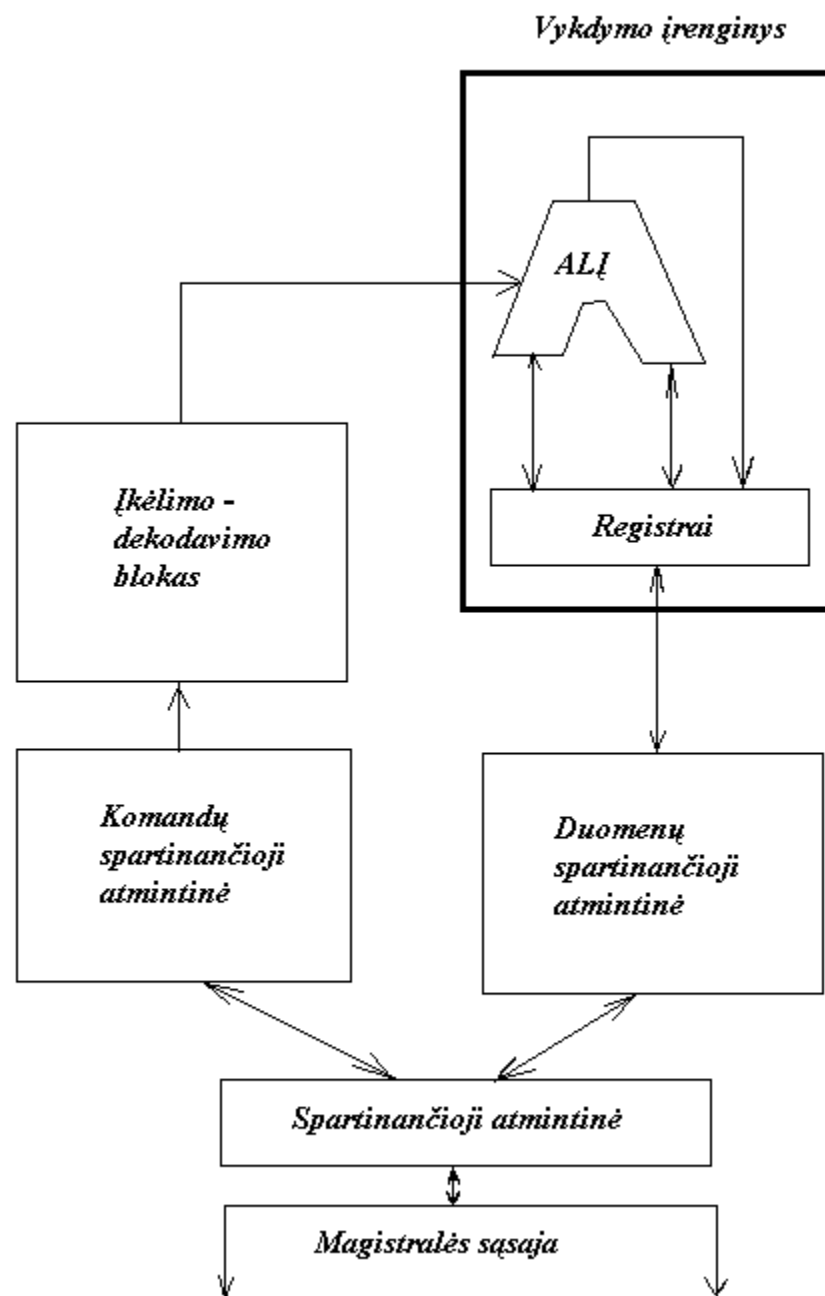
Kalbėsime

- Komandų vykdymas
- Pertrauktys
- Operacinės sistemos komponentės
- Operacinės sistemos paslaugos
- Sisteminiai kvietiniai
- Sisteminės programos
- Operacinių sistemų architektūros

Bitas, Baitas, Žodis, Blokas

- Mažiausias informacijos kiekis – **bitas**. 1 bitas gali turėti tik dvi reikšmes: 0 arba 1, „teisinga“ arba „klaidinga“.
- 1 B (**baitas**) = 8 b (bitai); Vienas baitas gali turėti 2^8 skirtingų reikšmių – jei saugomas sveikas skaičius, tai reikšmė gali būti nuo 0 iki 255.
 - KB (kilobaitas) 1024 B; [MB (megabaitas), GB (gigabaitas), TB (terabaitas) ,]
- Žodis: Šis terminas naudojamas pažymėti fiksuoto ilgio bitų grupei, kuri kompiuterio yra apdorojama kartu. Bitų kiekis žodyje (arba žodžio ilgis) yra svarbi charakteristika kompiuterių architektūroje.
- Moderniuose kompiuteriuose naudojamas žodžio ilgis yra 16, 32, ar 64 bitų
- Dauguma CPU registrai operuoja žodžiais, žodžiais vyksta mainai tarp OA ir CPU.
- .
- Blokas – informacijos mainų vienetas tarp OA ir išorinės atmintinės

Procesoriaus komponentės



Komandų vykdymas

- Procesoriaus AL įrenginys vykdo kompiuterio komandas:

- Komanda – tai arba aritmetinė operacija

$+$ $-$ $*$ $/$

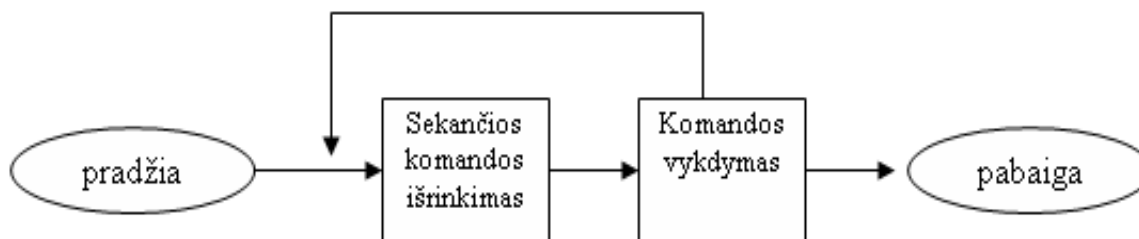
arba loginis palyginimo veiksmas:

$>$ $<$ $=$ **not** $=$.

Visa kita yra suvedama į šias kelias operacijas, tik viena operacija yra vykdoma kiekviename mašinos cikle.



Bazinis programos vykdymo ciklas



komandos adresas –yra saugomas **programos skaitliuke PC**

Išrinkta vykdymui komanda yra talpinama į komandų registrą **IR**

Komandos gali būti įvairaus tipo. Tai gali būti:

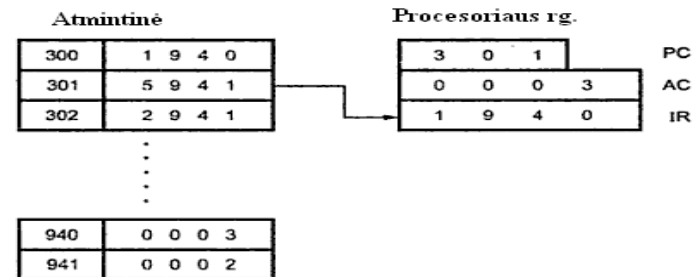
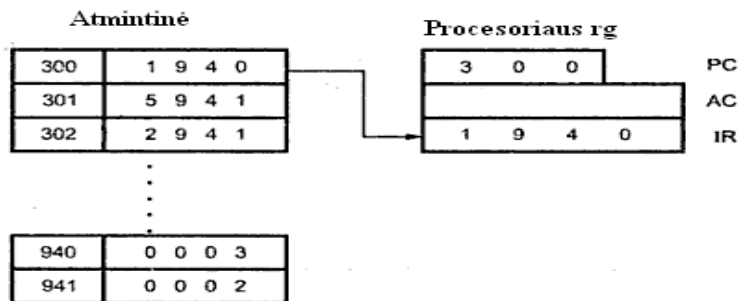
- aritmetiniai arba loginiai veiksmai su duomenimis,
- duomenų perdavimas tarp atmintinės ir procesoriaus,
- duomenų perkėlimas iš išorinės atmintinės,
- kontrolės veiksmai, leidžiantys keisti komandų vykdymo seką.

Komandų vykdymas

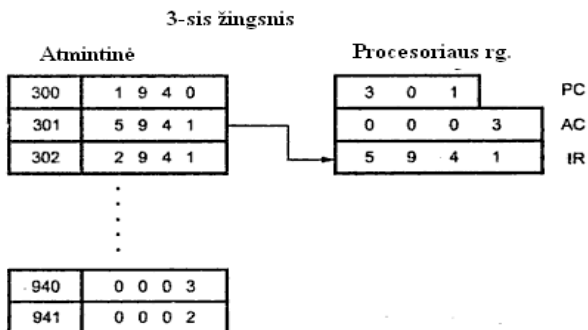
Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės

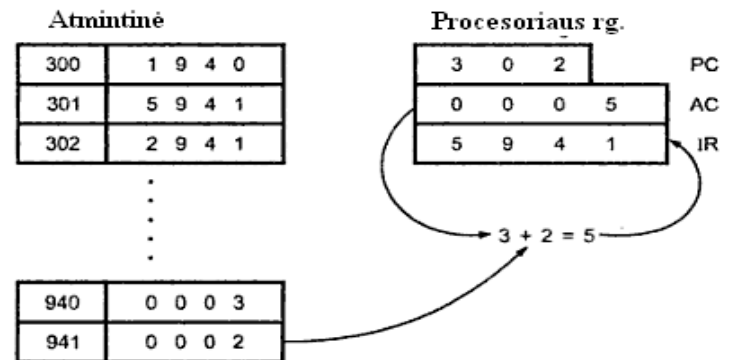
2-sis žingsnis



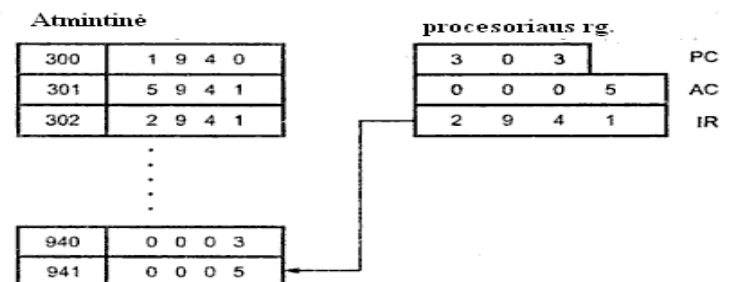
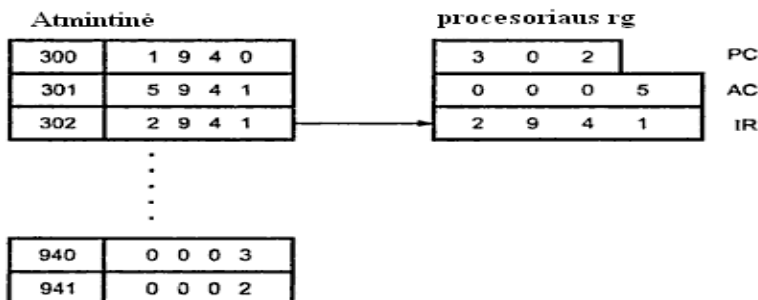
4-tas žingsnis



3-sis žingsnis



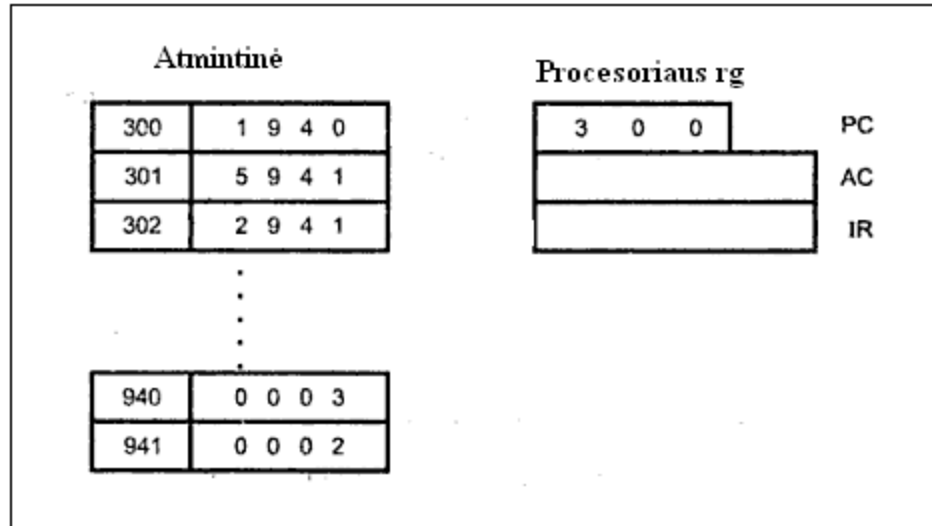
6 žingsnis



Komandų vykdymas

Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

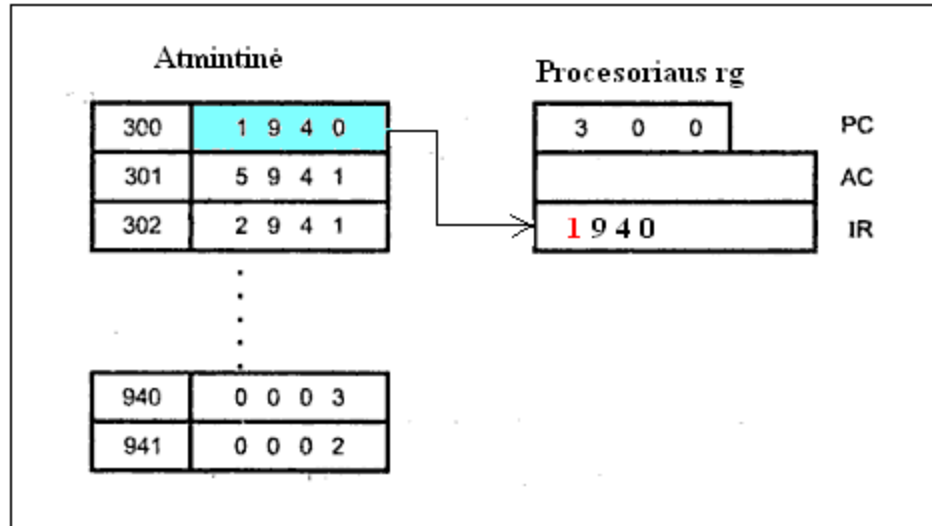
Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės



Komandų vykdymas

Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

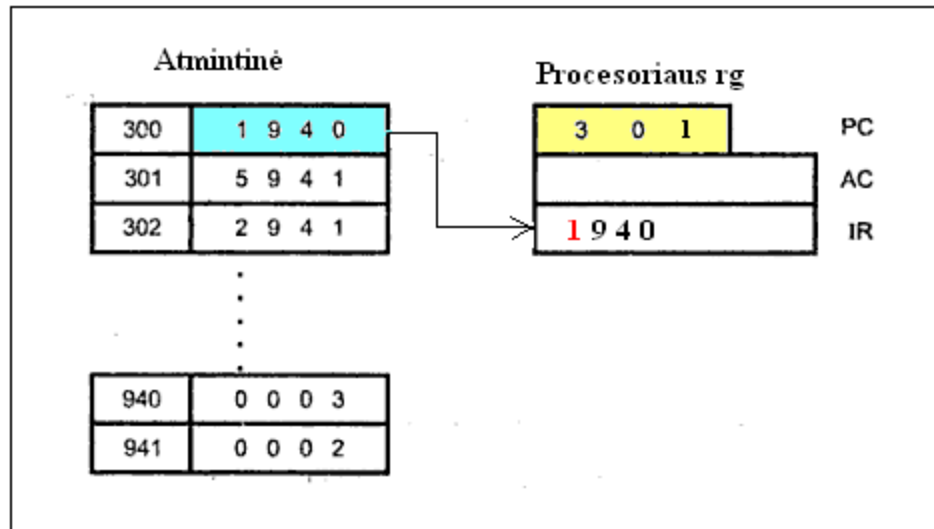
Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės



Komandų vykdymas

Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

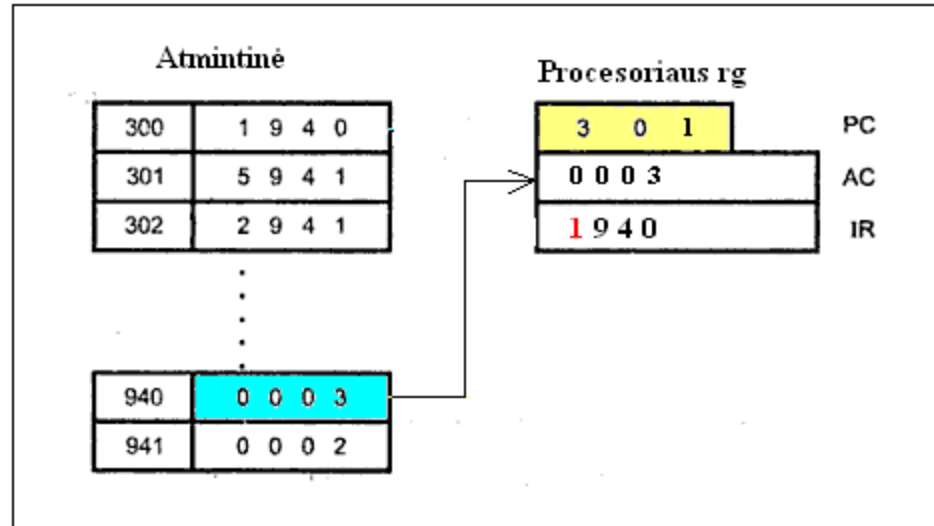
Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės



Komandų vykdymas

Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

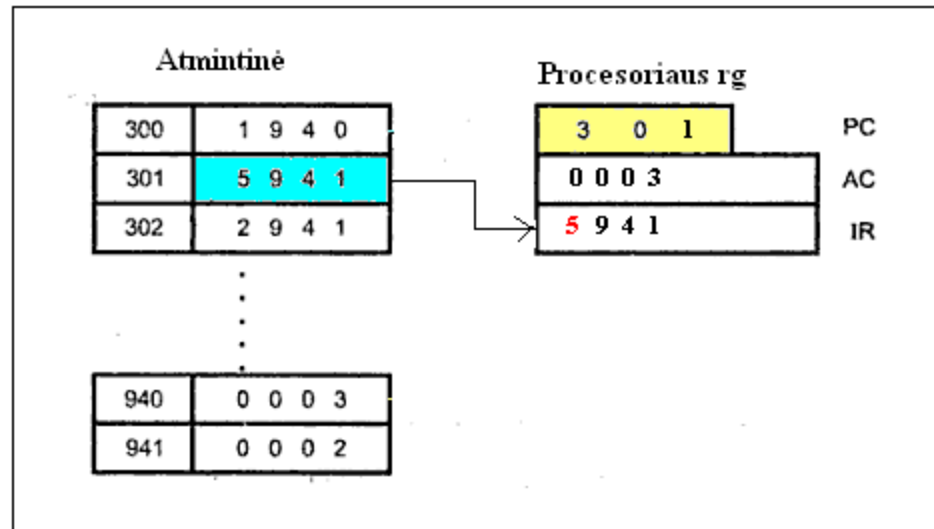
Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės



Komandų vykdymas

Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

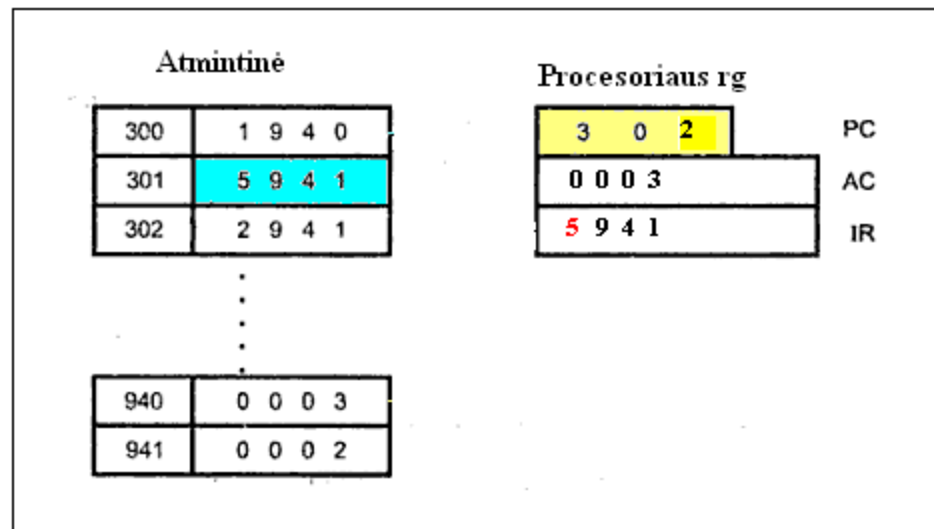
Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės



Komandų vykdymas

Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

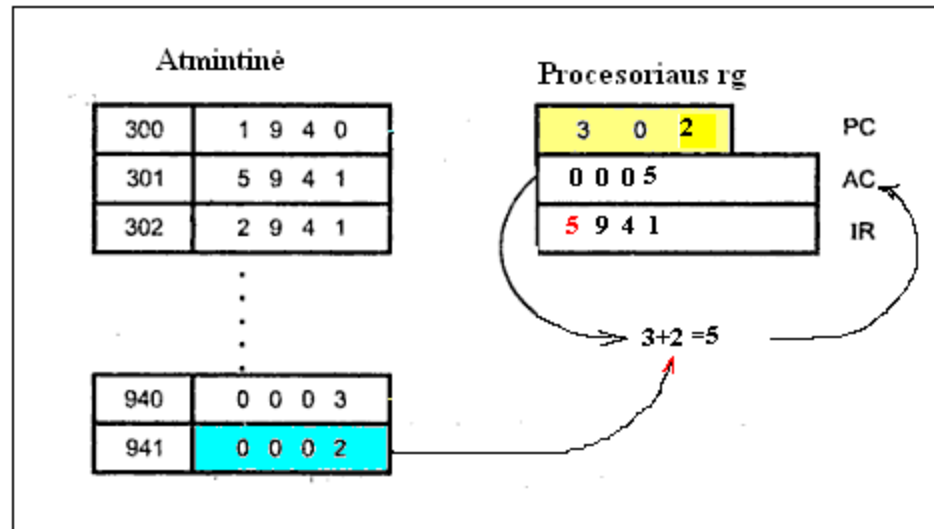
Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės



Komandų vykdymas

Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

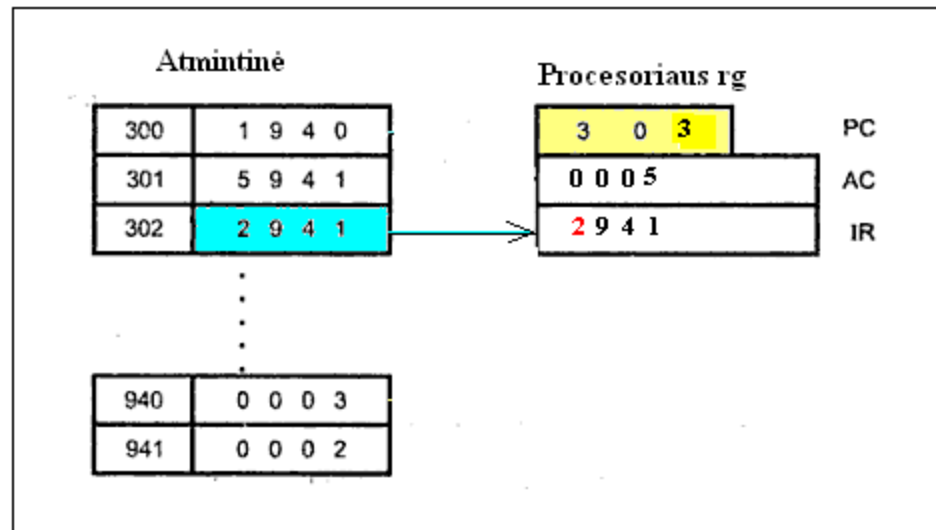
Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės



Komandų vykdymas

Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

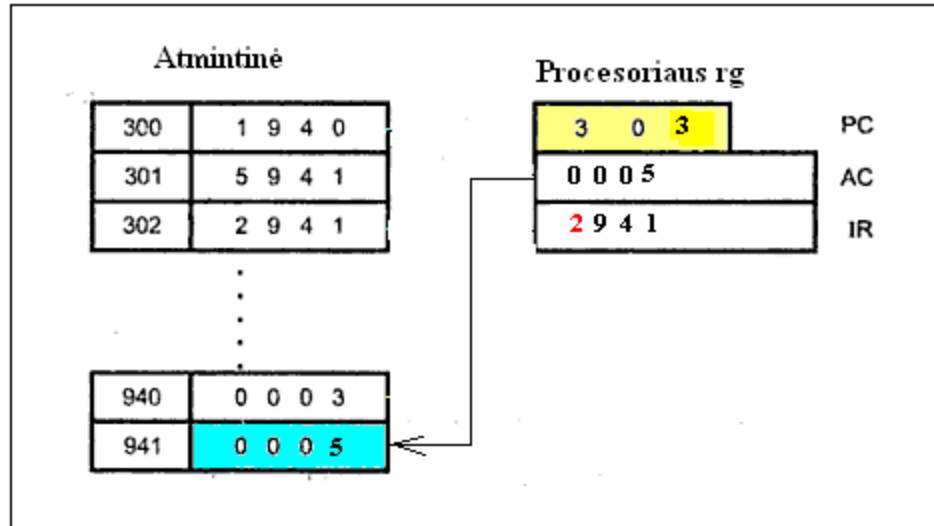
Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės



Komandų vykdymas

Rg	Paskirtis
PC	Komandos adresas
IR	Vykdoma komanda
AC	Tarpinė kaupyklė

Nr.	Op. kodas	Paskirtis
1.	0 0 0 1	Įkelti iš atmintinės
2.	0 0 1 0	Išsaugoti atmintinėje
3.	0 1 0 1	Pridėti prie kaupiklio iš atmintinės



Operacinės sistemos modelis

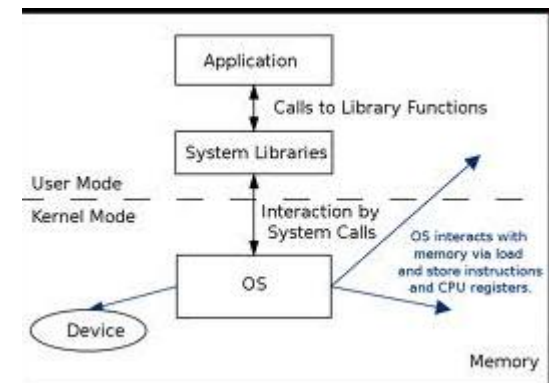
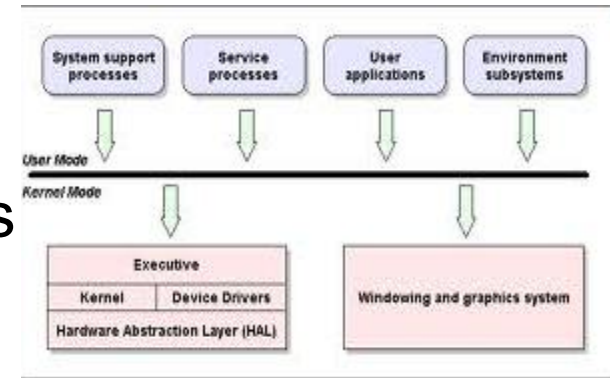
- Daugumoje daugiavartotojiškų operacinių sistemų taikomosios programos yra atskirtos nuo operacinės sistemos:

- Operacinės sistemos kodas vykdomas privilegijuotame režime:

- Turi prieigą prie sisteminių duomenų ir techninės įrangos

- Taikomųjų programų kodas vykdomas vartotojiškam režime:

- Ribota prieiga prie sisteminių duomenų ir nėra tiesioginio ryšio su technine įranga



Vartotojo ir branduolio režimas

■ Branduolio režimas

□ Kai CPU vykdo veiksmus šiame režime:

- Jis gali vykdyti bet kokią komandą
- Jis gali keisti bet kurios atmintinės vietos turinį
- Jis gali modifikuoti bet kokio registro turinį tiek CPU, tiek įrenginio
- Tokiu metu egzistuoja pilna kompiuterio kontrolė

□ OS servais vykdomi branduolio režime

Vartotojo ir branduolio režimas

■ Vartotojo režimas

□ Kai CPU vykdo veiksmus šiame režime:

- CPU gali naudoti tik ribotą komandų aibę
- CPU gali modifikuoti tik tas atmintinės sekcijas, kurios yra skirtos proceso vykdomai programai
- CPU turi prieigą tik prie ribotos aibės CPU registrų ir neturi prieigos prie įrenginių registrų
- Yra ribota prieiga prie kompiuterio resursų

□ Vartotojų programos vykdomos vartotojų režime

Vartotojo ir branduolio režimas

- Kai OS startuoja, ji startuoja branduolio režime
- Branduolio režime OS nustato pertraukčių vektorius ir inicializuoja visus įrenginius
- Tada ji startuoja pirmą procesą ir persijungia į vartotojo režimą
- Vartotojo režime ji vykdo visus foninio režimo sisteminius procesus (demonus arba servisus)
- Tada ji paleidžia vartotojo shello procesą arba “windows manager”

Vartotojo ir branduolio režimas

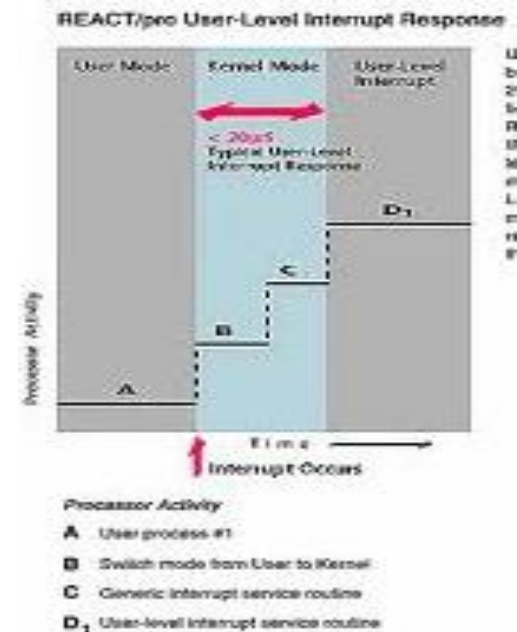
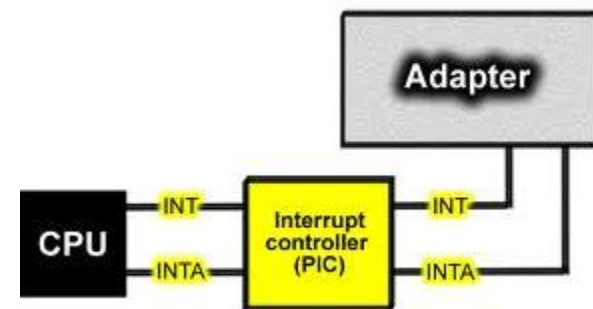
- Daugumą laiko CPU veikia vartotojo režime
- Vartotojų programos vykdomos vartotojų režime
- Persijungimas į branduolio režimą:
 - užprašant sisteminių paslaugų (sisteminiai kvietiniai)
 - pasirodžius pertraukčiai
- Pertrauktys yra apdorojamos branduolio režime
- Pasibaigus pertraukties apdorojimui grįžtama į vartotojišką režimą

Vartotojo ir branduolio

- Branduolio ir vartotojo režimas yra naudojamas dėl:
 - *Saugumo*: OS kvietinių vykdymas branduolio režime užtikrina tai, kad šie veiksmai bus atlikti saugiai
 - *Patikimumo*: jei procesas kreipsis netinkamu adresu, OS galės nužudyti procesą, OS veiks toliau. Jei įvyks nenumatyta situacija procese – žus tik procesas, OS tęs savo darbą; nenumatyta situacija branduolio režime gali pražudyti ir OS
 - *Teisingumo*: OS kvietiniai vykdomi branduolio režime gali užtikrinti teisingą prieigą

Pertrauktys

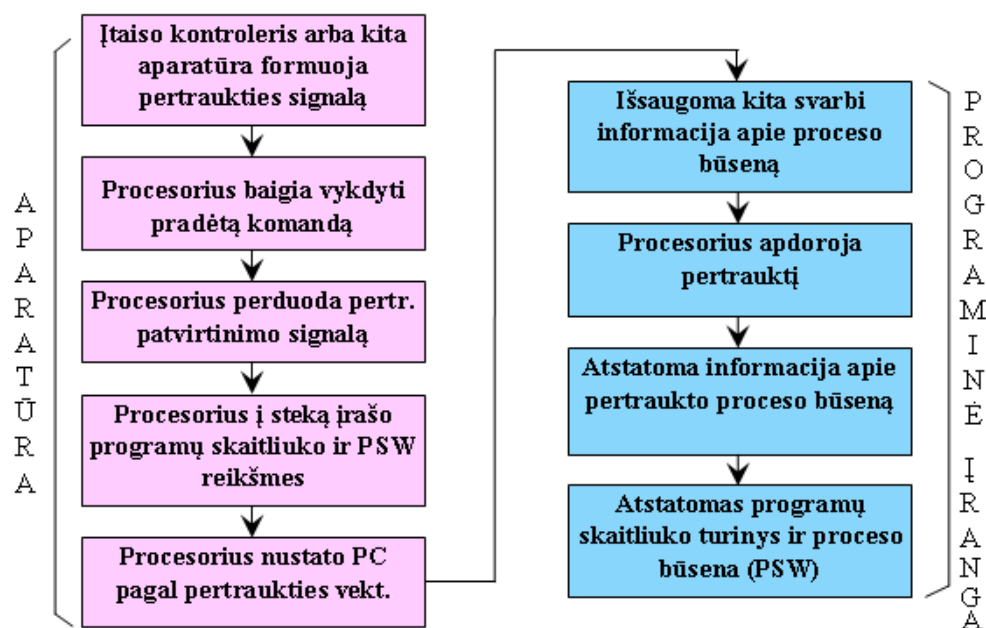
- Pertrauktis – tai įvykis, kuris reikalauja neatidėliotino dėmesio.
- Pertrauktis (angl. interrupt) – tai programos ar įrenginio įvykdytas veiksmas, kurio metu nutraukiamas einamosios programos vykdymas ir valdymas perduodamas *pertraukties dorokliui*.
- Įvykus pertraukčiai yra išsaugomi būvio duomenys
- Baigus apdoroti pertrauktį bus grįžtama į tą pačią vietą, kurioje sistema buvo prieš pasirodant pertraukčiai.
- Pertrauktys:
 - Pelės judesys
 - Klavišo paspaudimas
 - Atėjęs ethernet paketas



Pertrauktys ir išskirtinės situacijos

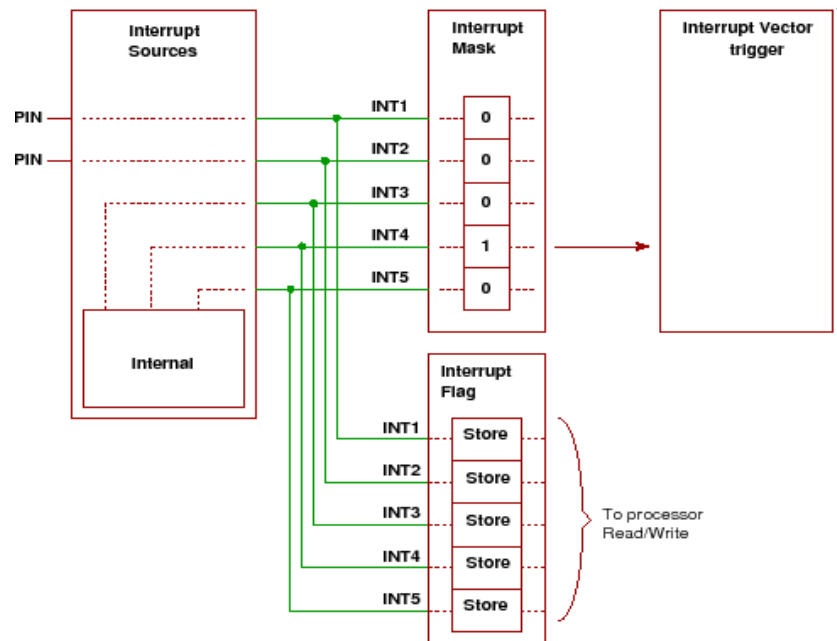
- Pertrauktis yra asinchroninis įvykis, nesurištas su tuo metu procesoriaus vykdomais veiksmais
 - I/O veiksmas, laikrodžio mechanizmas
 - Gali būti leistini, gali būti uždraudžiami
- Išskirtinė situacija yra sinchroninis įvykis, susijęs su procesoriaus vykdomais veiksmais
 - Dalyba iš nulio, prieiga prie neleistinos adresų srities
 - Sisteminiai kvietiniai yra interpretuojami kaip išskirtinės situacijos

Pertraukčių apdorojimo schema

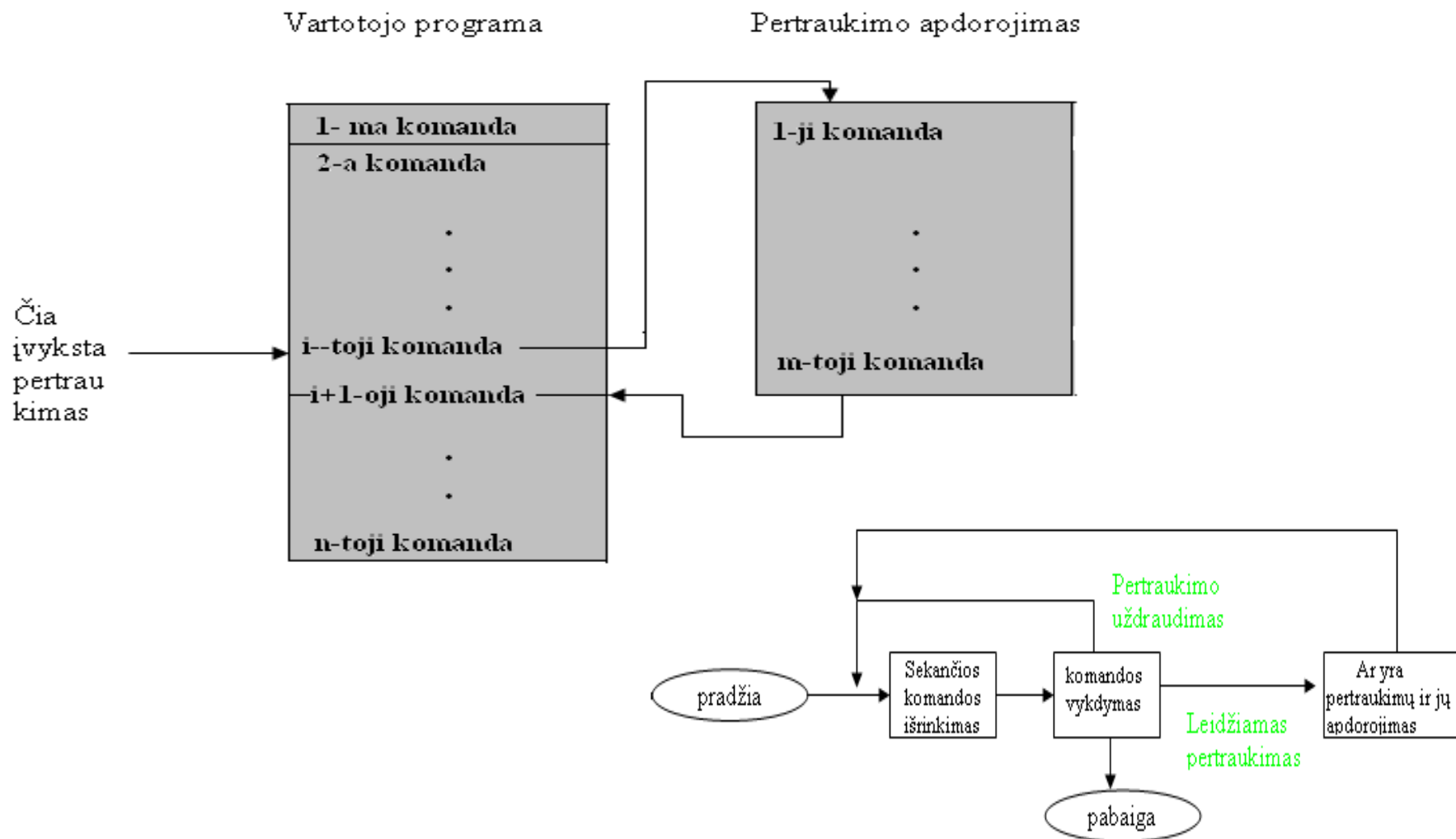


Pertraukčių apdorojimo schema

ISR	Interrupt Service Routine.
Interrupt vector	The address that holds the location of the ISR.
Interrupt mask	Controls which interrupts are active.
NMI	Non Maskable Interrupt - an interrupt that is always active.
Asynchronous event	An event that could happen at any time.
Context switching	Saving/ <u>restoring</u> data before & after the ISR.



Valdymo perdavimas pertraukties metu



“Užimtas laukimas” – busy waiting

- Kartais OS gali būti nuspręsta nenaudoti pertraukčių kai kurių įrenginių atžvilgiu ir tokiu atveju CPU sukasi tuščiu ciklu kol užklausa baigiama:

OS užprašo Įrenginio operacijos

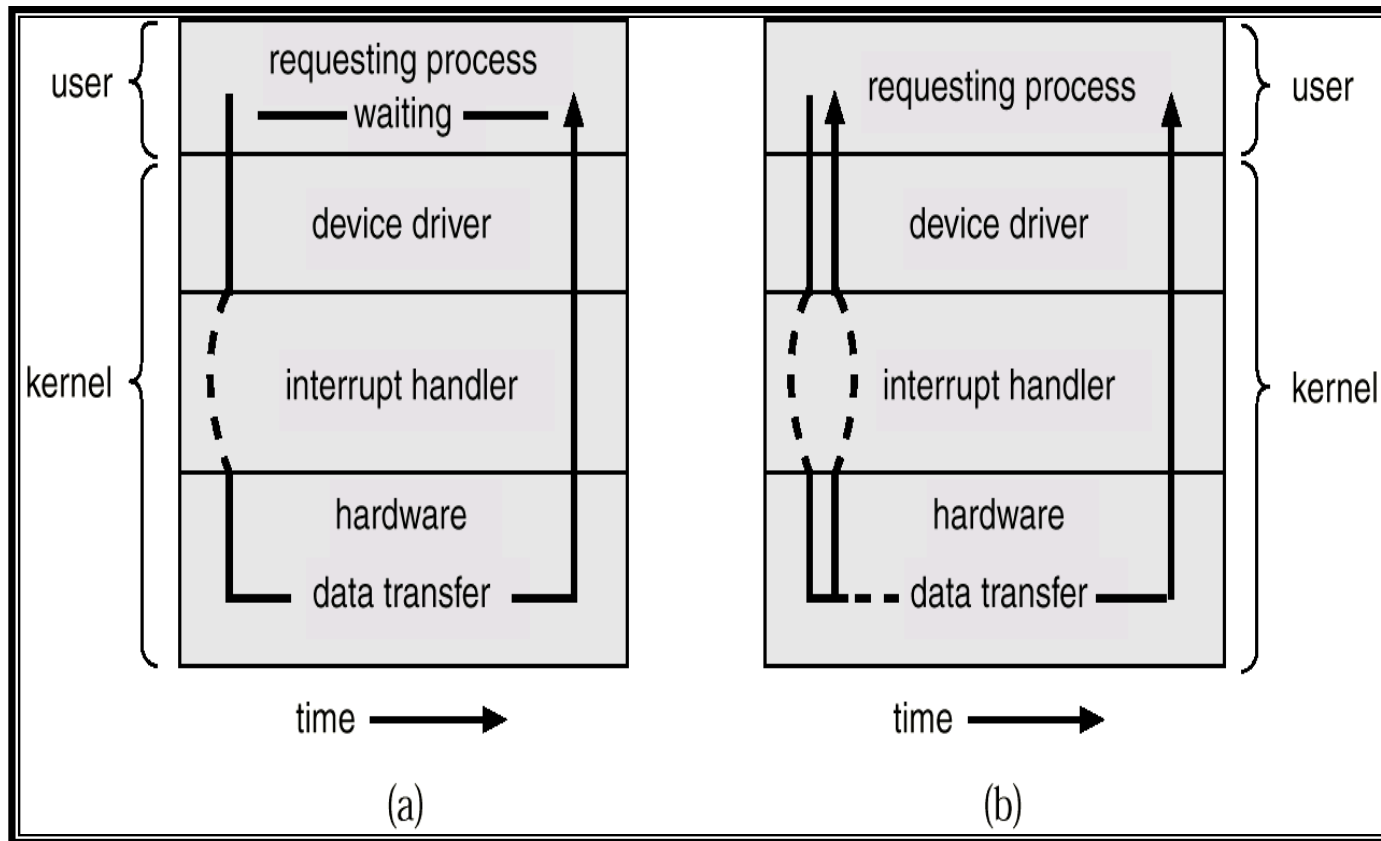
Kol operacija nebaigiama

Nedaro nieko;

Tęsia veiksmus

- Tokio tipo vykdymas vadinamas užimtu laukimu (busy waiting arba polling)– yra tuščiai eikvojami CPU ciklai
- Tai gali būti naudojama atspausdinimui derinimo pranešimų prieš tęsiant veiksmus toliau

Synchronous/asynchronous I/O



Pertraukčių vektoriai

- Tai yra nuorodų masyvas. Nuorodos rodo į skirtingus pertraukčių doroklius skirtingo tipo pertrauktims

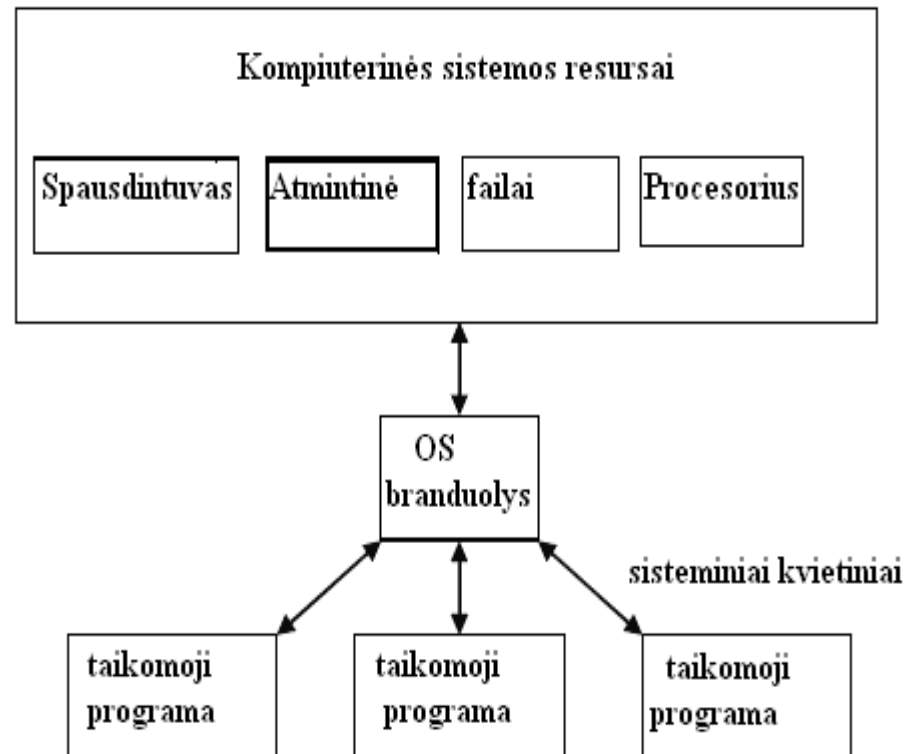


Pertraukčių tipai

1. Įrenginių generuojamos pertrauktys
2. Matematinės išskirtinės situacijos
3. Adresacijos klaida
4. Programinės pertrauktys užprašant operacinės sistemos paslaugų

Sisteminiai kvietiniai

- Sisteminiai kvietiniai teikia sąsają tarp taikomųjų procesų ir operacinės sistemos.
- Sisteminiai kvietiniai leidžia procesams užprašyti OS tam tikrų paslaugų, kurių procesams patiems neleidžiama vykdyti.
- Šie veiksmai dažnai yra susiję su privilegijuotų komandų vykdymu.



Sisteminių kvietinių tipai

Sisteminiai kvietiniai gali būti susiję su:

- **Procesų** kontrole:

- `fork()`, `system()`, `execve()`, ...

- **Failų** valdymu:

- `open()`, `close()`, ...

- **Įrenginių** valdymu

- `read()`, `write()`, ...

- **Informacijos palaikymu**

- `mmap()`, `gettimeofday()`, `set attributes()` ...

- **Komunikacijomis:**

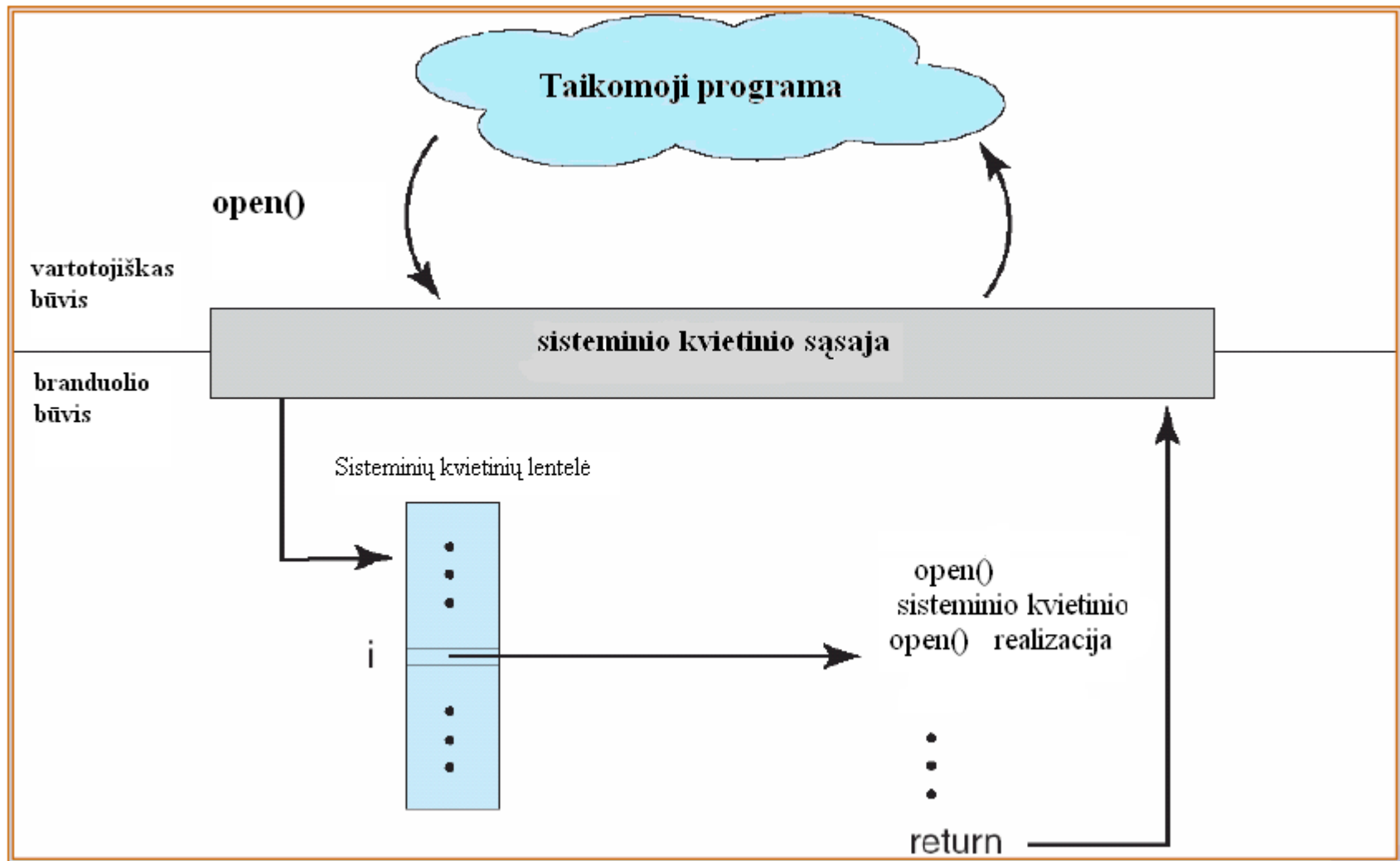
- `signal()`, `kill()`, ...

- Sisteminiai kvietiniai yra tarsi OS API žiūrint iš vartotojo programos pusės

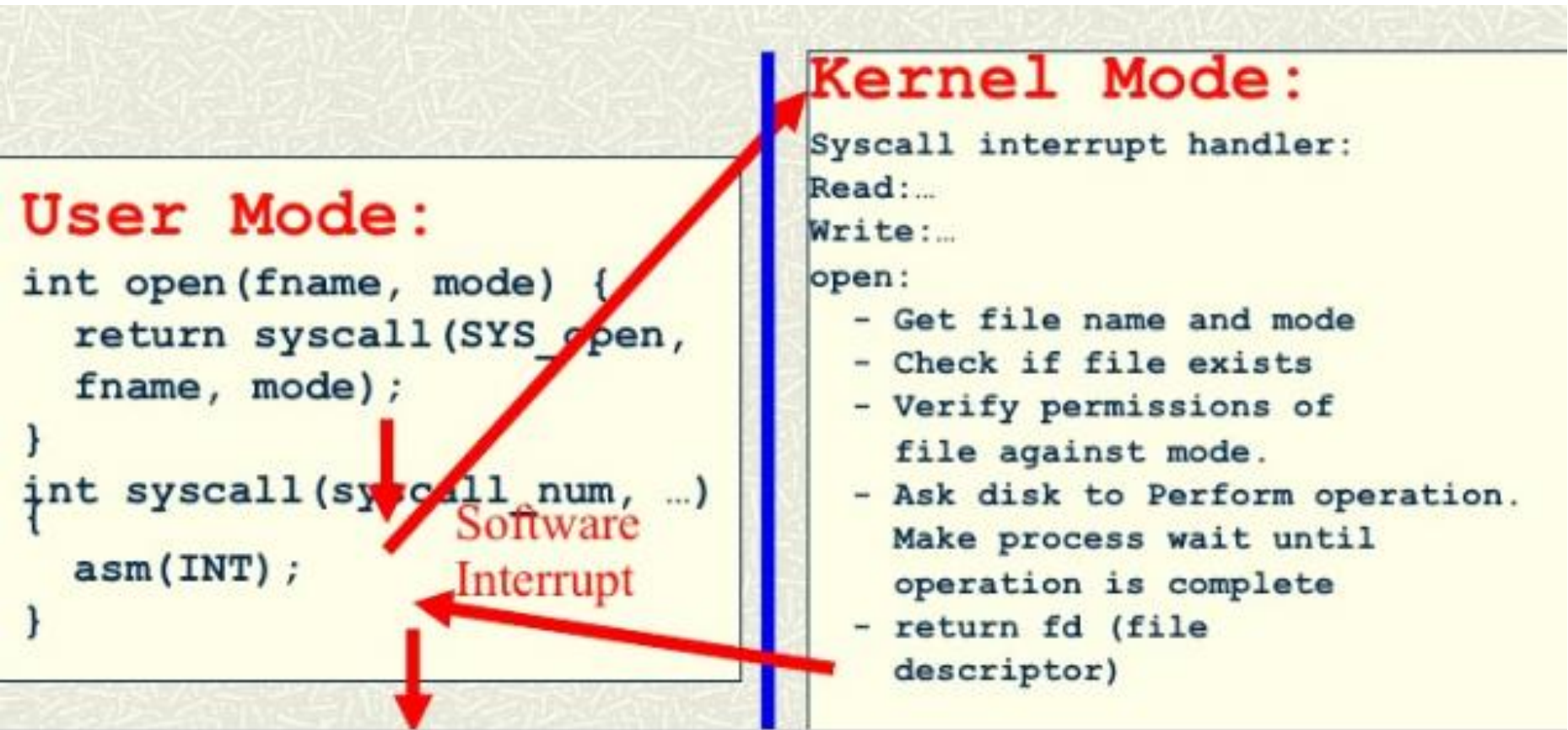
Kodėl naudojamos sisteminės pertrauktys vietoje funkcijos kvietinio?

- Funkcijos, tokios kaip $\sin(x)$ nėra sisteminiai kvietiniai
- Tik operacijos, kurios reikalauja, kad jos būtų vykdomos OS branduolio režime yra dalimi sisteminių kvietinių
- Sisteminės pertrauktys perjungs CPU į branduolio režimą
- OS paslaugos reikalauja būti vykdomos branduolio režimu, nes:
 - Jos reikalauja privilegijuotų komandų
 - Reikia prieigos prie įrenginių ir branduolio duomenų struktūrų

Sisteminiai kvietiniai



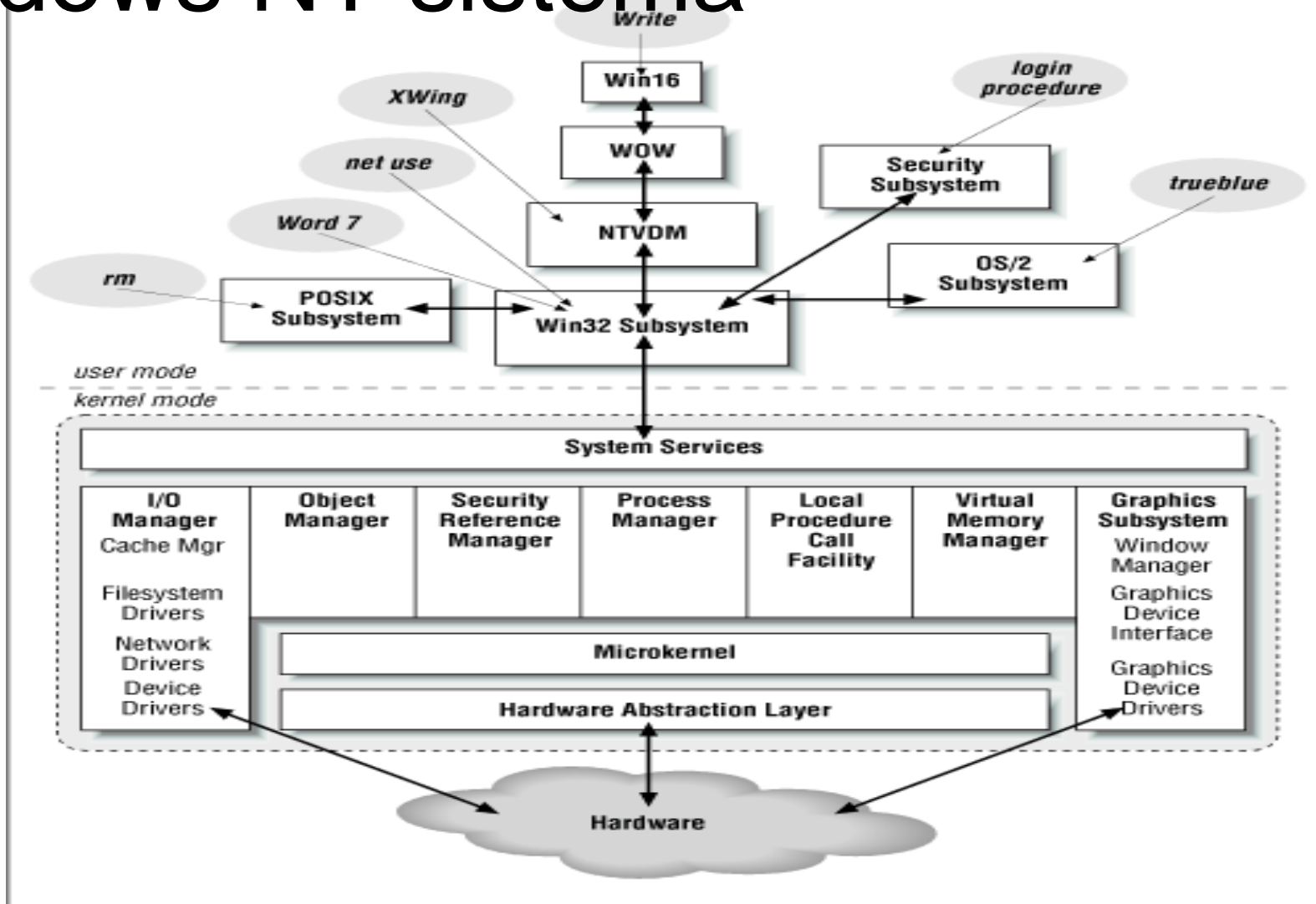
Sisteminio kvietinio vykdymas



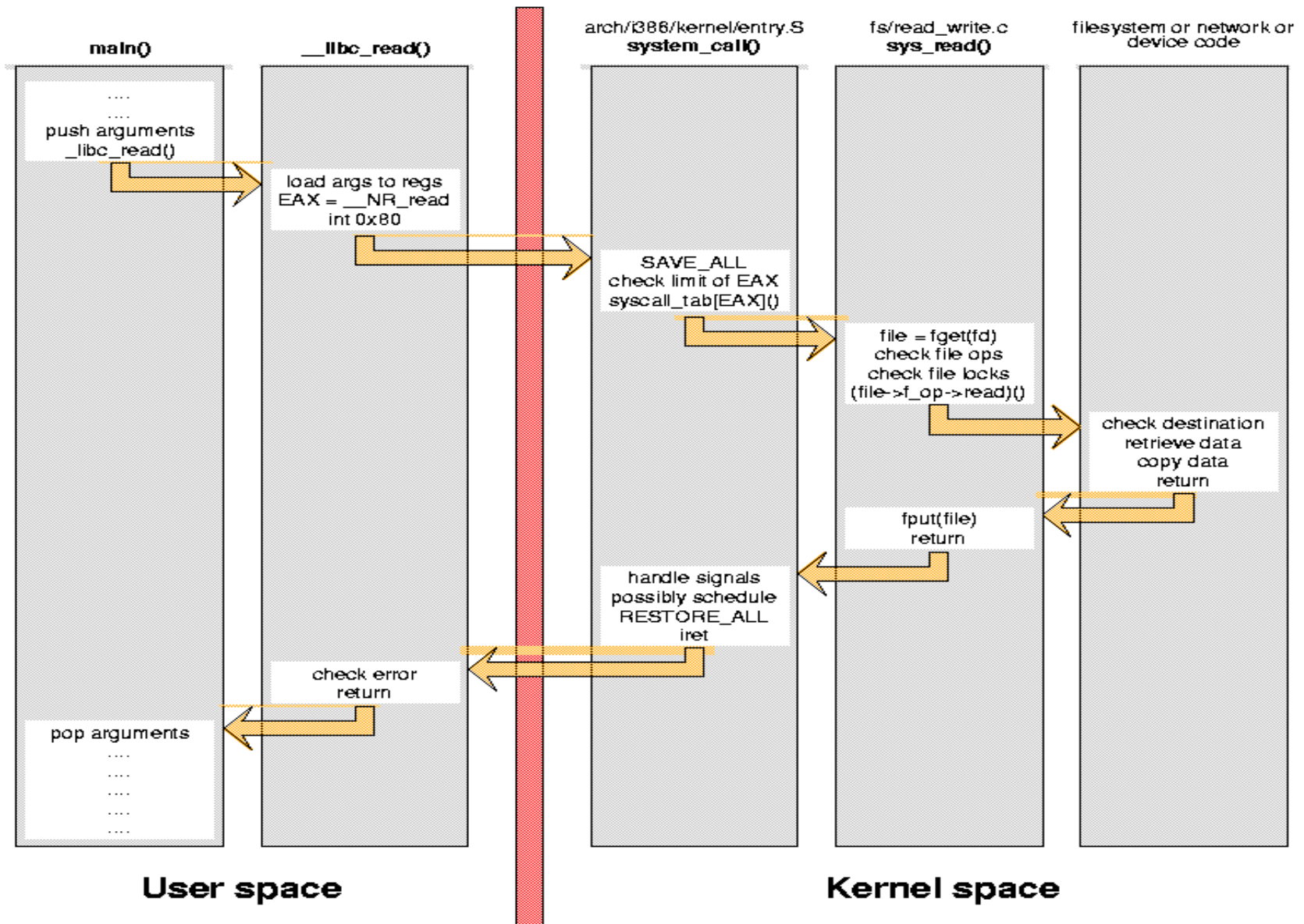
Sisteminiai kvietiniai ir taikomųjų programų sąsaja

- Sisteminiai kvietiniai dažniau yra kviečiami ne tiesiogiai, o per **taikomųjų programų sąsają API** (Application Program Interface).
- Trys dažniausiai naudojamos API yra:
 - *Win32 API - Windows*
 - *POSIX API - UNIX, Linux, and Mac OS X*
 - *Java API - Java virtualioms mašinoms (JVM)*
- Paprastai su kiekvienu sisteminiu kvietiniu yra surištas tam tikras **numeris**. Sisteminių kvietinių sąsaja palaiko lentelę, indeksuotą šiais numeriais. Jos pagalba yra iškviečiamas atitinkamo sisteminio kvietinio vykdymas OS branduolyje.
- API paslepia daugumą sąsajos detalių nuo programuotojo. Sisteminiai kvietiniai yra realizuojami per **bibliotekines** funkcijas.

Windows NT sistema

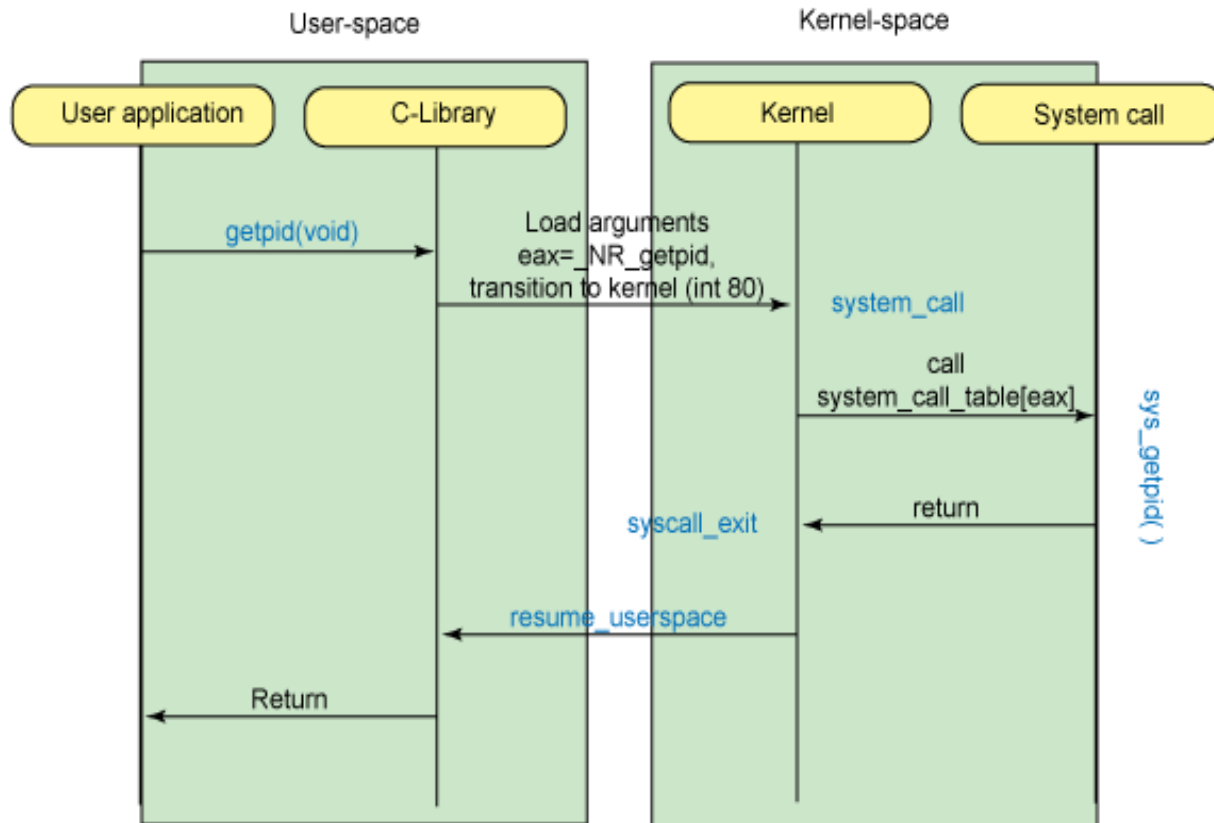


n=read(fd, bufer,size)



Supaprastinta Linux sisteminio kvietinio schema

The simplified flow of a system call using the interrupt method



C bibliotekos standartinės funkcijos panaudojimo pavyzdys

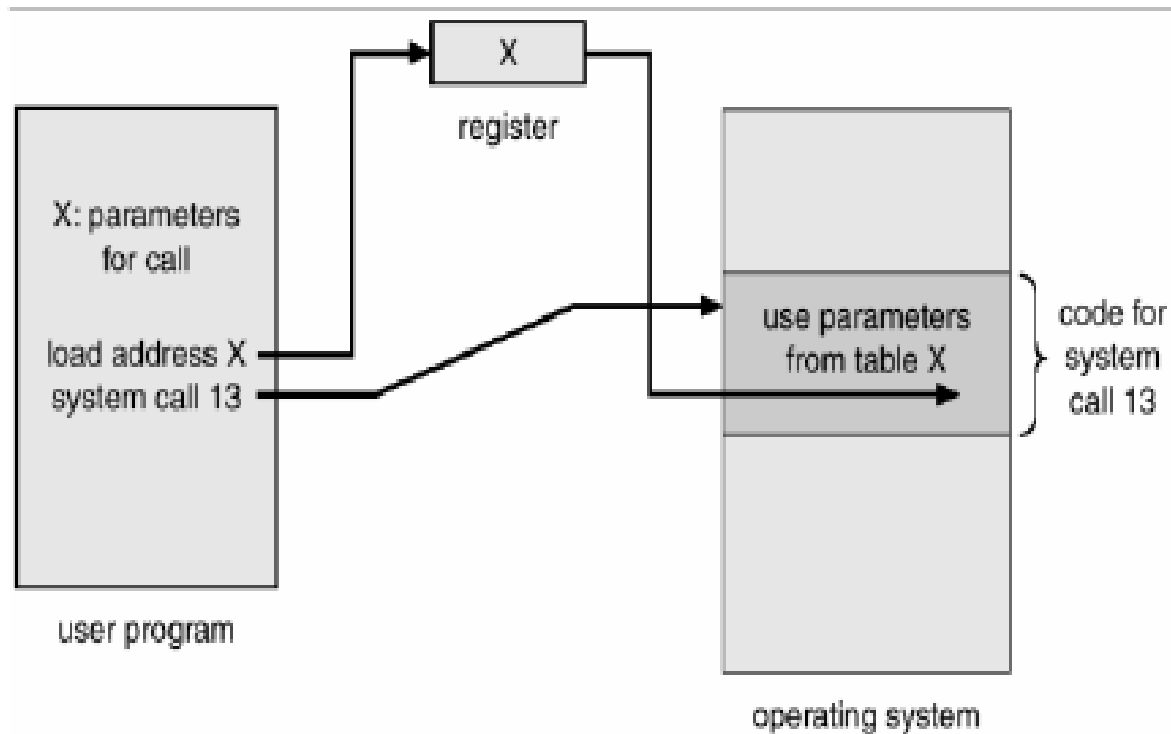
- C kalba parašyta programa kviečia bibliotekinę funkciją, kuri kviečia `write(fd, buff, n)` sisteminių kvietinių
- Generuojama pertrauktis sisteminio kvietinio realizacijai
- Pertraukčių doroklis patikrina argumentus, patikrina, ar `fd` yra failų deskriptorius failui atidaromam rašymo režime, taip pat tikrinama, ar `[buff, buff+n]` yra teisėtas atmintinės režis – jei kuris nors tikrinimas duoda neigiamą rezultatą grąžinamas -1
- OS liepia kietam diskui vykdyti rašymą iš buferio į diską – į failą, nusakomą deskriptoriumi `fd`.
- OS sustabdo einamąjį procesą kol bus baigta rašymo operacija- OS persijungia prie kito proceso vykdymo
- Diskui baigus vykdyti operaciją generuojama pertrauktis
- Pertraukčių doroklė perjungia procesą, iškvietusį `write` į pasiruošusį būvį, taigi šį procesą OS galės vėl pratęsti vykdyme.

Parametrų reikšmių perdavimas sisteminių kvietinių metu

Dažnai reikia ne tik identifikuoti reikiamą sisteminių kvietinį bet ir nurodyti daugiau informacijos. Informacijos kiekis ir tipas gali skirtis priklausomai nuo sisteminio kvietinio ir OS

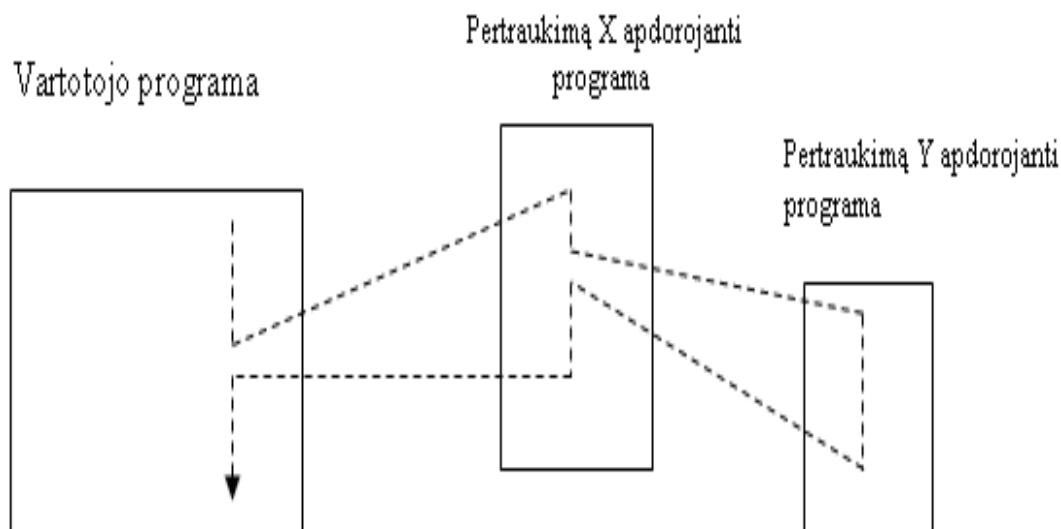
- Yra galimi trys pagrindiniai metodai perduodant informaciją OS
 - patalpinant juos registruose
 - Parametrus galima išsaugoti atmintinėje tam tikrame bloke ar lentelėje ir šio bloko adresą perduoti kaip parametą į registrą.
 - Šis metodas taikomas Linux ir Solaris
 - Parametrus *programoje* sudėti (*push*) į *steką* ir išimti (*pop*) iš operacinės sistemos steko
- Tiek parametrų perdavimas per bloką, tiek per steką neriboja nei parametrų kiekio ar jų dydžio.

Parametrų perdavimas lentelė



Daugybiniai- kartotiniai pertraukimai

- Vykdam pertraukimo apdorojimą sistemoje gali atsirasti kiti pertraukimo signalai. Galimi du sprendimo būdai:
 - Uždrausti pertraukimus vykdam pertraukimo apdorojimą.
 - Leisti atlikti pertraukimo apdorojimo pertraukimą.



Windows sistemos pertraukčių prioritetinė schema (pertraukčių užklausų lygiai)

31	Aukštas (naudojama kai sistema yra stabdoma)
30	Įtampos kritimas
29	Pertrauktis tarp procesorių (užprašant kitą procesorių atlikti užduotį)
28	Laikrodis
27	sinchronizacijos
26	n-tasis įrenginys
	...
	...
5	CMSI
4	...
3	Įrenginys 1
2	DPC/dispečerio (procedūrų kvietiniai vykdomi branduolio režimu)
1	APC (asinkroninės procedūros vykdomos vartotojo gijos aplinkoje)
0	Pasyvus

Pertrauktys, surištos su technine įranga

Programinės įrangos pertrauktys

Normalus gijų vykdymas

Kas yra operacinė sistema?

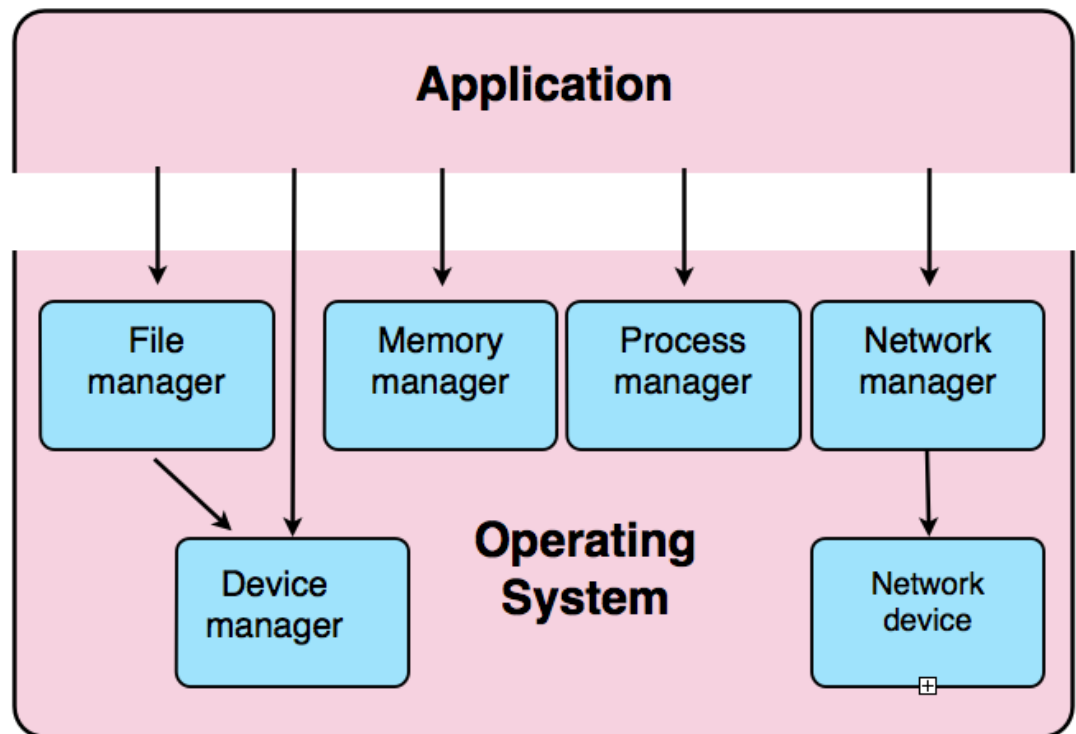
- Tai programa, kuri yra tarpininku tarp techninės įrangos ir taikomųjų programų.
- Tai programa kuri startuoja užkraunant sistemą.

Ką siūlo operacinė sistema?

- Ji siūlo paslaugas, kurios yra bendros įvairioms taikomosioms programoms: spausdinimas, bibliotekos, prieiga prie failų,...
- Ji įgalina daugelio procesų vienalaikį vykdymą
- Ji siūlo apsaugą/saugą
- Ji siūlo patikimumą – kiekviena programa vykdoma savo „smėlio dėžėje“.
- Ji siūlo sąžiningą resursų skirstymą.
- Ji leidžia keliems vartotojams dirbti su ta pačia mašina

Operacinės sistemos komponentės

- **Procesų** valdymas
- Pagrindinės **atmintinės** valdymas
- **Failų** valdymas
- **I/O sistemos** valdymas
- **Antrinės** atmintinės valdymas
- Darbo **tinkle** palaikymas
- **Apsaugos** sistema
- Komandų **interpretavimo** sistema

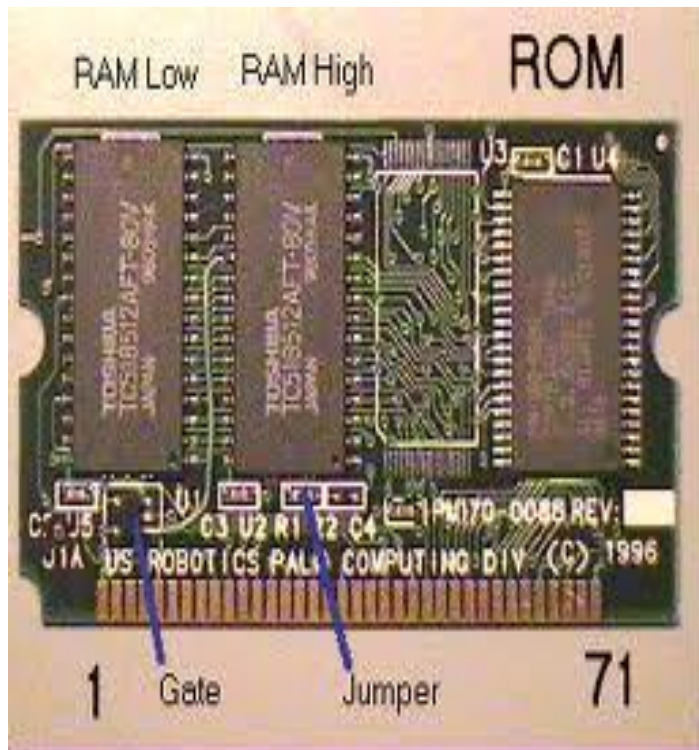


Procesų valdymas



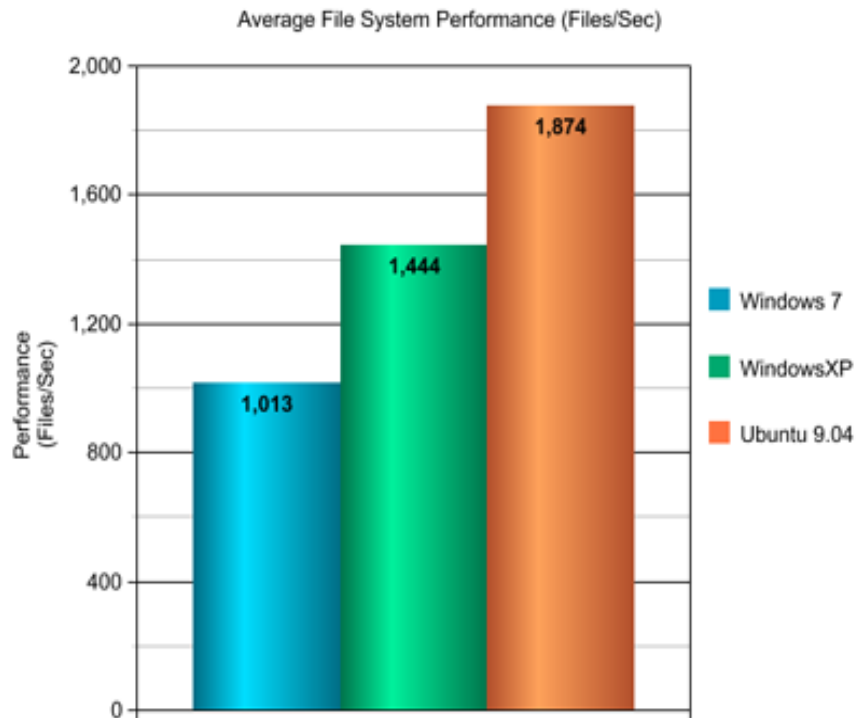
- Procesas – tai sąvoka apibūdinanti visą programos vykdymo kontekstą.
- Procesui reikia tam tikrų išteklių
- Operacinė sistema yra atsakinga už šiuos veiksmus, susijusius su vartotojų bei sisteminių procesų valdymu:
 - Procesų sukūrimas ir užbaigimas
 - Procesų suspendavimas ir atnaujinimas
 - Techninės įrangos resursų skirstymas procesams
 - Užtikrinimas mechanizmų, skirtų:
 - Procesų sinchronizacijai
 - Procesų komunikacijoms
 - Mirties taško situacijos vengimui bei apdorojimui

Pagrindinės atmintinės valdymas



- Atmintinę galima įsivaizduoti kaip *didelį žodžių ar baitų masyvą*, kurių kiekvienas turi savo adresą.
- Pagrindinė atmintinė -tai atmintinė su greita **prieiga** prie duomenų, kuriais dalinasi procesorius ir I/O įrenginiai.
- Tai yra trumpalaikės atminties įrenginys. Ji praranda savo turinį atjungus įtampą.
- Operacinė sistema vykdo tokius uždavinius, susijusius su pagrindinės atmintinės valdymu:
 - Seka tai, kurios atmintinės sritys yra užimtos ir kieno.
 - Sprendžia tai, kokį procesą įkelti į atmintinę, kai ši tampa prieinama.
 - Išskiria atmintinės sritį procesams arba atima, iškilus reikalui.

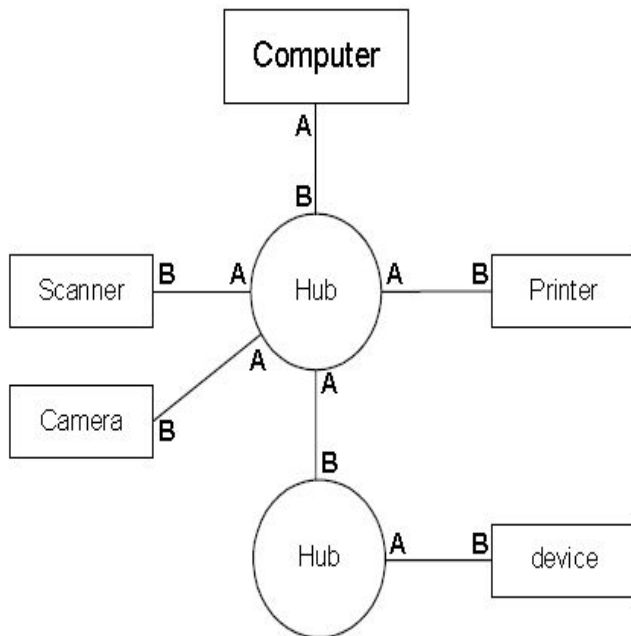
Failų valdymas



- Failas yra tarpusavy susijusios informacijos rinkinys. Paprastai failai atitinka **programas** bei **duomenis**. Jie saugomi diske (ilgalaikis saugojimas).
- Failų sistema paprastai yra sudaryta iš katalogų.
- Operacinė sistema atsakinga už šių veiksmų, susijusių su failais vykdymą:
 - ☐ Failų sukūrimas ir šalinimas.
 - ☐ Katalogų sukūrimas ir naikinimas.
 - ☐ Pateikimą priemonių, skirtų įvairioms manipuliacijoms su failais bei katalogais.
 - ☐ Failų patalpinimą antrinėje atmintinėje.
 - ☐ Failų kilojimą tarp pagrindinės ir antrinės atmintinės.

I/O sistemos valdymas

USB Hub Configuration.



- I/O sistemą sudaro:
 - **Buferiai, spartinanti atmintinė**
 - Kiekvieną įrenginį valdo tam įrenginiui skirta **tvarkyklė**
 - Tik tvarkyklei yra žinomi atskirų įrenginių savitumai
- Rūpinasi tuo, kad visi I/O įrenginiai būtų traktuojami vienodai vartotojo požiūriu
- Slepia daugumą I/O detalių žemame lygyje, taip, kad procesai ir kiti aukštesni lygiai galėtų vykdyti veiksmus su įrenginiais naudodami paprastas operacijas, tokias kaip read, write, open, close, lock, unlock

Antrinės atmintinės valdymas



- Dauguma modernių kompiuterinių sistemų naudoja diskus programų bei duomenų saugojimui.
- Operacinė sistema yra atsakinga už sekančių užduočių, susijusių su disko valdymu vykdymą:
 - Laisvos vietos valdymas
 - Atminties priskyrimas
 - Disko darbo planavimas

Komandų interpretavimas

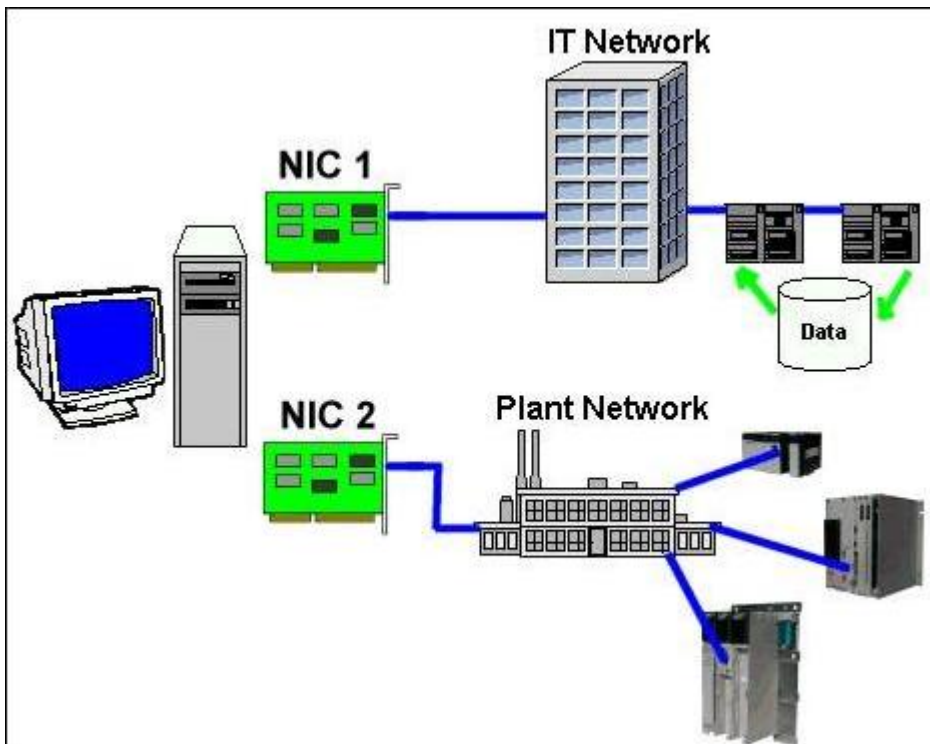


- Komandų interpretatorius yra sąsaja tarp vartotojo ir operacinės sistemos. Tai programa, kuri priima ir interpretuoja, analizuoja ir vykdo reikiamus veiksmus pagal vartotojo pateikiamas komandas.

Vadinama įvairiai: komandinės eilutės interpretatorius, shell'as (apvalkalas).

- Paskirtis – priimti ir įvykdyti vartotojo pateiktą komandą
- Paprastai tai nėra operacinės sistemos branduolio dalis, nes gali būti naudojami keli interpretatoriai

Darbo tinkle palaikymas



- Darbo tinkle palaikymas leidžia skirtingiems kompiuteriams komunikuoti ir veikti tarsi didžiulei virtualiai sistemai.
- Komunikacijos tinkle vyksta pagal tam tikrus protokolus
- Operacinė sistema turi įdiegtą protokolų steką ir įrenginių tvarkyklės tinkliniams įrenginiams

Apsaugos sistema



- Jei kompiuterinė sistema turi daug vartotojų ir leidžia daugelio procesų lygiagrečių vykdymą, tai šie procesai turi būti apsaugoti vienas nuo kito aktyvumo.
- Apsauga žymi tai, kad turi būti kontroliuojama prieiga prie programų, procesų, kompiuterinės sistemos išteklių.



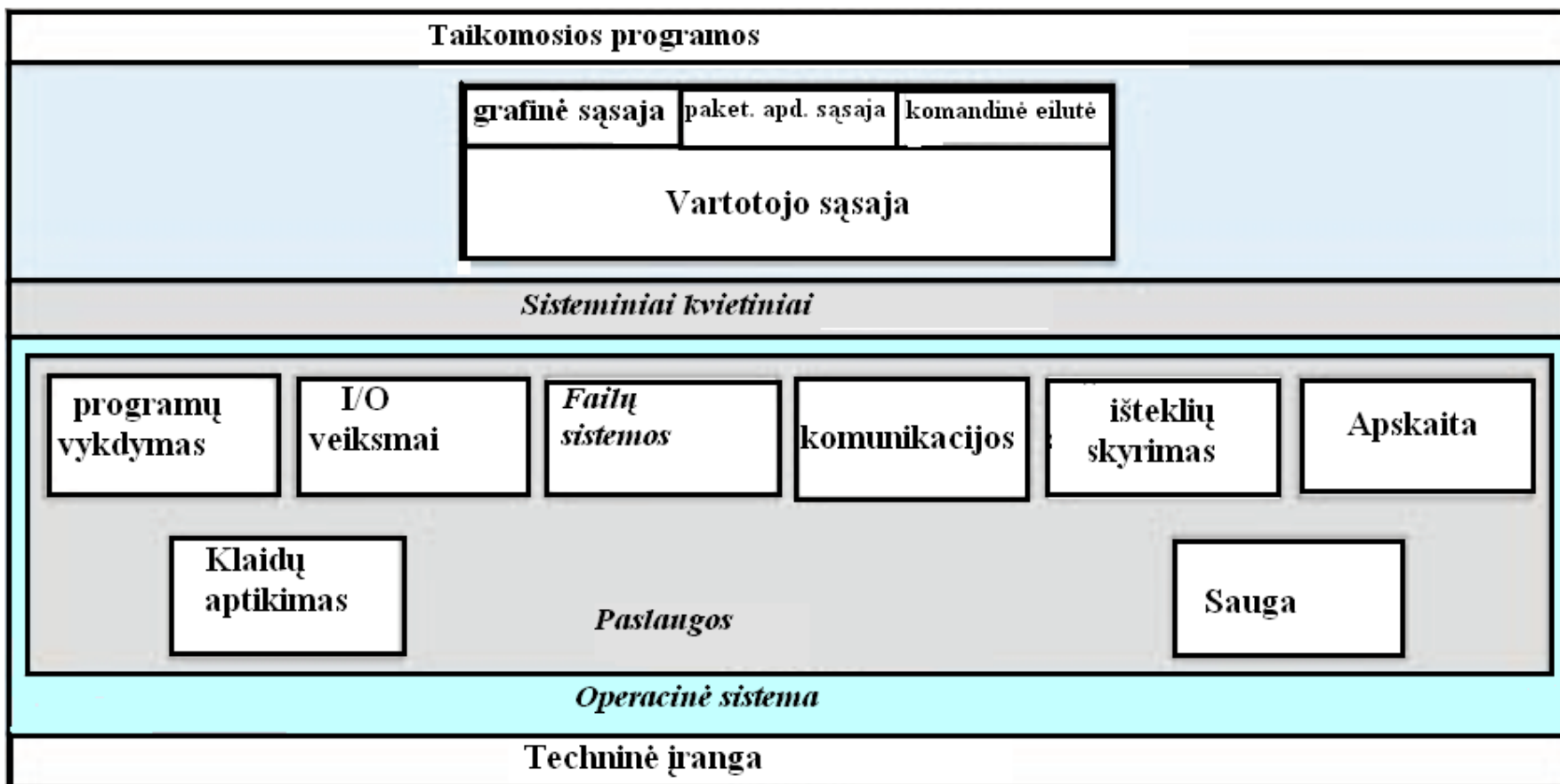
Operacinės sistemos paslaugos

- Programų vykdymas
- Įvedimo-išvedimo veiksmai
- Įvairūs veiksmai susiję su failų sistema
- Komunikacijos
- Klaidų aptikimas

Papildomos operacinės sistemos funkcijos

- Papildomos funkcijos yra skirtos ne tam, kad padėti vartotojui, o tam, kad užtikrinti efektyvų sistemos veikimą.
 - **Resursų priskyrimas**
 - **Apskaita**
 - **Apsauga**

Operacinės sistemos paslaugos



Sisteminės programos

- Sisteminės programos teikia vartotojams eilę bazinių priemonių, kuriomis jie gali naudotis:
 - *Failų redaktoriai*
 - Specialios komandos skirtos *paieškai* failo turinyje ar tiesiog teksto transformavimui.
 - Programinių kalbų palaikymas (*kompiliatoriai, assembleriai,...*)
 - Programų *užkrovėjai, derintojai, ...*
 - *Informacinio* tipo programos (vartotojų informacija, disko informacija, informacija apie datą, laiką, ...)
- Kai kurios iš jų paprasčiausiai realizuoja sąsają su sistemineis kvietiniais, kitos yra sudėtingesnės

Operacinių sistemų projektavimas

Operacinių sistemų projektavimas tai nėra jau galutinai “išspręsta” problema, *nauji vartotojų poreikiai, naujos techninės galimybės reikalauja ir naujų sprendimų.*

- Skirtingų operacinių sistemų vidinė struktūra gali smarkiai skirtis.

Operacinės sistemos savybės

■ Vartotojų tikslai:

- Spartaus panaudojimo įgūdžių įgijimo
- Patogumo naudojime
- Patikimumo
- Saugaus darbo
- Greičio

■ Sistemos tikslai

- Ją turi būti lengva projektuoti, diegti ir palaikyti
- Ji turi būti lanksti:
 - Galimybės keisti
 - Plėtimo galimybės
 - Portatyvumo

Efektyvi:

- Sistemos pralaidumas
- Reakcijos laikas
- Tolygus visų kompiuterio resursų panaudojimas

Projektavimo politika-priemonės

- Projektavimo politika nusakoma atsakymu į klausimą:
 - Kas turi būti padaryta?
- Turimos priemonės:
 - Kaip tai padaryti?
- Politika didžia dalimi apsprendžia tuos tikslus, kurių siekiama projektuojant naujas sistemas (vartotojų patogumas, darbas realiam laike, saugos užtikrinimas,...)
- Rekomenduojama tiksliai apibrėžti ir skirti politiką nuo priemonių, nes tai leidžia maksimalų lankstumą jei politikos sprendimai yra vėliau keičiami.

Operacinių sistemų architektūros

- Monolitinės OS
- Lygiais grįstos sistemos
- Mikrobranduolinės sistemos
- Virtualios mašinos

Monolitinė architektūra

- Monolitinė operacinė sistema
 - Visos operacinės sistemos komponentės randasi branduolyje
 - Bet kuri komponentė gali ***tiesiogiai komunikuoti*** viena su kita
 - Visi operacinės sistemos veiksmai vykdomi su maksimaliom privilegijom

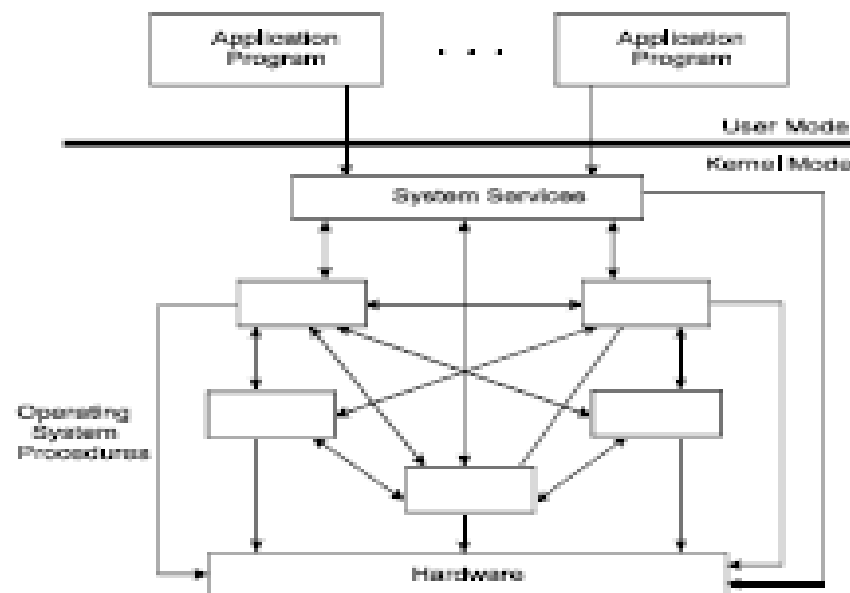
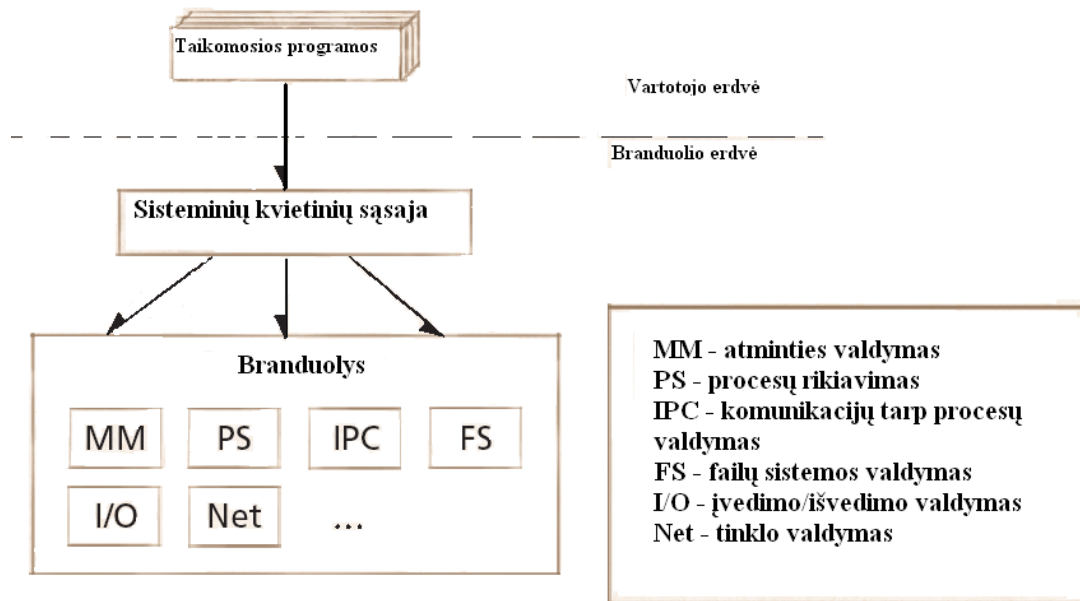


Figure 2.1: Monolithic Operating System

Monolitinė architektūra



Pliusai-minusai

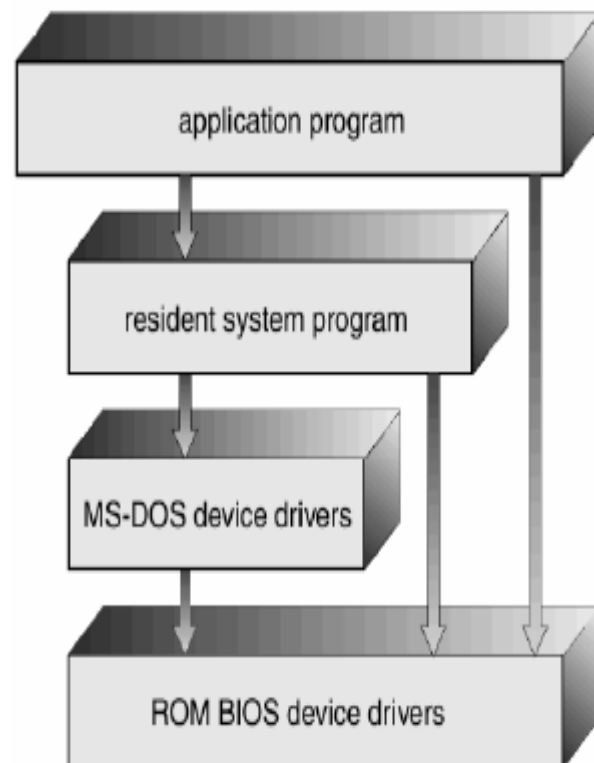
Stipriai surištas kodas, esantis vienoje adresų erdvėje, neatsparus klaidoms

Veikimo efektyvumas

Pirmi projektai – plečiamumo trūkumas, naujesni-galima įkelti dinamiškai vykdomuosius modulius (Linux, Free BSD, Solaris)

MS-DOS sistemos struktūra

- MS-DOS – buvo suprojektuota teikti maksimaliai įmanomą funkcionalumą naudojant mažiausiai erdvės.
 - Nėra suskirstyta į modulius
 - Nors MS-DOS turi tam tikrą struktūrą, jos funkcionalumo lygiai nėra griežtai atskirti.
 - 512 kB RAM, 5 mB kieto disko erdvės (priklausomai nuo versijos)
 - FAT failų sistema
 - Komandų interpretatorius

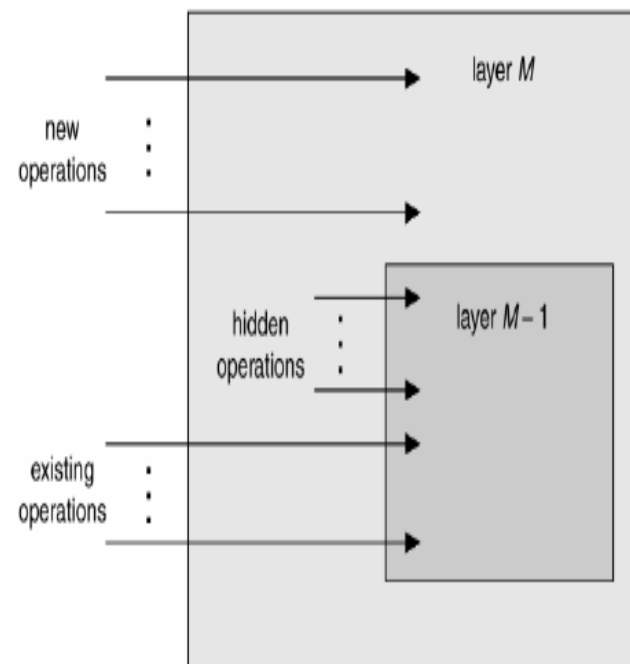


UNIX sistemos struktūra

(the users)		
shells and commands compilers and interpreters system libraries		
<i>system-call interface to the kernel</i>		
signals terminal handling character I/O system terminal drivers	file system swapping block I/O system disk and tape drivers	CPU scheduling page replacement demand paging virtual memory
<i>kernel interface to the hardware</i>		
terminal controllers terminals	device controllers disks and tapes	memory controllers physical memory

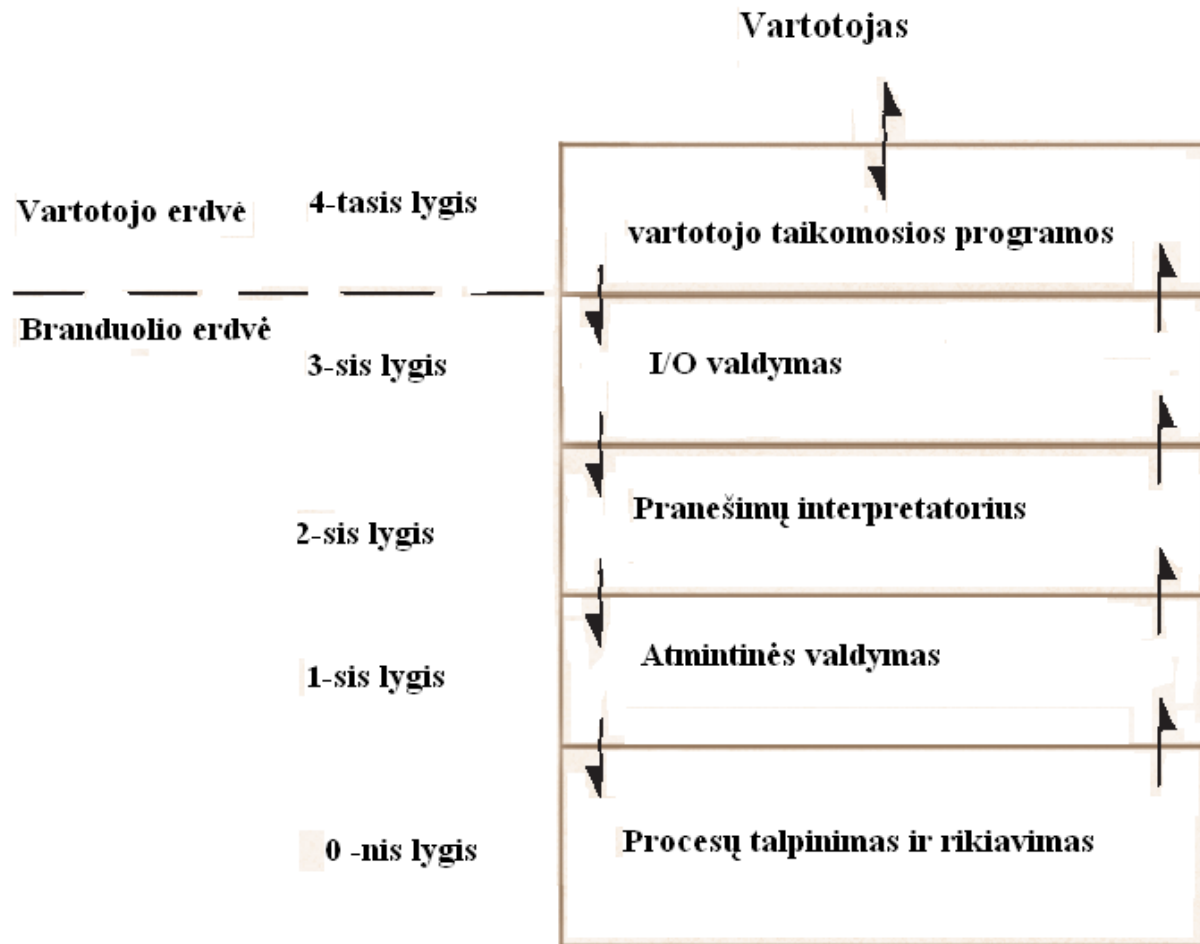
Lygių architektūra

- Operacinės sistemos projektuojamos naudojant lygių koncepciją
 - Grupuojamos komponentės, kurios vykdo panašias funkcijas į lygius
 - Kiekvienas lygis komunikuoja tik su gretimais aukščiau bei žemiau esančiais lygiais.
 - Procesų užklausoms gali tekti pereiti keletą lygių kol jos bus įvykdytos
 - Sistemos pralaidumas gali būti žemesnis nei monolitinių branduolių atveju
 - Galima įtraukti papildomus metodus, kurie perduotų bei kontroliuotų duomenis.

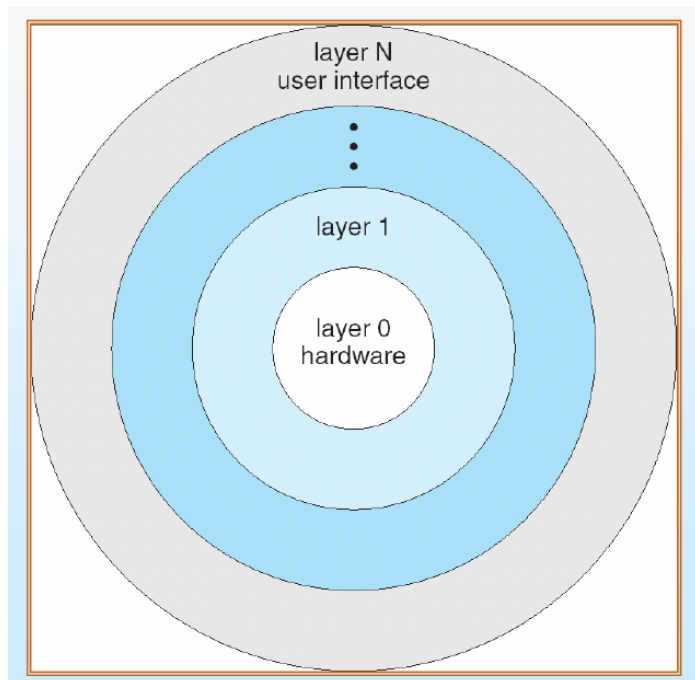


Lygiais grįsta architektūra

(THE –Dijkstra, 1968m.)



Lygių architektūra



MULTICS– tai žymiausia OS ,
turinti 8 lygius (*protection rings*) ,
kurių ribas galima kirsti tik
naudojant specializuotas
komandas

Siekama pasiekti geresnės
kokybės lygiams suteikiant
skirtingas **privilegijas**:

***Labiausiai
privilegiuotas lygis
apima kodą, kuris tvarko
pertrauktis, konteksto
perjungimą.***

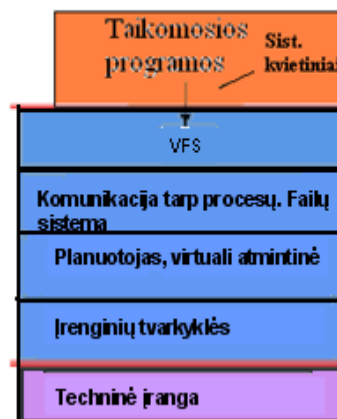
Aukštesni lygiai susiję su
įrenginių **tvarkyklėmis**,
atmintinės valdymu,
failų sistema, vartotojų
sąsaja.

Mažiausią privilegiją turi
taikomųjų programų lygis

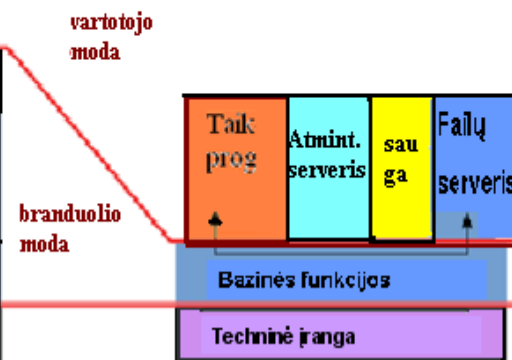
Mikrobranduolio architektūra

- **Branduolyje palikta nedaug paslaugų**
 - Vartotojo lygmens serveriai
 - Komunikacija - naudojant pranešimų siuntimą
- Naudojamas modulinis projektavimas
- Privalumai:
 - Vienoda sąsaja tiek su taikomosiomis programomis tiek su vartotojo lygio paslaugomis
 - Plėtimosi, portatyvumo, keičiamumo galimybės
 - Labiau patikima
 - Saugesnė

Monolitinio branduolio operacinė sistema

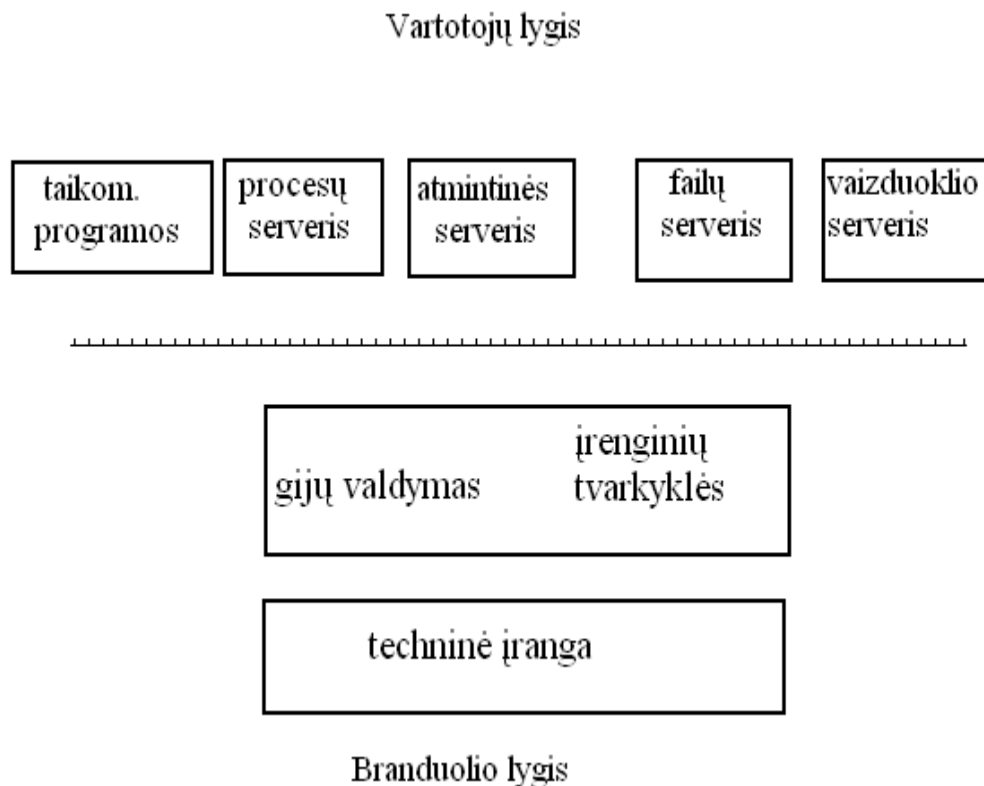


Mikrobranduolinė operacinė sistema



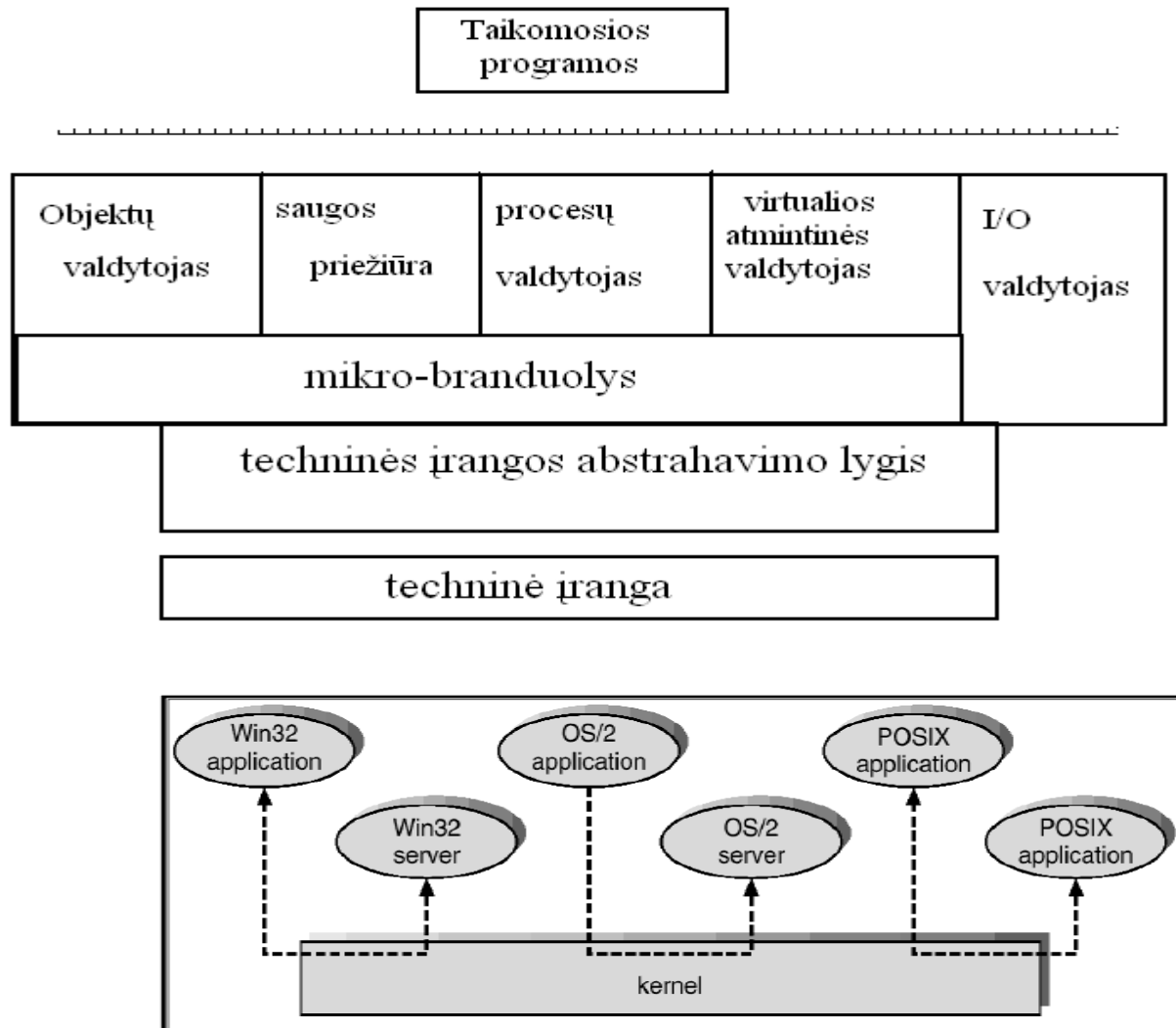
Mach operacinė sistema

- Viena iš pirmųjų mikrobranduolinių OS, sukurta Canegie Mellon Universitete
- Dauguma kitų naudoja Mach koncepcijas.
- Mach buvo suprojektuota **lygiagrečių ir paskirstytų skaičiavimų** palaikymui.
- Šio mikrobranduolio pasekėjai yra modernių operacinių sistemų MacOS X, GNU Hurd branduoliai



Windows NT

Naudoja modifikuotą mikro-branduolinę architektūrą (kai kurios paslaugos branduolio lygmenyje funkcionavimo efektyvumui padidinti)

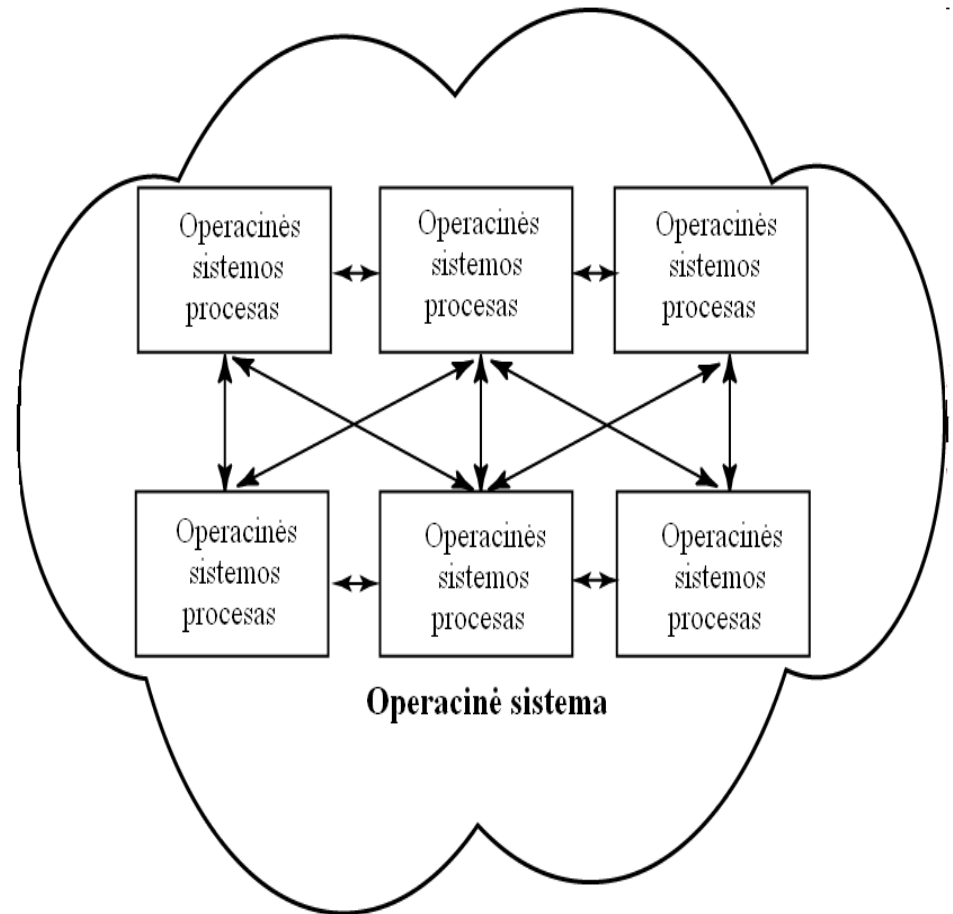


Mikrobranduolinių sistemų pavyzdžiai

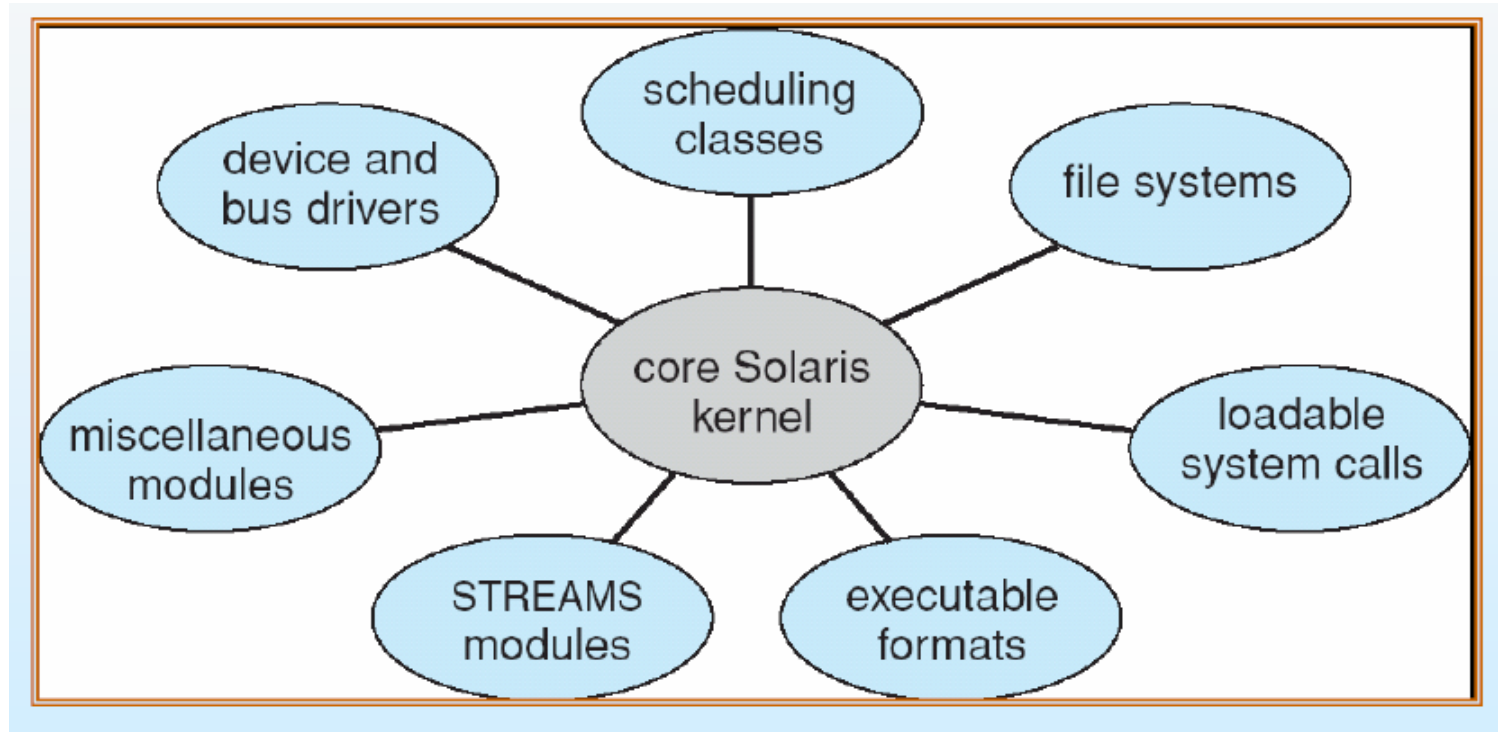
- AIX
- AmigaOS
- Amoeba
- Chorus microkernel
- EROS
- Haiku
- K42
- LSE/OS (nanobranduolinė)
- KeyKOS (nanobranduolinė)
- L4 mikrobranduolinė šeima
- Mach, naudojama GNU Hurd
- NEXTSTEP, OPENSTEP IR MacOS X
- MERT
- Minix
- MorphOS
- NewOS
- QNX
- Phoenix-RTOS
- RadiOS
- Spring OS
- VSTa
- SymbianOS
- Android
- OSE

Modulinės operacinės sistemos

- Taikomas objektinio projektavimo principas: sistema interpretuojama kaip tarpusavyje sąveikaujančių objektų visuma
- Kiekviena OS komponentė yra interpretuojama kaip atskiras modulis
- Kiekvienas modulis bendrauja su kitu per žinomą sąsają
- Lankstesnė, nei lygių architektūra



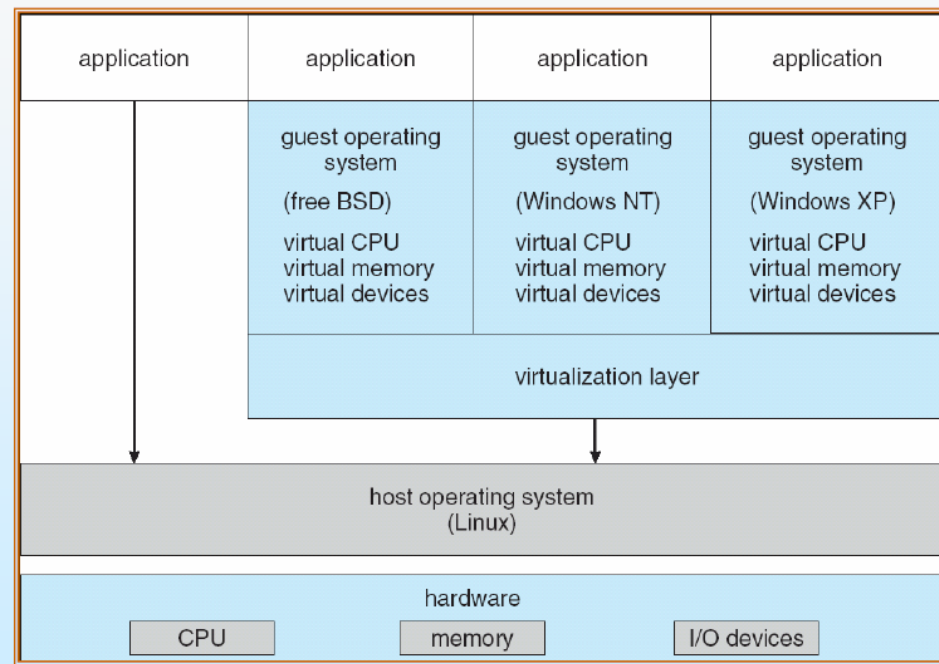
Solaris modulinė OS



Kai kurie autoriai skaito, kad geriausia metodologija projektuojant OS yra panaudojant objektinį programavimą sukurti moduliais grįstą branduolį. Moduliai –dinamiškai įkeliami(pav., Solaris, Linux, Mac OS X).

Virtualios mašinos

- Virtualiose mašinose taikomas lygiais grįstas principas
- Prisilaikant šio principo techninė įranga ir operacinės sistemos branduolys yra traktuojami kaip “techninė įranga”
- Fiziniai resursai yra dalinami

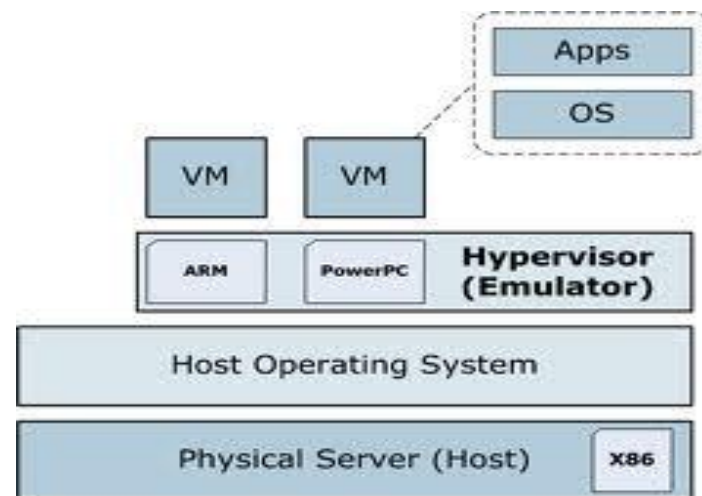
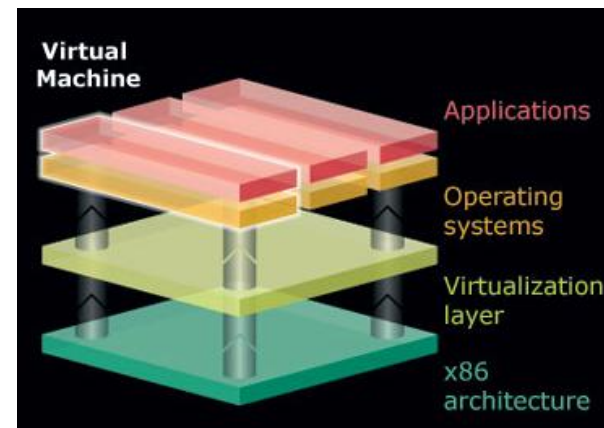


VMware Architektūra

Virtualizacija

- Pilna virtualizacija:

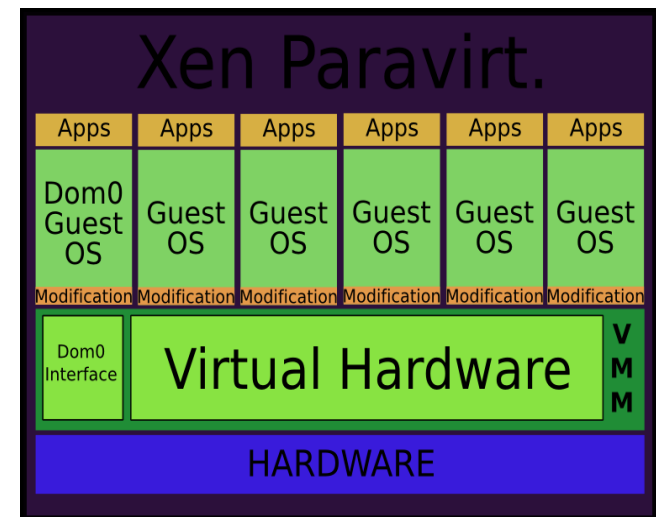
- ☐ Pilna virtualizacija žavi tuo, kad ji įgalina paleisti kelias skirtingas svečio teisēm veikiančias OS jų nemodifikuojant



Virtualizacija

■ Paravirtualizacija

- Paravirtualizacijos esmė yra tame, kad egzistuoja labai paprasta bei minimaliai naudojanti bendrus kompiuterio išteklius programinė įranga, kuri turi tiesioginę sąsają su technine įranga bei tiekia paslaugas virtualiems kompiuteriams. Pastarųjų operacinės sistemos veikia svečio teisėmis.



Virtualizacija

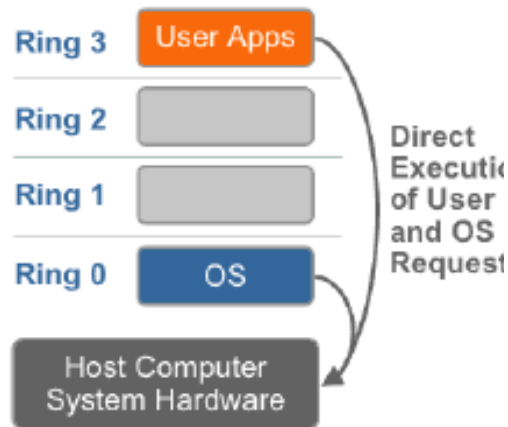
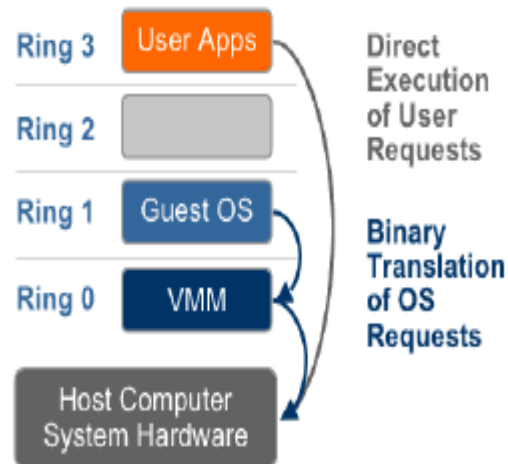
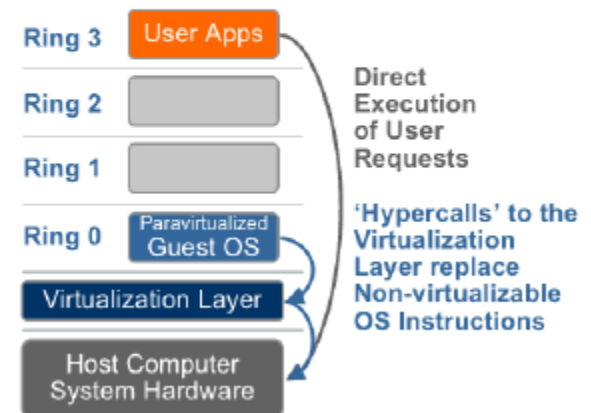


Figure 4 – x86 privilege level architecture without virtualization

Full Virtualization using Binary Translation

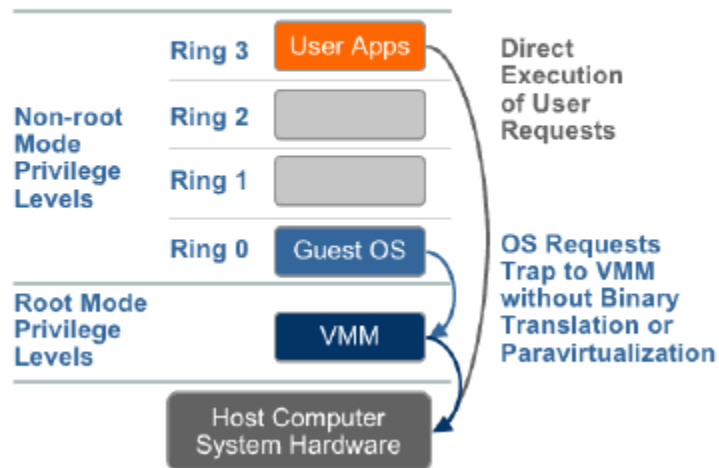


OS Assisted Virtualization or Paravirtualization

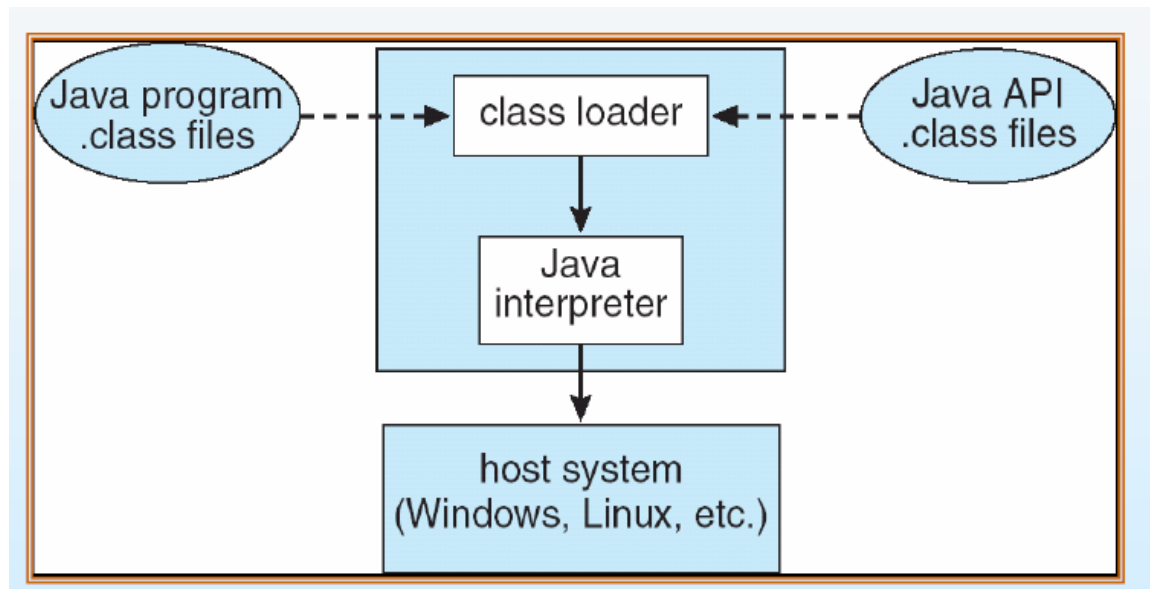


Virtualizacija

Hardware Assisted Virtualization



Java virtuali mašina



Nepriklauso nuo techninės architektūros

Java virtualią mašiną sudaro:

- Klasių įkėlimo programa

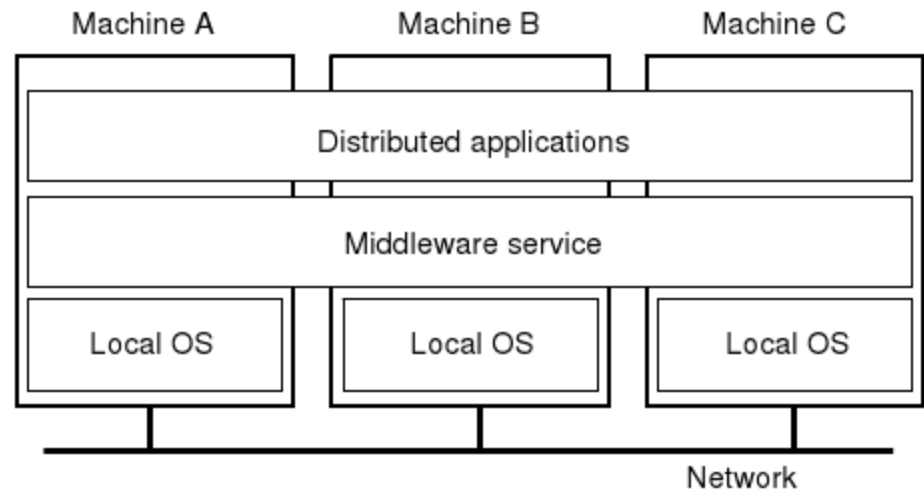
- Java interpretatorius

Tinklinės operacinės sistemos

- Tinklinės operacinės sistemos
 - Veikia viename kompiuteryje
 - Leidžia savo procesams pasiekti resursus nutolusiuose kompiuteriuose

Paskirstytosios operacinės sistemos

- Viena operacinė sistema
- Valdo išteklius, esančius daugiau nei vienoje kompiuterinėje sistemoje
- Vartotojams sudaromas vienos sistemos įvaizdis



Operacinių sistemų architektūros

- Šiuolaikinės operacinės sistemos yra gana sudėtingos:
 - Jos teikia daugelį paslaugų
 - Jos palaiko įvairią gausią techninės bei programinės įrangos įvairovę.
- Atitinkama operacinių sistemų architektūra padeda sprendžiant įvairius operacinei sistemai keliamus uždavinius:
 - Padeda organizuoti operacinės sistemos komponentus.
 - Nusako privilegijas, kuriomis naudojasi įvairūs operacinės sistemos komponentai jų vykdymo metu.
- Projektuotojai ir toliau ieško tokių operacinių sistemų struktūrų, kurios geriausiai tiktų palaikant naujus taikomųjų programų tipus ir darbo tinkle poreikius.