

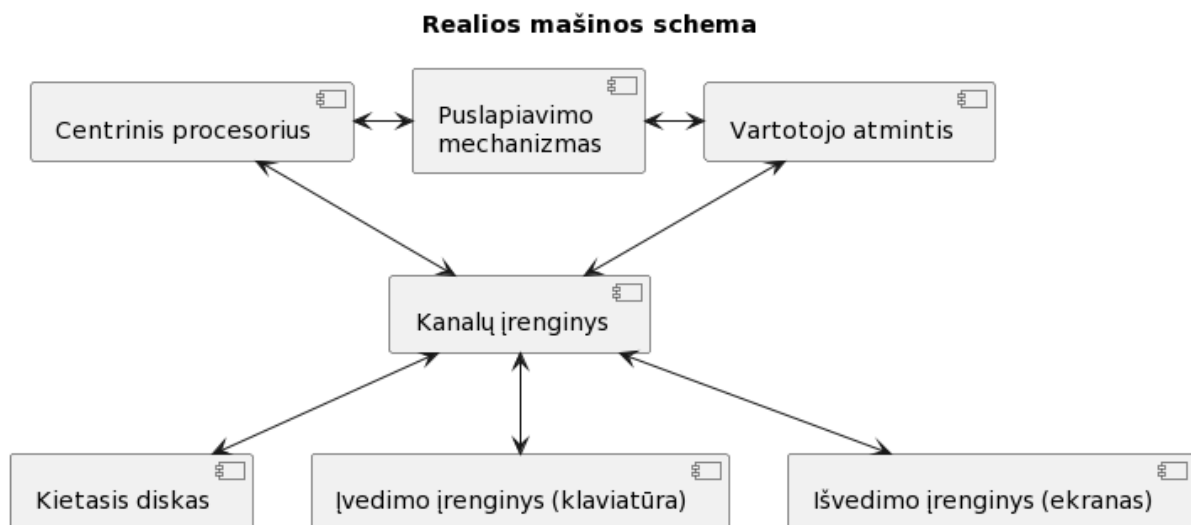
# Operacinės sistemos

## Virtualios mašinos bei realios mašinos apibrėžimas ir realizacija

### Realios mašinos apibrėžimas

Projektuojama reali mašina turės šiuos komponentus:

1. Centrinis procesorius
2. Atmintis
3. Puslapiavimo mechanizmas
4. Duomenų perdavimo kanalų įrenginys
  - Kietasis diskas
  - Įvedimo įrenginys (klaviatūra)
  - Išvedimo įrenginys (ekranas)



Komponentų paskirtis ir jų dalių apibrėžimai:

1. **Centrinis procesorius** - Vykdo programos instrukcijas. Procesorius gali dirbti dvejais režimais:
  - Supervizoriaus - Šiame režime procesorius gali įvykdyti bet kokią architektūros palaikomą instrukciją
  - Vartotojo - Šiame režime procesorius vykdo instrukciją imituodamas virtualų procesorių

Centrinio procesoriaus registrai:

- MODE - 1 baido procesoriaus darbo režimo registras
  - PC - 4 baitų programos komandos registras
  - PTR - 4 baitų puslapių lentelės registras. Prieš pradėdant dirbti virtualiai mašinai, šis registras yra nustatomas reikalinga reikšmė
  - SP - 4 baitų steko viršūnės adreso registras
  - TI - 1 baido taimerio registras
  - SI - 1 baido supervizorinių pertraukimų registras
  - PI - 1 baido programinių pertraukimų registras
2. **Atmintis** - Realios mašinos atmintis susideda iš:
- Vidinės atminties - Iš viso skiriama 1024 puslapiai (puslapio dydis - 256 žodžiai, žodžio ilgis - 4 baitai)
  - Išorinės atminties (Kietasis diskas) - Skiriami 1024 sektoriai po 256 žodžius
3. **Kanalų įrenginys** - Skirtas darbui su išorine atmintimi ir valdyti įvedimo/išvedimo įrenginiams. Kanalų įrenginys turi šiuos kanalus bendrauti su įrenginiais:
- CH1 - Jungia kietąjį diską
  - CH2 - Jungia įvedimo įrenginį (klaviatūrą)
  - CH3 - Jungia išvedimo įrenginį (ekraną)
4. **Puslapiavimo mechanizmas** - Realios mašinos vartotojo atminčiai viso yra skiriami 1536 puslapiai. Kiekvienai virtualiai mašinai bus išskiriami 16 puslapių, kurie bus sunumeruoti nuo 0 iki 15. Tam, kad turėdami virtualios mašinos puslapį rastumė tą patį puslapį realioje mašinoje naudojame puslapiavimo mechanizmą.
5. **Timeris** - Skirtas geriau išlygiagretinti užduotims. Kai programa pradeda darbą registrui TI yra priskiriama tam tikra reikšmė, įvykdžius virtualios mašinos instrukciją ši reikšmė yra sumažinama pagal instrukcijos momentų trukmę. Kai TI registro reikšmė tampa lygi 0, mikrokomanda Test() aptinka taimerio pertraukimą.
6. **Pertraukimai** - Tai yra tam tikri signalai apie specialius įvykius. Pertraukimai gali būti aptikti tik procesoriui dirbant vartotojo režime. Įvykus pertraukimui virtualios mašinos registrų reikšmės yra išsaugomos ir procesorius yra perjungiamas į supervizoriaus režimą, kuriame nustatomas pertraukimo pobūdis bei išskviečiama pertraukimą apdorojanti programa. Vėliau valdymas grįžta į virtualią mašiną (registrai atstatomi ir procesorius perjungiamas į vartotojo režimą) ir operacinė sistema sprendžia, ką daryti toliau. Pertraukimus aptinka mikrokomanda Test()

Galimi trijų rūšių pertraukimai:

- **Timerio** - registras yra TI. Galimi atvejai:
  - $TI = 0$  - taimerio reikšmė lygi 0
- **Programiniai** - registras yra PI. Galimi atvejai:
  - $PI = 1$  - neteisingas adresas
  - $PI = 2$  - neteisingas operacijos kodas
- **Supervizoriniai** - registras yra SI. Galimi atvejai:
  - $SI = 1$  - komanda GD
  - $SI = 2$  - komanda PD

- SI = 3 - komanda HALT

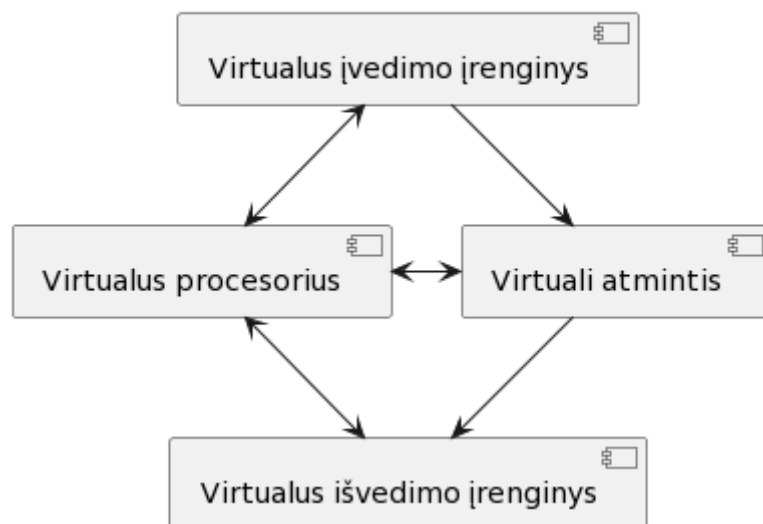
## Pertraukimų apdorojimo mechanizmas

Pertraukimai apdorojami iškviečiant nurodytą procedūrą susijusią su pertraukimu, grįžus iš šios procedūros pertraukimas yra laikomas apdorotu ir pertraukimo registro reikšmė yra atstatoma į neturinčią jokio pertraukimo registro reikšmę.

## Virtualios mašinos apibrėžimas

Virtuali mašina - tai realios mašinos modelis, kuris veikia kaip tam tikras tarpininkas. Ji smarkiai supaprastinta tiek ir programų rašymą tiek ir pačią realizaciją. Virtualios mašinos pagrindinė paskirtis vykdyti vartotojo programą.

**Virtualios mašinos schema**



Virtualios mašinos komponentai ir jų paskirtis:

1. **Virtuali atmintis** - Virtualios mašinos atmintis susideda iš 16 puslapių, kur vienas puslapio dydis yra 256 žodžiai. Virtuali atmintis yra suskirstyta pagal paskirtį naudojant šią schemą:
  - Duomenys - Skiriami 7 puslapiai (nuo 0 iki 6)
  - Kodas - Skiriami 7 puslapiai (nuo 7 iki 13)
  - Stekas - Skiriami 2 puslapiai (nuo 14 iki 15)
2. **Virtualus procesorius** - Pagrindinė virtualaus procesoriaus paskirtis vykdyti programą, kuri yra virtualioje atmintyje. Virtualus procesorius yra supaprastinta realios mašinos centrinio procesoriaus versija. Šio procesoriaus registrai:
  - PC - Komandų skaitliukas rodantis kitos vykdomos instrukcijos adresą
  - SP - Steko viršūnės indekso registras

Virtualios mašinos komandų sistema:

- ADD - Iš steko pašalina du viršutinius elementus ir jų šių elementų sumos rezultatą įrašo į steko viršūnę
- INC - Steko viršūnėje esančią reikšmę padidina 1
- SUB - Iš steko pašalina du viršutinius elementus ir iš elemento, kuris buvo steko viršūnėje atima elementą, kuris buvo antras nuo viršūnės steko elementas. Šio veiksmo rezultatą įrašo įterpia į steko viršūnę
- DEC - Steko viršūnėje esančią reikšmę sumažina 1
- MUL - Iš steko pašalina du viršutinius elementus ir jų šių elementų daugybos rezultatą įrašo į steko viršūnę
- DIV - Iš steko pašalina du viršutinius elementus elementą, kuris buvo steko viršūnėje padalija iš elemento, kuris antras nuo steko viršūnės elemento. Šio veiksmo rezultatą įrašo į steko viršūnę
- MOD - Iš steko pašalina du viršutinius elementus elementą, kuris buvo steko viršūnėje apskaičiuoja liekaną iš elemento, kuris antras nuo steko viršūnės elemento. Šio veiksmo rezultatą įrašo į steko viršūnę
- PUSH VALUE
  - Kai VALUE - Skaičius skaičiaus reikšmę įterpia į steko viršūnę
  - KAI VALUE - Tekstas iš duomenų segmento esančia reikšmę šiuo pavadinimu įterps į steko viršūnę
- POP [VALUE]
  - Kai VALUE nėra iš steko pašalins reikšmę esančia viršūnėje
  - Kai VALUE yra į reikšmę pavadinimu VALUE duomenų segmente perneš reikšmę esančia steko viršūnėje
- CMP - Iš steko pašalina du viršutinius elementus ir juos palygina. Jei jie buvo lygūs į steką įrašo reikšmę 0, jei elementas buvęs steko viršūnėje buvo mažesnis įrašo -1, kitu atveju į steką įrašo reikšmę 1
- MOV DEST VALUE
  - Kai VALUE - Skaičius duotąją reikšmę įterps į duomenų segmente esančią reikšmę
  - Kai VALUE - Tekstas iš duomenų segmento pavadinimu VALUE įrašys reikšmę kuri bus DEST vietoje
- JMP LABEL - Valdymas yra perduodamas kodo sričiai esančiai adresu LABEL
- JE LABEL - Iš steko yra pašalinama reikšmė, jeigu ji yra lygi 0, valdymas yra perduodamas kodo sričiai esančiai adresu LABEL
- JNE LABEL - Iš steko yra pašalinama reikšmė, jeigu ji nėra lygi 0, valdymas yra perduodamas kodo sričiai esančiai adresu LABEL
- JB LABEL - Iš steko yra pašalinama reikšmė, jeigu ji yra mažesnė už 0, valdymas yra perduodamas kodo sričiai esančiai adresu LABEL
- JAL LABEL - Iš steko yra pašalinama reikšmė, jeigu ji yra didesnė už 0, valdymas yra perduodamas kodo sričiai esančiai adresu LABEL
- HALT - Programos sustabdymo komanda

- PRINTN - Išspausdina reikšmę esančią steko viršūnėje kaip skaičių

## Programos segmentai

Kodo segmentas turi prasidėti parašant \$CODE, duomenų segmentas turi prasidėti parašant \$DATA. Kodo segmente yra galimos tik duotosios virtualaus procesoriaus komandos.

Duomenų segmente yra galimos tik duomenų aprašymo komandos. Jos susideda:

1. LABEL DW VALUE - LABEL kintamojo pavadinimas, VALUE - šio kintamojo reikšmė.

## Virtualaus ir realaus adresų konversija

Realios mašinos vartotojo atminties dydis yra 1536 puslapiai, kuriuos gali naudoti vartotojo programos. Virtuali mašina šiems puslapiams naudoja savo virtualų adresavimą, kuris turi realaus adreso atitikmenį. Realaus adreso apskaičiavimui naudojamas puslapiavimo mechanizmas. Kiekvienai virtualiai mašinai sukuriamą puslapio lentelė (16 žodžių) vartotojo atmintyje, kurioje virtualių adresų tvarka surašyti puslapiavimo lentelės įrašai. Atitinkamame lentelės įrašė laikysime virtualaus puslapio realų adresą realioje atmintyje

# Operacinės sistemos apibrėžimas ir realizacija

## Procesai

### Procesų bendras aprašymas

**Multiprograminės operacinės sistemos (MOS)** pasižymi sugebėjimu naudotis vienu centriniu procesoriumi atlikti kelias užduotis vienu metu. Realybėje vienu metu procesoriumi gali naudotis tik vienas procesas, bet dažniausiai vieno proceso vykdymo laikas netrunka ilgiau dešimtosios sekundės ir prieinama prie kito. Toks greitas procesų aptarnavimas suteikia lygiagretaus veikimo įspūdį.

**Procesas** - tai vykdoma programa, kartu su esamomis registrų reikšmėmis ir savo kintamaisiais. Kiekvienam procesui yra sukuriama atskiras virtualus procesorius. Programa nuo proceso skiriasi tuo, kad procesas - tai kokioje nors veiklumo stadijoje esanti programa, o programa - tik tam tikras baitų rinkinys. Veiklumo stadiją proceso deskriptorius. Jame laikomi visi procesui reikalingi parametrai - virtualaus procesoriaus registrų reikšmės, procesui reikalingi kintamieji.

Sistemos veikimo eigoje yra paleidžiamas vienas šakninis procesas, iš kurio seks likusieji.

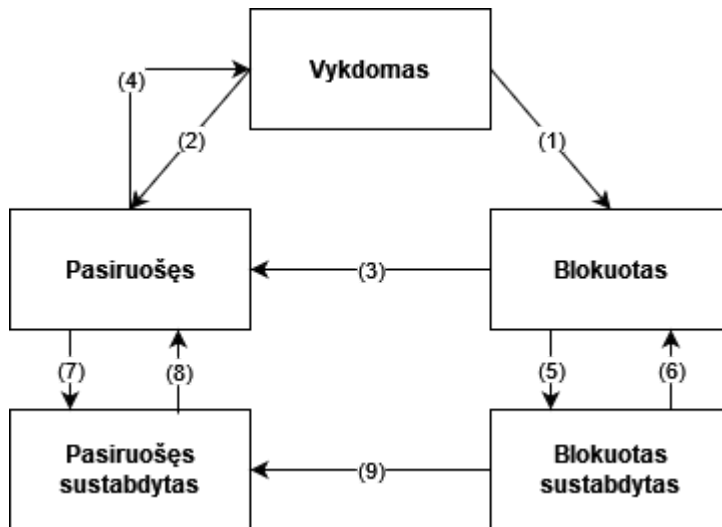
Procesai gali būti sisteminiai arba vartotojo. Sisteminiai procesai atlieka su MOS darbu susijusias užduotis, o vartotojo procesai - rūpinasi vartotojo užduoties atlikimu.

### Procesų būsenos

Procesas gali gauti procesorių tik tada, kai jam netrūksta jokio kito resurso. Procesas gavęs procesorių tampa vykdomu. Procesas, esantis šioje būsenoje, turi procesorių, kol sistemoje neįvyksta pertraukimas arba einamasis procesas nepaprašo kokio nors resurso. Procesas blokuojasi priverstinai (nes jis negali tęsti darbo be reikiamo resurso). Tačiau, jei procesas nereikalauja jokio resurso, iš jo gali būti atimamas procesorius, todėl, kad pernelyg ilgai dirbo. Galime išskirti procesų būsenas:

- **Vykdomas** - Turi procesorių
- **Blokuotas** - Prašo resurso (išskyrus procesorių)
- **Pasiruošęs** - Vienintelis trūkstamas resursas yra procesorius
- **Sustabdytas** - Kito proceso sustabdytas procesas

## Proceso būsenos veikimo schema



1. Vykdomas procesas blokuojasi jam prašanti ir negavus resurso
2. Vykdomas procesas tampa pasiruošusiu atėmus iš jo procesorių dėl kokios nors priežasties (išskyrus resurso negavimą)
3. Blokuotas procesas tampa pasiruošusiu, kai yra suteikiamas reikalingas resursas
4. Procesas tampa vykdomu, kai jam yra suteikiamas procesorius
5. Procesas gali tapti sustabdytu blokuotu, jei einamasis procesas jį sustabdo, kai jis jau ir taip yra blokuotas
6. Procesas tampa blokuotu, kai einamasis procesas nuima būseną sustabdytas
7. Procesas gali tapti pasiruošusiu sustabdytu, jei einamasis procesas jį sustabdo, kai jis yra pasiruošęs
8. Procesas tampa pasiruošusiu, jei einamasis procesas nuima būseną sustabdytas
9. Procesas tampa pasiruošusiu sustabdytu, jei procesui yra suteikiamas reikalingas resursas

## Planuotojas

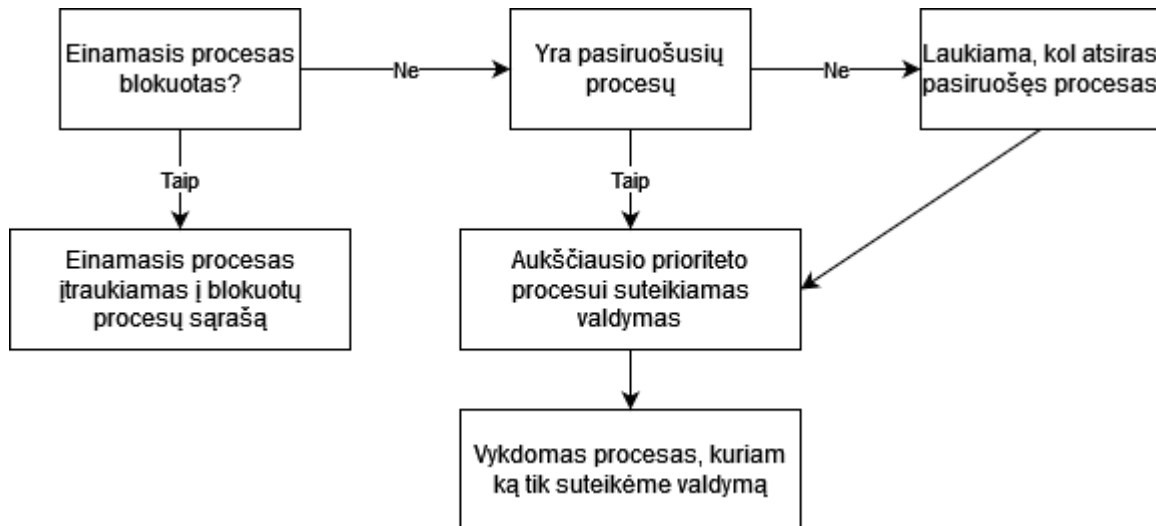
**Planuotojas** - tai procesoriaus resurso paskirstytojas. Jis dirba su branduolio primityvais, pasiruošusių procesų sąrašu, procesų deskriptoriais, bei procesoriaus resurso deskriptoriumi. Planuotojas šį darbą atlieka atimdamas procesorių iš vieno proceso ir perduodamas jį kitam procesui.

Planuotojo tikslai:

- Pasirūpinti, kad kiekvienas procesas tinkamą laiko tarpą turėtų procesorių (jei procesas trunka ilgai, procesorius iš šio proceso gali būti atimtas)
- Laikyti procesoriaus apkrovą netoli 100%
- Pasirūpinti, kad atsakymo laikas vartotojams būtų kuo trumpesnis
- Siekti kuo didesnio efektyvumo

Planavimo įgyvendinimui yra naudojami įvairūs algoritmai, vienas iš jų yra prioritetais besiremiantis modelis: kiekvienam procesui yra suteikiamas prioritetas iš sutarto intervalo ir vykdomas tas procesas, kurio prioritetas yra didžiausias.

## Planuotojo veikimo schema



## Procesų primitivai

Procesų primityvų paskirtis - pateikti vienodą ir paprastą vartotojo sąsają darbui su procesais. Kiekvieno primitivo programos gale yra kviečiamas planuotojas. Galime aprašyti primitivus:

- **Kurti procesą** - Šiam primitivui perduodama nuoroda į jo tėvą, jo pradinė būsena, prioritetas, perduodamų elementų sąrašas ir išorinis vardas. Pačio primitivo viduje vyksta proceso kuriamasis darbas. Jis yra registruojamas bendrajame procesų sąraše, tėvo-sūnų sąraše, skaičiuojamas vidinis identifikacijos numeris, sukuriama jo vaikų procesų sąrašas, sukuriama resursų sąrašas
- **Naikinti procesą** - Pradedama naikinti proceso sukurtus resursus ir vaikus. Vėliau išmetamas iš tėvo sukurtų procesų sąrašo. Galiausiai naikinami visi jam perduoti resursai ir proceso deskriptorius yra sunaikinamas
- **Sustabdyti procesą** - Keičiama proceso būsena iš blokuotos į blokuotą sustabdytą arba iš pasiruošusios į pasiruošusią sustabdytą
- **Aktyvuoti procesą** - Keičiama proceso būsena iš blokuotos sustabdytos į blokuotą arba į pasiruošusią iš pasiruošusios sustabdytos



# Resursai

## Bendras resursų aprašymas

Resursas yra tai, dėl ko varžosi procesai. Dėl resursų trūkumo procesai blokuojasi, gavę reikiamą resursą procesai tampa pasiruošusiais. Resursus galima suskirstyti į:

- **Statiniai** - Kuriami sistemos paleidimo metu. Tai mašinos resursai, tokie kaip procesorius, vidinė ir išorinė atmintys ar kiti resursai, kurie sistemos veikimo metu nėra naikinami
- **Dinaminiai** - Kuriami ir naikinami sistemos darbo metu. Šie resursai naudojami kaip pranešimai. Kartu su jais gali ateiti naudinga informacija. Kartais šio tipo resursas pats yra pranešimas

## Resursų primitivai

Resursų primitivai - pagrindiniai veiksmai su resursais. Galima išskirstyti į:

- **Kurti resursą** - Resursus kuria tik procesas. Resurso kurimo metu perduodami kaip parametrai: nuoroda į proceso kūrėją, resurso išorinis vardas. Resursas kurimo metu yra: pridedamas prie bendro resursų sąrašo, pridedamas prie tėvo sukurtų resursų sąrašo, jam priskiriamas unikalus vidininis vardas, sukuriama resurso elementų sąrašas ir sukuriama laukiančių procesų sąrašas
- **Naikinti resursą** - Resurso deskriptorius yra išmetamas iš jo tėvo sukurtų resursų sąrašo, naikinamas jo elementų sąrašas, atblokuojami procesai, laukiantys šio resurso ir galiausiai naikinamas pats deskriptorius
- **Prašyti resurso** - Šį primitivą kartu su primitivu "atlaisvinti resursą" procesai naudoja labai dažnai. Procesas, iškvietęs šį primitivą yra užblokuojamas ir įtraukiamas į to resurso laukiančių procesų sąrašą. Sekantis šio primitivo žingsnis yra kviešti resurso paskirstytoją
- **Atlaisvinti resursą** - Šį primitivą kviečia procesas, kuris nori atlaisvinti jam nereikalingą resursą arba tiesiog perduoti pranešimą ar informaciją kitam procesui. Resurso elementas, primitivui perduodamas kaip funkcijos parametras, yra pridedamas prie resurso elementų sąrašo. Šio primitivo pabaigoje yra kviečiamas resursų paskirstytojas

# Procesų detalus aprašymas

Galime apibrėžti šį resursų paketą:

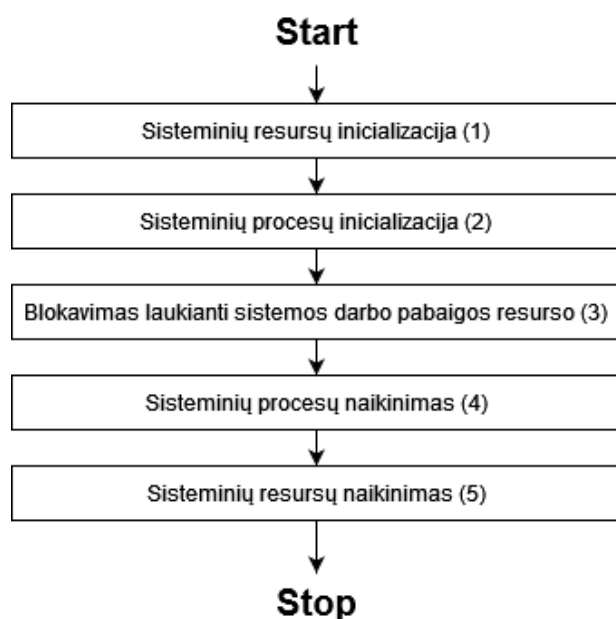
- **StartStop** - Šakninis procesas, sukuriantis bei naikinantis sisteminius procesus ir resursus
- **ReadFromInterface** - Užduoties nuskaitymo iš įvedimo srauto procesas
- **JCL** - Suskaidyti užduoties programą blokais ir organizuoti juos kaip resursus
- **Interrupt** - Apdoroti virtualios mašinos pertraukimą sukėlusią situaciją
- **MainProc** - Valdyti JobGovernor procesus
- **JobGovernor** - Virtualios mašinos proceso tėvas. Skirtas valdyti virtualios mašinos proceso darbą
- **VirtualMachine** - Atsakyti už vartotojiškos programos vykdymą

Beveik visi procesai yra sukuriami sistemos darbo pradžioje proceso StartStop. StartStop nekuria tik 2 procesų - JobGovernor (kiekvienai naujai vartotojo užduočiai MainProc kuria po naują procesą JobGovernor) Ir VirtualMachine, kurį kuria JobGovernor.

## Procesas StartStop

Šis procesas yra atsakingas už sistemos darbo pradžią ir pabaigą. Įjungus kompiuterį, šis procesas pasileidžia automatiškai. Šio proceso paskirtis, sisteminių procesų bei resursų sukūrimas.

Šio proceso diagrama:



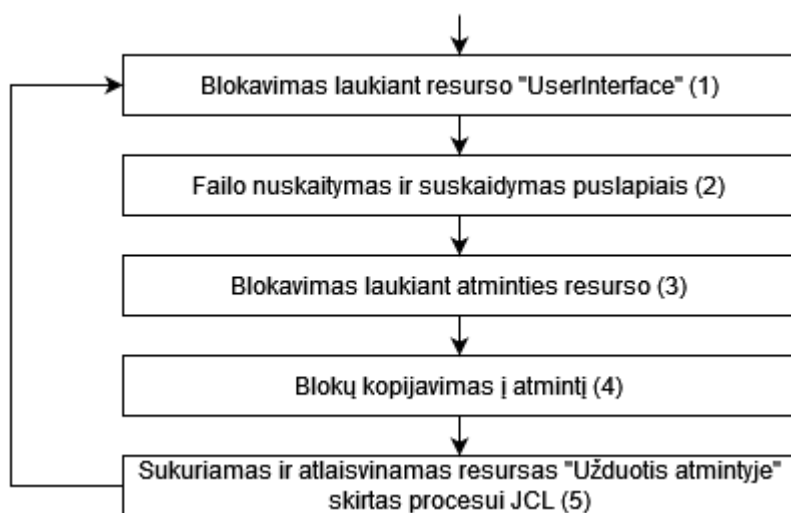
Proceso būsenos paaiškinimas:

1. Procesas, gavęs procesorių, savo darbą pradeda sukurdamas visus sisteminius resursus
2. Sukūręs resursus, procesas kuria sisteminius procesus (t.y. tokius procesus, kurie bus aktyvūs visą OS gyvavimo laiką)
3. Prašo resurso "POS pabaiga" (t.y. procesas blokuojasi iki tol kol bus gautas pranešimas apie POS darbo pabaigą)
4. Visų sisteminių procesų naikinimas
5. Visų sisteminių resursų naikinimas. OS darbo pabaiga

## Procesas ReadFromInterface

Šį procesą kuria ir naikina procesas StartStop. Šio proceso paskirtis - gavus informaciją iš įvedimo srauto ir atlikus pirminį jos apdorojimą, atiduoti informaciją tolesniam apdorojimui, kurį atliks procesas JCL.

Šio proceso diagrama:



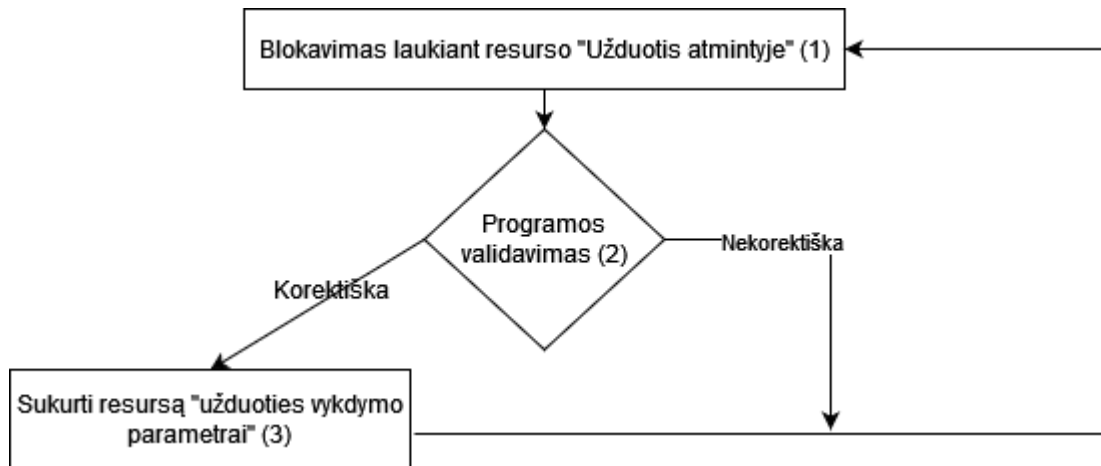
Proceso būsenos paaiškinimas:

1. Laukiama duomenų įvedimo iš įvedimo srauto. Įvedimo srautu ateina išorinio failo pavadinimas, kuriame yra vartotojo programa
2. Failo skaitymas ir skaidymas puslapiais (po 256 žodžius)
3. Apdorojimo rezultatas - blokų sąrašas yra laikomas proceso atmintyje, todėl prašome atminties, į kurią galėtume perkelti programos blokus
4. Kopijuojame programos blokus į atmintį
5. Proceso programa yra ciklinė, todėl ReadFromInterface vėl užsiblokuoja

## Procesas JCL

Procesą JCL kuria ir naikina procesas StartStop. Proceso JCP paskirtis - gautus puslapius iš proceso ReadFromInterface suskirstyti į programos puslapius ir atidavusi procesui MainProc, vėl blokuotis laukiant pranešimo iš ReadFromFlash.

Šio proceso diagrama:



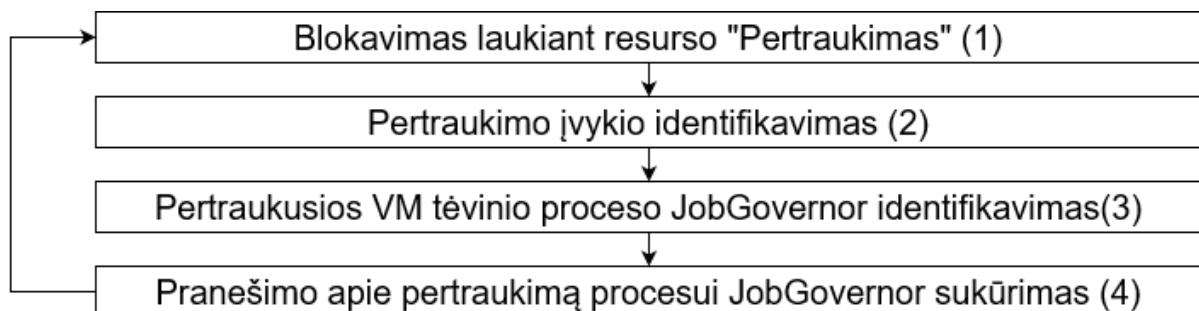
Proceso būsenos paaiškinimas:

1. Laukiame resurso "Uždutis atmintyje", kurį sukurs procesas ReadFromInterface
2. Validuojama programa
3. Jei programa korektiška, sukuriame resursą "užduoties vykdymo parametrai", kitu atveju procesas toliau laukia resurso "Užduotis atmintyje"

## Procesas Interrupt

Procesą kuria ir naikina procesas StartStop. Šio proceso paskirtis - reguoti į pertraukimus, kilusius virtualios mašinos darbo metu. Pradžioje procesas laukia resurso "Pertraukimas". Procesas nustato pertraukimo tipą apklausinėdamas programų nustatytas sisteminių kintamųjų reikšmes. Galiausiai kuriamas resursas "Interrupt", skirtas procesui JobGovernor.

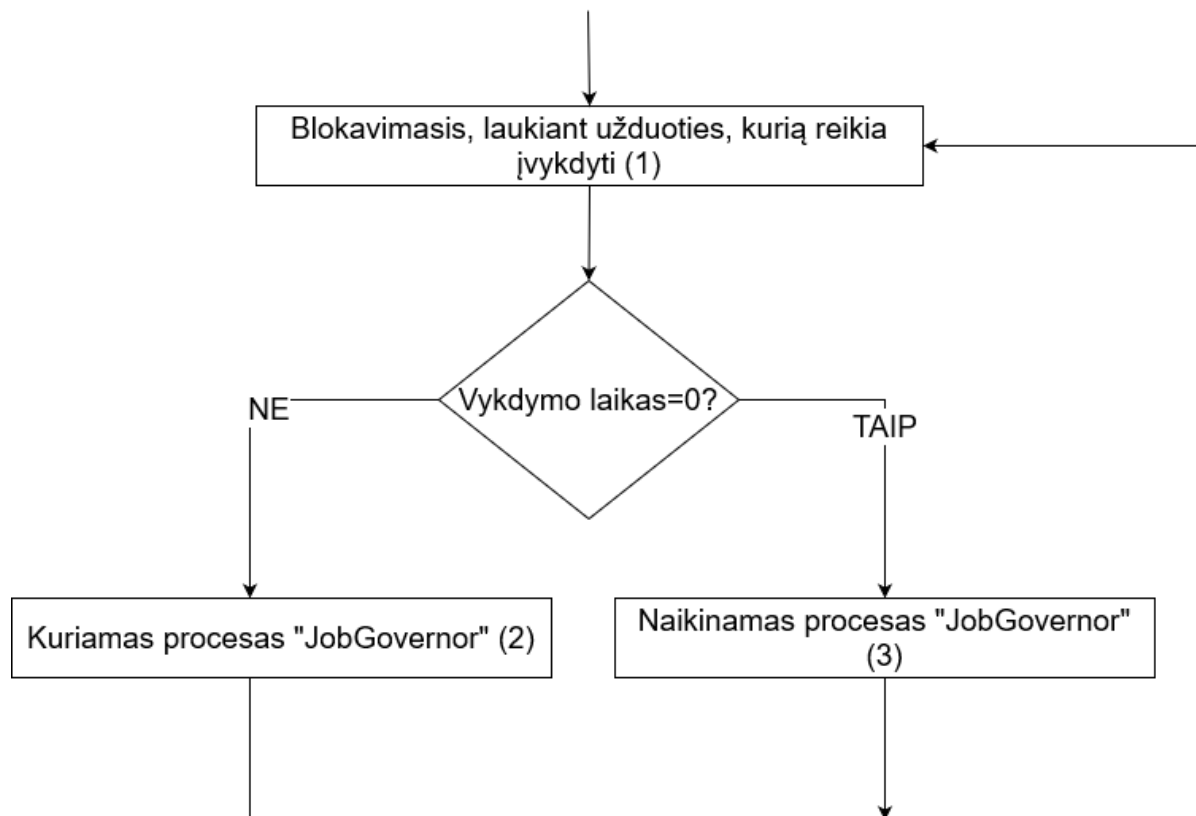
Šio proceso diagrama:



## Procesas MainProc

Procesą kurią ir naikina procesas StartStop. Šio proceso paskirtis - kurti ir naikinti procesus JobGovernor.

Šio proceso diagrama:



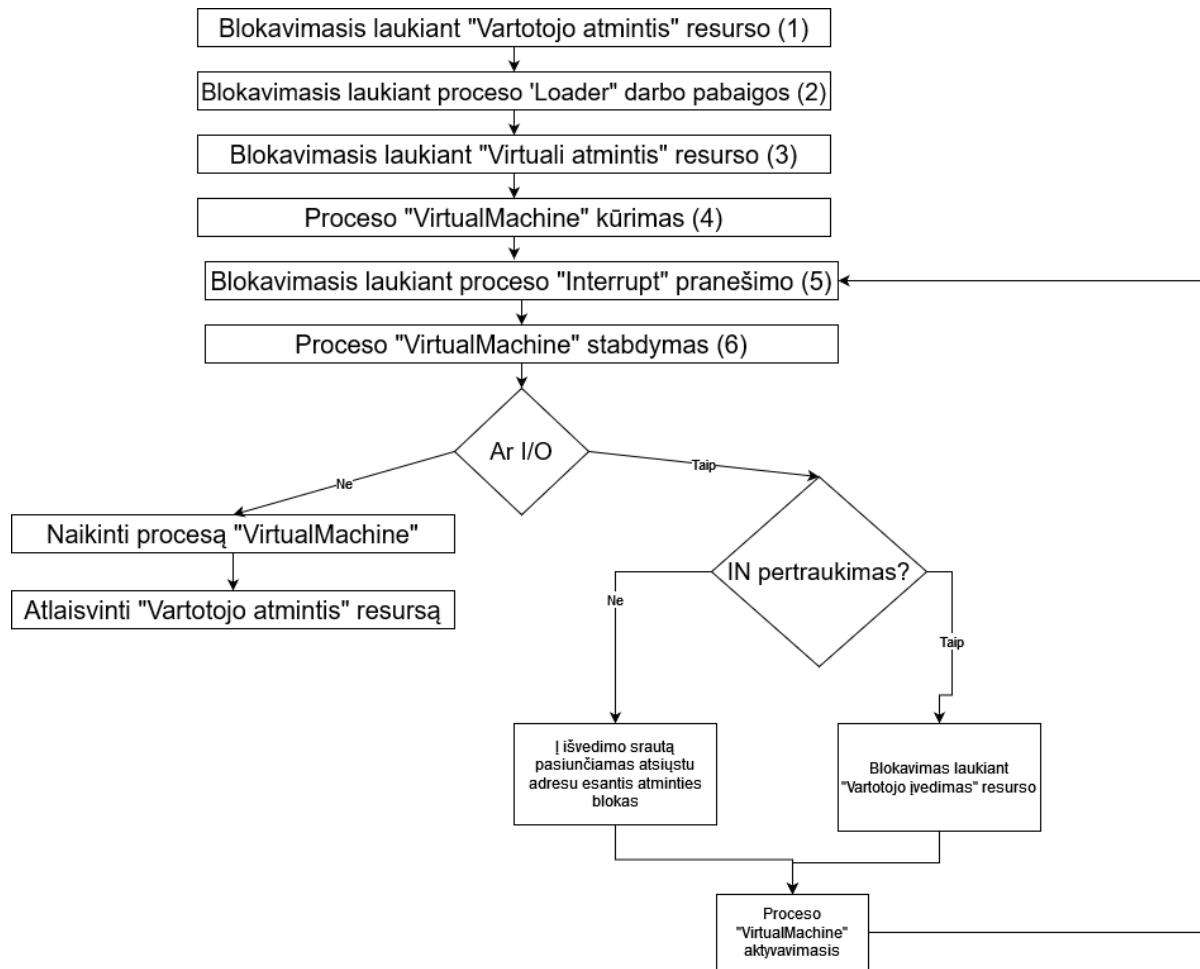
Proceso būsenos paaiškinimas:

1. Procesas prašo užduoties, kurią reikia įvykdyti
2. Tikrinama ar resurso vykdymo laikas nelygus 0 (t.y. ar užduotis nevykdoma per ilgai)
3. Jei laikas nelygus 0, kuriamas procesas JobGovernor, priešingu atveju - naikinamas procesas JobGovernor, kuris atsiuntė pradinį resursą su nuliniu vykdymo laiku
4. Atlikęs savo darbą procesas MainProc vėl blokuojasi laukdamas resurso

## Procesas JobGovernor

Procesus (jų gali būti keli) kuria procesas MainProc. Šių procesų paskirtis - padėti procesui VirtualMachine atlikti savo darbą (atlikti veiksmus, kurių virtuali mašina procesoriui dirbant vartotojo režimu nesugeba atlikti).

Šio proceso diagrama:



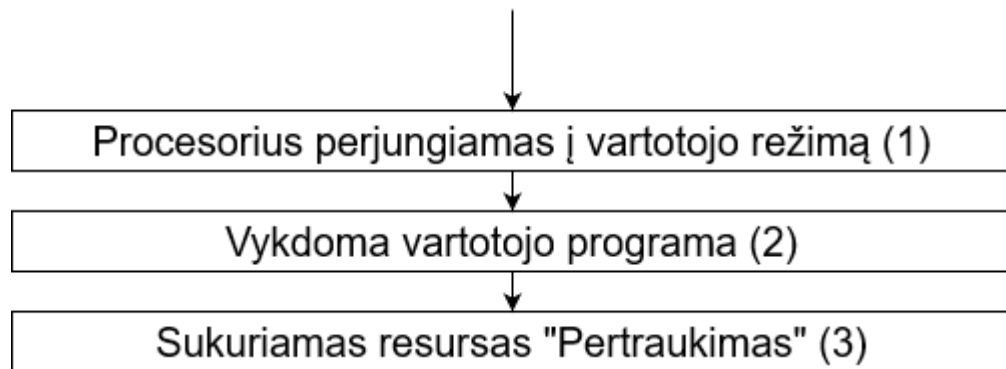
Proceso būsenos paaiškinimas:

1. Procesas prašo atminties, kurioje galėtų patalpinti užduoties programą
2. Perkėlimas užduoties programos į atmintį
3. Procesas prašo išskirti reikalingą atminties kiekį virtualiai mašinai
4. Kuriama nauja virtuali mašina (t.y. procesas VirtualMachine)
5. Procesas blokuojasi laukdamas pertraukimo proceso VirtualMachine vykdymo metu
6. Gavus pranešimą apie pertraukimą, nustatomas, koks tai pertraukimas
7. Jei tai nėra įvedimo/išvedimo pertraukimas - naikiname virtualios mašinos procesą ir atlaisviname vartotojo atmintį
8. Jei tai yra įvedimo pertraukimas, procesas blokuojasi, kol atsiras įvesties duomenų
9. Jei tai yra išvedimo pertraukimas, į išvedimo įrenginį naudojantis kanalų įrenginiu yra pasiunčiami išvedimui skirti duomenys
10. Apdorojus pertraukimą, procesas VirtualMachine yra aktyvuojamas ir JobGovernor procesas toliau pertraukimo pranešimo

# Procesas VirtualMachine

Procesą kuria ir naikina procesas JobGovernor. Proceso VirtualMachine paskirtis yra vykdyti vartotojo užduoties programą. Šių procesų yra tiek, kiek yra procesų JobGovernor.

Šio proceso diagrama:



Proceso būsenos paaiškinimas:

1. Persijungiama į vartotojo režimą
2. Vykdoma programa iki kol įvyksta pertraukimas
3. Įvykus pertraukimui virtuali mašina išsaugo savo procesoriaus būseną, valdymas yra perduodamas pertraukimą apdrosiančioms programoms. Kuriamas resursas "Pertraukimas" skirtas procesui Interrupt, kuris identifikuos pertraukimą ir perduos informaciją procesui JobGovernor