**Kurso „Operacinės sistemos“ trečiasis darbas**

**Multiprograminės operacinės sistemos projektas**

Darbą parengė:

Rokas Urlakis

Ernestas Zdanys

Marius Žilinskas

Pratybų dėstytojas:

Liutauras Ričkus

[1. Užduoties sąlyga 3](#_Toc417892884)

[2. Įžanga 4](#_Toc417892885)

[3. Procesai 4](#_Toc417892886)

[3.1 Procesų bendras aprašymas 4](#_Toc417892887)

[3.2 Procesų būsenos 4](#_Toc417892888)

[3.3 Planuotojas 6](#_Toc417892889)

[3.4 Procesų primityvai - Pagrindiniai veiksmai su procesais 6](#_Toc417892890)

[4. Resursai 7](#_Toc417892891)

[4.1 Bendras resursų aprašymas 7](#_Toc417892892)

[4.2 Resursų primityvai – Pagrindiniai veiksmai su resursais 7](#_Toc417892893)

[5. Procesų detalus aprašymas 9](#_Toc417892894)

[5.1 Procesų sąrašas 9](#_Toc417892895)

[5.2 Bendra procesų schema 11](#_Toc417892896)

[5.3 Procesas StartStop 12](#_Toc417892897)

[5.4 Procesas ReadFromFlash 13](#_Toc417892898)

[5.5 Procesas JCL 14](#_Toc417892899)

[5.6 Procesas MainProc 16](#_Toc417892900)

[5.7 Procesas JobGovernor 17](#_Toc417892901)

[5.8 Procesas Loader 19](#_Toc417892902)

[5.9 Procesas VirtualMachine 20](#_Toc417892903)

[5.10 Procesas Interrupt 21](#_Toc417892904)

[5.11 Procesas PrintLine 22](#_Toc417892905)

# Užduoties sąlyga

Užduotis E.

Projektuojama paketinė operacinė sistema:

* Virtualios mašinos procesoriaus komandos operuoja su duomenimis, esančiais steko viršūnėje.
  + Yra komandos duomenų persiuntimui iš atminties į steką ir atvirkščiai
  + Aritmetinės (sudėties, atimties, daugybos, dalybos)
  + Sąlyginio ir besąlyginio valdymo perdavimo
  + Įvedimo
  + Išvedimo
  + Šakojimo (sukuria proceso kopiją; pagal jos grąžinamą reikšmę procesas, nustato, ar jis yra tėvinis, ar vaikinis)
  + Programos pabaigos komandos
  + Registrai yra du:
    - Komandų skaitiklio
    - Steko viršūnės
  + Atminties dydis yra 16 blokų po 16 žodžių (žodžio ilgį pasirinkite patys)
* Realios mašinos procesorius gali dirbti dviem režimais:
  + Vartotojo
  + Supervizoriaus
  + Yra taimeris, kas tam tikrą laiko intervalą generuojantis pertraukimus
  + Įvedimui naudojamas virtualių „flash atmintinių“ nuskaitymo įrenginys
  + Išvedimui naudojamas spausdintuvas
  + Yra išorinės atminties įrenginys - kietasis diskas
* Virtualios mašinos atmintis atvaizduojama į vartotojo atmintį naudojant puslapių transliaciją
* Vartotojas užduočių paketą pateikia „prijungęs“ atmintinę.
  + Sistema perkelia visas joje esančias užduotis į diską, patikrindama jų sintaksę, ir, tuo pat metu, jei tik yra reikiamų resursų, pradeda jas vykdyti.

# Įžanga

Kompiuteryje gali veikti kelios programos vienu metu. Multiprograminėje operacinėje sistemoje tai pasiekiama procesų perjungimų sistema. Kiekvienas procesas yra vykdomas labai trumpą laiko tarpą ir taip vienoje sekundėje gali būti aptarnauti keli ar keliolika procesų. Tokiu būdu vartotojui atrodo, kad jie tarsi veikia vienu metu. O toliau, viskas šiek tiek detaliau.

# Procesai

## Procesų bendras aprašymas

Multiprograminės operacinės sistemos pasižymi sugebėjimu naudojantis vienu centriniu procesoriumi atlikti kelias užduotis vienu metu. Realybėje vienu metu procesoriumi gali naudotis tik vienas procesas, bet dažniausiai vienas procesas netrunka ilgiau dešimtosios sekundės ir pereinama prie kito. Toks greitas procesų aptarnavimas suteikia lygiagretaus veikimo įspūdi.

Procesas – tai vykdoma programa, kartu su esamomis registrų reikšmėmis ir savo kintamaisiais. Kiekvienam procesui sukuriamas 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 apibūdina deskriptorius. Jame laikomi visi procesui reikalingi parametrai – virtualaus procesoriaus registrų reikšmės, jam reikalingi kintamieji.

Sistemos veikimo eigoje turi būti vienas pagrindinis procesas, iš kurio seks likusieji.

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

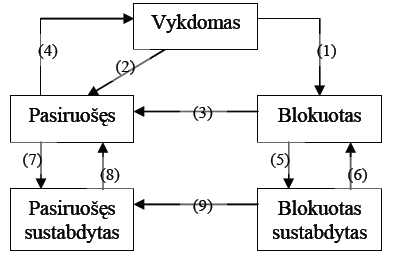
## Procesų būsenos

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 (pavyzdžiui, prašydamas įvedimo iš klaviatūros). Procesas blokuojasi priverstinai (nes jis vis tiek negali tęsti savo darbo be reikiamo resurso). Tačiau, jei procesas nereikalauja jokio resurso, iš jo gali būti atimamas procesorius, pavyzdžiui, vien tik dėl to, kad pernelyg ilgai dirbo. Galime išskirti procesų būsenas:

1. • Vykdomas – turi procesorių
2. • Blokuotas – prašo resurso (išskyrus procesorių)
3. • Pasiruošęs – vienintelis trūkstamas resursas yra procesorius. pvz. laukiama išorinio įrenginio įsiterpimo

Tačiau šių būsenų gali neužtekti. Gali susiklostyti tokia situacija, kad tam tikram procesui negalima leisti gauti procesoriaus, nors jis ir yra pasiruošęs. Toks procesas vadinamas sustabdytu. Galime papildyti būsenų sąrašą:

1. • Sustabdytas – kito proceso sustabdytas procesas.



1. Vykdomas procesas blokuojasi jam prašant 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. Pasiruošę procesai varžosi dėl procesoriaus. Gavęs procesorių procesas tampa vykdomu.
5. Procesas gali tapti sustabdytu blokuotu, jei einamasis procesas jį sustabdo, kai jis jau ir taip yra blokuotas.
6. Procesas tampa blokuotu iš blokuoto sustabdyto, jei einamasis procesas nuima būseną sustabdytas.
7. Procesas gali tapti pasiruošusiu sustabdytu, jei einamasis procesas jį sustabdo, kai jis yra pasiruošęs.
8. Procesas tampa pasiruošusiu iš pasiruošusio sustabdyto, jei einamasis procesas nuima būseną sustabdytas.
9. Procesas tampa pasiruošusiu sustabdytu iš blokuoto sustabdyto, jei procesui yra suteikiamas jam 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 ir atlieka atima procesorių iš vieno proceso ir perduoda jį kitam.

Planuotojo tikslai:

* Pasirūpinti, kad kiekvienas procesas tinkamą laiko tarpą tūrėtų procesorių (jei procesas trunka ilgai, procesorius gali būti iš jo atimtas);
* Laikyti procesoriaus apkrova netoli 100%;
* Pasirūpinti, kad atsakymo laikas vartotojams būtų kuo trumpesnis;
* Siekti kuo didesnio efektyvumo.

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

*Planuotojo veikimo schema:*

Einamasis procesas blokuotas?

Einamasis procesas įtraukiamas į blokuotų procesų sąrašą

Yra pasiruošusių procesų?

Aukščiausio prioriteto procesui suteikiamas valdymas

Vykdomas procesas, kuriam ką tik suteikėme valdymą

Laukiama, kol atsiras pasiruošęs procesas

Taip

Taip

Ne

Ne

## Procesų primityvai - Pagrindiniai veiksmai su procesais

Procesų primityvų paskirtis – pateikti vienodą ir paprastą vartotojo sąsają darbui su procesais. Kiekvieno primityvo programos gale kviečiamas planuotojas. Yra 4 primityvai:

* **Kurti procesą**. Šiam primityvui perduodama nuoroda į jo tėvą, jo pradinė būsena, prioritetas, perduodamų elementų sąrašas ir išorinis vardas. Pačio primityvo viduje vyksta proceso kuriamasis darbas. Jis yra registruojamas bendrame procesų sąraše, tėvo-sūnų sąraše, skaičiuojamas vidinis identifikacijos numeris, sukuriamas jo vaikų procesų sąrašas (tuščias), sukurtų resursų sąrašas.
* **Naikinti procesą**. Pradedama naikinti proceso sukurtus resursus ir vaikus. Vėliau išmetamas iš tėvo sukurtų procesų sąrašo. Toliau išmetamas iš bendro procesų sąrašo ir, jei reikia, iš pasiruošusių procesų sąrašo. Galiausiai naikinami visi jam perduoti resursai ir proceso deskriptorius yra sunaikinamas.
* **Stabdyti procesą**. Keičiama proceso būsena iš blokuotos į blokuotą sustabdytą arba iš pasiruošusios į pasiruošusią sustabdytą. Einamasis procesas stabdomas tampa pasiruošusiu sustabdytu.
* **Aktyvuoti procesą**. Keičiama proceso būsena iš blokuotos sustabdytos į blokuotą, ar pasiruošusios sustabdytos į pasiruošusią.
* **Keisti prioritetą**. Procesas pašalinamas iš sąrašo ir įterpiamas į sąrašą su nauju prioritetu.

# Resursai

## Bendras resursų aprašymas

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

1. • **Statinius resursus**. Kuriami sistemos kūrimo metu. Tai mašinos resursai, tokie kaip procesorius, atmintis ar kiti resursai, kurie sistemos veikimo metu nėra naikinami.
2. • **Dinaminius resursus**. 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 primityvai – Pagrindiniai veiksmai su resursais

1. **Kurti resursą.** Resursus kuria tik procesas. Resurso kūrimo metu perduodami kaip parametrai: nuoroda į proceso kūrėją, resurso išorinis vardas. Resursas kūrimo metu yra: pridedamas prie bendro resursų sąrašo, pridedamas prie tėvo sukurtų resursų sąrašo, jam priskiriamas unikalus vidinis vardas, sukuriamas resurso elementų sąrašas ir sukuriamas laukiančių procesų sąrašas.
2. **Naikinti resursą.** Resurso deskriptorius išmetamas iš jo tėvo sukurtų resursų sąrašo, naikinamas jo elementų sąrašas, atblokuojami procesai, laukiantys šio resurso, išmetamas iš bendro resursų sąrašo, ir galiausiai naikinamas pats deskriptorius.
3. **Prašyti resurso.** Šį primityvą kartu su primityvu “atlaisvinti resursą” procesai naudoja labai dažnai. Procesas, iškvietęs šį primityvą, yra užblokuojamas ir įtraukiamas į to resurso laukiančių procesų sąrašą. Sekantis šio primityvo žingsnis yra kviesti resurso paskirstytoją.
4. **Atlaisvinti resursą.** Šį primityvą kviečia procesas, kuris nori atlaisvinti jam nereikalingą resursą arba tiesiog perduoti pranešimą ar informaciją kitam procesui. Resurso elementas, primityvui perduotas kaip funkcijos parametras, yra pridedamas prie resurso elementų sąrašo. Šio primityvo pabaigoje yra kviečiamas resursų paskirstytojas.

# Procesų detalus aprašymas

## Procesų sąrašas

|  |  |  |  |
| --- | --- | --- | --- |
| Pavadinimas | Kas sukuria/ sunaikiną procesą | Paskirtis | Su kokiais resursais dirba |
| StartStop | Sukuriamas ir naikinamas  automatiškai kartu su sistema | Sukurti sisteminius procesus bei resursus, pasirūpinti korektiška sistemos darbo pradžia ir pabaiga | Sisteminiai statiniai |
| ReadFromFlash | StartStop | užduoties nuskaitymui iš įvedimo srauto („flash atmintinės“) | “Flash atmintinė”  Supervizorinė atmintis  “Užduotis supervizorinėje atmintyje” |
| JCL | StartStop | suskaidyti užduoties programą blokais ir organizuoti juos kaip resursus; | Sisteminiai statiniai  “Užduotis supervizorinėje atmintyje”  “Eilutė atmintyje”  “Užduoties programa supervizorinėje atmintyje”  “Užduoties duomenys supervizorinėje atmintyje” |
| Loader | StartStop | Perkelti duomenis iš supervizorinės į vartotojo atmintį; | Sisteminiai statiniai  “Kanalų įrenginio”  “Pakrauk programą”  “Pakrauta” |
| MainProc | StartStop | valdyti JobGovernor procesus | Sisteminiai statiniai  “Programa parengta” |
| JobGovernor | MainProc | virtualios mašinos proceso tėvas. valdyti virtualios mašinos proceso darbą | Sisteminiai statiniai  “Iš input/output”  “Pakrauta”  “Pakrauk programą”  “Iš interrupt”  „Vartotojo atmintis“  “Programa parengta”  „Bendra atmintis“ |
| VirtualMachine | JobGovernor | atsakyti už vartotojiškos programos vykdymą | Sisteminiai statiniai  “Pertraukimas” |
| Interrupt | StartStop | apdoroti virtualios mašinos pertraukimą sukėlusią situaciją | Sisteminiai statiniai  “Pertraukimas”  “Iš interrupt” |
| PrintLine | StartStop | Persiųsti į išvedimo įrenginį eilutę (spausdintuvą) iš supervizorinės atminties | Supervizorinė atmintis  Kanalų įrenginys |

## Bendra procesų schema

ReadFromFlash

JCL

MainProc

JobGovernor

VirtualMachine

JobGovernor

VirtualMachine

Interrupt

.............

## Procesas StartStop

Šis procesas atsakingas už sistemos darbo pradžią ir pabaigą. Įjungus kompiuterį (mūsų atveju - paleidus POS modelį), šis procesas pasileidžia automatiškai. Šio proceso paskirtis, sisteminių procesų bei resursų kūrimas.

**START**

**STOP**

Sisteminių statinių procesų inicializacija (2)

Blokavimasis laukiant „POS pabaiga“ resurso (3)

Sisteminių procesų naikinimas (4)

Sisteminių resursų naikinimas (5)

Sisteminių resursų inicializacija (1)

Trumpas paaiškinimas:

1. Procesas, gavęs procesorių, savo darbą pradeda sukurdamas visus sisteminius resursus (1).
2. Sukūręs resursus, procesas kuria statinius procesus (2), t. y. tokius procesus, kurie bus aktyvūs visą POS gyvavimo laiką (tai procesai: ReadFromFlash, JCL, Loader, MainProc, Interrupt, PrintLine).
3. Prašo resurso „POS pabaiga“ (3), t. y. šioje vietoje procesas blokuojasi ir laukia, kol bus atlaisvintas pranešimas apie POS darbo pabaigą. Priklausomai nuo prioriteto šis procesas bus atblokuotas ir tęs darbą.
4. Visų sisteminių procesų naikinimas (4).
5. Sisteminių resursų naikinimas (5). POS darbo pabaiga.

## Procesas ReadFromFlash

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.

Sukuriamas resursas „Programos duomenys supervizorinėje atmintyje“ (5)

Blokavimasis laukiant resurso „Flash atmintinė“ (1)

Programos failų nuskaitymas ir suskaidymas puslapiais (2)

Blokavimasis laukiant resurso „Supervizorinė atmintis“ (3)

Puslapių kopijavimas į supervizorinę atmintį (4)

Sukuriamas resursas „Užduotis supervizorinėje atmintyje“ skirtas procesui JCL (6)

Trumpas paaiškinimas:

1. Pirmasis žingsnis – laukti įvedimo srauto (1). Įvedimo srautu ateina išorinio failo pavadinimas t. y. kelių programų paketas (tame faile saugomi kitų failų, kuriuose yra programos, pavadinimai). Šis failas realizuoja „užduočių paketą“. Įvedant failo pavadinimą, reikalinga failų sistema.
2. Failų skaitymas ir skaidymas puslapiais (po 16 baitų) (2).
3. Apdorojimo rezultatas – puslapių sąrašas yra laikomas supervizorinėje atmintyje, todėl mes jos prašome (3) ir kopijuojame puslapius į atmintį (5).
4. Yra sukuriamas ir atlaisvinamas resursas, skirtas procesui JCL, kuriame yra informacija apie puslapių padėtį atmintyje (6).
5. Proceso programa yra ciklinė, todėl ReadFromFlash vėl užsiblokuoja laukdamas.

## Procesas JCL

Procesą JCL kuria ir naikina procesas StartStop. Proceso JCL paskirtis – gautus puslapius iš proceso ReadFromFlash suskirstyti į programos blokus, ir atidavus procesui MainProc, vėl blokuotis laukiant pranešimo iš ReadFromFlash.

Imamas sekantis blokas iš supervizorinės atminties (6)

Prijungiamas blokas prie programos sąrašo (5)

TAIP

NE

Blokavimasis laukiant resurso „Užduotis supervizorinėje atmintyje“ (1)

Programos blokų sąrašo inicijavimas (2)

Imamas pirmas blokas iš supervizorinės atminties (3)

Ar tai „DATA“ blokas?(4)

Atlaisvinamas resursas „Eilutė atmintyje“, su parametrais: „Nėra vartotojo programos” (8)

Sukuriamas „Užduoties vykdymo parametrai supervizorinėje atmintyje“ (9)

Ar tai „CODE“ blokas?(7)

NE

TAIP

Blokavimasis laukiant resurso „Supervizorinė atmintis“ (0)

Trumpas paaiškinimas:

1. Procesas pradeda savo darbą blokuodamasis, laukdamas pranešimo iš ReadFromFlash (1).
2. Sulaukus šio pranešimo, procesas pasiruošia darbui ir inicijuoja programos blokų sąrašą (2).
3. Imamas pirmas blokas (3).
4. Žiūrima, ar pirmi 4 žodžiai yra „DATA”(4).
5. Jei taip, prijungiamas puslapis prie programos sąrašo (išskyrus pirmus 4 baitus, kuriuose talpinamas „DATA“) (5).
6. Jei ne, tai žiurime ar sekantis puslapis yra „CODE“(7).
7. Toliau yra imamas sekantis puslapis ir toliau elgiamės kaip (7) žingsnyje.
8. Jei nėra „CODE“, tai elgiamės kaip (8) žingsnyje.
9. Jei yra „CODE“, atlaisvinamas resursas „Užduoties vykdymo parametrai supervizorinėje atmintyje“ (tai sisteminis resursas), skirtas MainProc procesui su nuoroda į programos blokų sąrašą (9). Toliau JCL blokuojasi laukdamas pranešimo iš ReadFromFlash (1).

Taigi, apibendrinimui galima pasakyti, jei programa parašyta nekorektiškai, tai ji nebus vykdoma, nes ji paprasčiausiai nebus saugoma jokioje atmintyje.

## Procesas MainProc

Šio proceso paskirtis – kurti ir naikinti procesus JobGovernor.

TAIP

NE

Kuriamas procesas „JobGovernor“(2)

Blokavimasis laukiant „Užduoties vykdymo parametrai supervizorinėje atmintyje“ resurso (1)

Naikinamas procesas „JobGovernor“ (3)

Vykdymo laikas=0?

Trumpas paaiškinimas:

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

## 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).

Resurso „Pakrauk programą” sukūrimas (0)

Blokavimasis laukiant „Vartotojo atmintis” resurso (2)

B

Kuriama puslapių lentelė (3)

Proceso „Virtual Machine“ kūrimas (4)

Blokavimasis laukiant proceso „Interrupt“ pranešimo (5)

Ar Output? ?,.hhewqeq????pertraukymas?

Proceso „Virtual Machine“ stabdymas (6)

NE

TAIP

Naikinti procesą „Virtual Machine“

Atlaisvinti „Vartotojo atmintis“ resursą

Kuriamas resursas „Eilutė atmintyje“ su pranešimu

Proceso „VitualMachine“ aktyvavimasis

Ar FORK?

? I/pertraukymas?

Perkelti visus tėvo turimus duomenis į supervizorinę atmintį.

NE

TAIP

Blokavimasis laukiant proceso „Loader“ darbo pabaigos (1)

Sukuriamas „Užduoties vykdymo parametrai supervizorinėje atmintyje“ resursas

PC ir SP nukopijuojami iš tėvo.

Blokavimasis laukiant resurso „Supervizorinė atmintis“

Trumpas paaiškinimas:

1. Procesas prašo vartotojo atminties, kad galėtų patalpinti vartotojo užduoties programą (1,2).
2. Perkėlus vartotojo užduoties programą į vartotojo atmintį programa pašalinama iš supervizorinės atminties, kur ji buvo iki šiol (1).
3. Procesas prašo vieno takelio virtualios mašinos puslapių lentelei. Lentelė užpildoma išskirtos vartotojo atminties adresais (3).
4. Kuriama virtuali mašina t. y. procesas VirtualMachine (4).
5. Procesas JobGovernor blokuojasi ir laukia pertraukimo proceso VirtualMachine vykdymo metu (5).
6. Gavus pranešimą apie pertraukimą (pertraukimai gali kilti, kai TI=0, PI<>0 arba SI<>0) stabdomas procesas VirtualMachine, tikrinama ar tai įvedimo/išvedimo pertraukimas (6).
7. Jei tai nėra įvedimo/išvedimo pertraukimas – tikriname ar tai yra FORK.
8. Jei tai ne FORK - naikinamas procesas VirtualMachine, atlaisvinama virtualios mašinos užimta vartotojo atmintis.
9. Jei tai FORK – kuriame virtualios mašinos (proceso VirtualMachine) kopiją.
10. Jei tai PRNT (išvedimas) - pertraukimas į išvedimo srautą naudojantis kanalų įrenginiu pasiunčiami išvedimui skirti blokai, esantys vartotojo atmintyje.
11. Vėliau aktyvuojamas procesas „VirtualMachine“ ir JobGovernor cikliškai grįžta blokuotis bei laukti „Interrupt“ pranešimo.

## Procesas Loader

Procesą kuria ir naikina procesas StartStop. Šio proceso paskirtis – supervizorinėje atmintyje esančius blokus perkelti į vartotojo atmintį.

Blokavimasis laukiant resurso „Pakrauk programą“ (1)

Blokavimasis laukiant resurso „Vartotojo atmintis“ (2.1)

Įvykdoma komanda XCHG (3)

Atslaisvinamas resursas „Programos duomenys supervizorinėje atmintyje“

Siunčiamas pranešimas apie darbo pabaigą (5)

Blokavimasis laukiant resurso „Programos duomenys supervizorinėje atmintyje“ (2.2)

Trumpas paaiškinimas:

1. Procesas pradeda darbą laukdamas resurso iš JobGovernor. Resurse laikoma informacija apie išorinės atminties takelius ir vartotojo atminties takelius – ką reikės perkelti, kur reikės perkelti (1).
2. Gavęs šį resursą procesas blokuojasi laukdamas resurso „Vartotojo atmintis“ (2.1).
3. Nustačius atitinkamus kanalo įrenginio registrus vykdoma komanda XCHG kiekvienam takeliui (3).

Atlaisvinamas resursas „Vartotojo atmintis“, „Supervizorinė atmintis“ ir sukuriamas bei atlaisvinamas pranešimas apie darbo pabaigą, skirtas būtent tam JobGovernor, kuris atsiuntė resursą. (4.1, 4.2, 5).

## Procesas VirtualMachine

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

Procesoriaus perjungimas į vartotojo režimą (1)

Vykdoma vartotojo programa (2)

Sukuriamas ir atlaisvinamas resursas „Pertraukimas“(3)

Trumpas paaiškinimas:

1. Persijungiama į vartotojo režimą (1).
2. Interpretuojama programa kol neįvyksta pertraukimas (2).
3. Įvykus pertraukimui virtuali mašina išsaugo savo procesoriaus būseną, valdymas perduodamas pertraukimą apdorosiančioms programoms. Kuriamas resursas „Pertraukimas“ skirtas procesui Interrupt, kuris identifikuos pertraukimą ir perduos informaciją procesui JobGovernor (3).

## Procesas Interrupt

Šio proceso paskirtis – reaguoti į pertraukimus, kilusius virtualios mašinos darbo metu. Pradžioje laukiama resurso „Pertraukimas“. Procesas nustato pertraukimo tipą apklausinėdamas programų nustatytas sisteminių kintamųjų reikšmes. Galiausiai kuriamas resursas „Interrupt“, skirtas procesui JobGovernor.

Blokavimasis laukiant resurso „Pertraukimo įvykis“

Pertraukimo įvykio identifikavimas

Pertrauktosios VM tėvinio proceso JobGovernor identifikavimas

Sukuriamas ir atlaisvinamas resursas „Interrupt“

## Procesas PrintLine

Procesą kuria ir naikina StartStop. Šio proceso paskirtis – į išvedimo srautą pasiųsti kokioje nors atmintyje esantį pranešimą. Procesas gavęs resursą „Eilutė atmintyje“ užsiblokuoja kol atsilaisvins kanalų įrenginys, jam atlaisvinus įvykdoma komanda XCHG, po kurios atlaisvinamas resursas „Kanalų įrenginys“. Rezultate, eilutė, esanti kokioje nors atmintyje, bus išvesta į konsolę vartotojui.

Blokavimasis laukiant resurso „Eilutė atmintyje“

Blokavimasis laukiant „2 kanalo“ resurso

Eilutės išvedimas

„2 kanalo“ resurso atlaisvinimas