**VILNIAUS UNIVERSITETAS**

**MATEMATIKOS IR INFORMATIKOS FAKULTETAS**

**PROGRAMŲ SISTEMŲ KATEDRA**

**Operacinės sistemos projektas**

Darbą atliko 2 kurso 3 grupės studentai:

Haroldas Baltrūnas

Deividas Frolovas

2014 m. Vilnius

**Turinys**

[1. Realios ir virtualios mašinos modeliai](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.8q7jiu472ozy) 1 psl.

[2. Realios mašinos procesorius](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.bvuh0kgngcxk) 5 psl.

[3. Virtualios mašinos procesorius](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.sbpfhf8e3133) 6 psl.

[4. Virtualios mašinos procesoriaus komandos](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.532jpqoms6xc) 7 psl.

[5. Realios mašinos atmintys](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.ht54drpq6des) 13 psl.

[6. Virtualios mašinos atmintis](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.huw3f7rg2rfd) 14 psl.

[7. Puslapiavimo mechanizmas](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.c2qll6g9lrw5) 14 psl.

[8. Taimerio mechanizmas](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.z6wcjiayqmd1) 15 psl.

[9. Pertraukimų mechanizmas](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.sd985ucou3zg) 15 psl.

[10. Kanalų mechanizmas](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.jam6wavufid0) 16 psl.

[11. Lemputė](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.jud41bfh7f78) 16 psl.

[12. Įvedimo įrenginys](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.r3mpl687drd6) 17 psl.

[13. Išvedimo įrenginys](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.u05e2n5qf6m8) 17 psl.

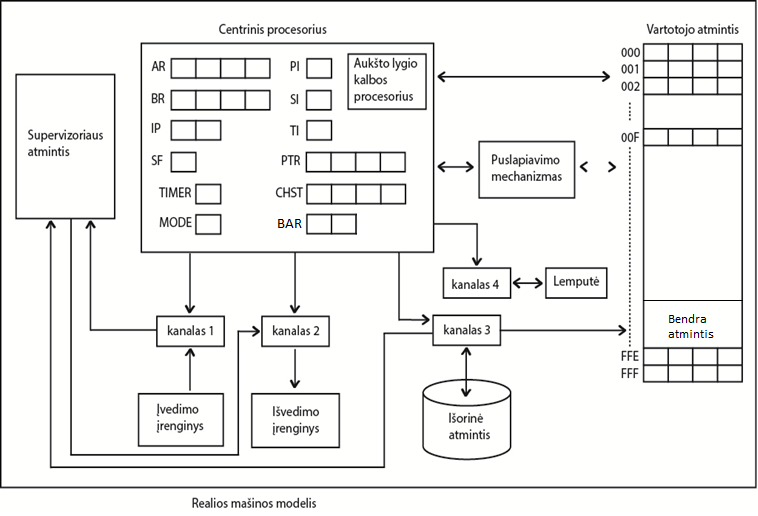
[14. Programos formatas](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.j256t4lz99e) 18 psl.

[15. Virtuali mašina operacinės sistemos kontekste](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.pd6qyjdad09r) 19 psl.

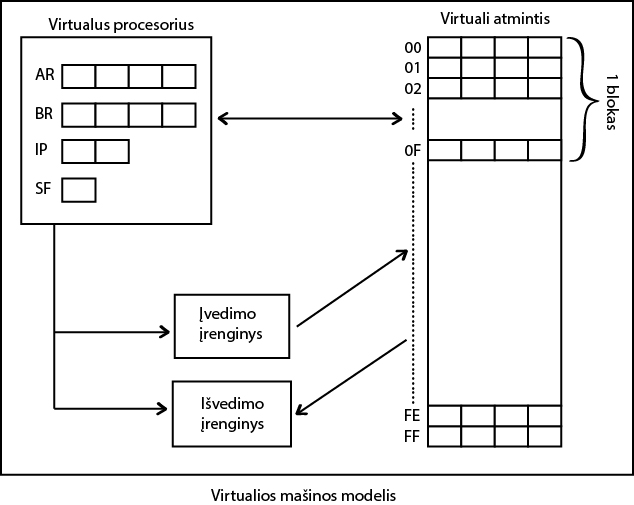
[16. Virtualios mašinos loginių komponentų sąryšis su realios mašinos techninės įrangos komponentais](https://docs.google.com/document/d/1JqXzC3nOuAUJisfRzEFeAUZIEeSF8pY-0FtdazmxO0A/edit#heading=h.5bf59qp4v2e8) 19 psl.

1. **Realios ir virtualios mašinos modeliai**

Kurdami multiprograminės operacinės sistemos modelį, remsimės šiuo realios mašinos modeliu:



Bei šiuo virtualios mašinos modeliu:



**2. Realios mašinos procesorius**

Realios mašinos procesoriaus paskirtis – skaityti komandą iš atminties ir ją vykdyti. Procesorius gali dirbti dviem režimais – supervizoriaus arba vartotojo. Supervizoriaus režime komandos, iš supervizorinės atminties, yra apdorojamos betarpiškai aukšto lygio kalbos procesoriaus. Vartotojo režime aukšto lygio kalbos procesorius vykdo užduoties programą (šiuo atveju aukšto lygio kalbos procesorius imituoja virtualios mašinos procesorių).

**Realios mašinos procesoriaus registrai:**

**AR –**4 baitų bendro naudojimo registras.

**BR –**4 baitų bendro naudojimo registras.

**PTR –** 4 baitų registras puslapių lentelei.

**BAR –** 2 baitų registras laikantis bendrosios atminties bloko adresą.

**IP –** 2 baitų virtualios mašinos atminties ląstelių skaitliukas.

**SF –**4 baitų statuso registras, į kurį įeina 4 vėlėvėlės:

**ZF -** nulio vėlėvėlė. (1 bitas),

**CF -** pernešimo vėlėvėlė. (1 bitas),

**SF -** ženklo vėlėvėlė. (1 bitas),

**BF -** baito vėlėvėlė. (1 bitas).

**MODE –** procesoriaus darbo režimo požymio registras („S“ – supervizorius, „U“ – vartotojas).

**TIMER -** taimeris.

**PI –** programinio pertraukimo (1 baitas).

**SI –** supervizoriaus režimo perjungimo (1 baitas).

**IOI –** įvedimo/išvedimo pertraukimo (1 baitas).

**TI –** taimerio pertraukimo (1 baitas).

**CHST[i], i=1, 2, 3, 4 –** kanalų būsenos registrai (1 baitas (reikšmė 1 arba 0)).

**3. Virtualios mašinos procesorius**

Virtualios mašinos procesoriaus paskirtis – vykdyti programą, kuri yra virtualioje atmintyje. Kiekvienas procesas turi savo „nuosavą“ virtualų centrinį procesorių, tačiau procesų programas vykdys aukšto lygio kalbos procesorius.

**Virtualios mašinos procesoriaus registrai:**

**AR –**4 baitų bendro naudojimo registras.

**BR –**4 baitų bendro naudojimo registras.

**IP –** 2 baitų virtualios mašinos atminties ląstelių skaitliukas.

**SF –**1 baito statuso registras, į kurį įeina 4 vėlėvėlės:

**ZF -** nulio vėlėvėlė. (1 bitas),

**CF -** pernešimo vėlėvėlė. (1 bitas),

**SF -** ženklo vėlėvėlė. (1 bitas),

**BF -** baito vėlėvėlė. (1 bitas).

**4. Virtualios mašinos procesoriaus komandos**

Atmintyje esantis 4 baitų žodis gali būti traktuojamas dvejopai: kaip žodis arba kaip komanda. Komandos yra dviejų tipų:

**1)** Pirmieji (vyresnieji) 2 baitai laikomi operacijos kodu, likę naudojami kaip operandai, paprastai tai bus virtualios atminties adresas.

**2)**   Visi 4 baitai laikomi operacijos kodu (pvz.: HALT, LDSB).

Mikroprogramos, interpretuojančios virtualaus procesoriaus komandas, bus vykdomos aukšto lygio kalbos procesoriaus. Pačios mikroprogramos turi būti pažymėtos tik skaitymui pažymėtoje supervizorinės atminties dalyje.

**Virtualios mašinos procesoriaus darbo su atmintimi ir registrais komandos:**

**LAxy** - atminties ląstelės, kurios adresas x\*10 + y turinio kopijavimas į registrą AR.

Trumpiau: AR:= [x\*10 + y];

**LBxy** - atminties ląstelės, kurios adresas x\*10 + y turinio kopijavimas į registrą BR.

Trumpiau: BR:= [x\*10 + y];

**WAxy** - registro AR turinio kopijavimas į atminties ląstelę, kurios adresas yra x\*10 + y.

Trumpiau: [x\*10 + y]: = AR;

**WBxy** - registro BR turinio kopijavimas į atminties ląstelę, kurios adresas yra x\*10 + y.

Trumpiau: [x\*10 + y]: = BR;

**STSB** - registro AR[AH , AL] AL 2 baitus kopijuoja į atminties vietą adresu AH, į poziciją, kurią nurodo B flagas.  
Jei B = 0, tai kopijuojama į aukštesniųjų 2 baitų vietą ir B = 1;

Jei B = 1, tai kopijuojama į žemesniųjų 2 baitų vietą ir B = 0, bei AH padidinamas vienetu;

Pvz. Atminties\_žodis[2\_aukštesni\_baitai, 2\_žemesni\_baitai];

**LDSB** - iš atminties vietos, kurios adresas yra AR[AH, AL] registro aukštesnysis baitas AH, kopijuojam 2 baitus į AR registro 2 žemesniųjų baitų vietą AL. Kuriuos atminties žodžio 2 baitus kopijuosim nurodo B flagas.  
Jei B = 0, tai kopijuojama iš aukštesniųjų 2 baitų vietos ir B = 1;

Jei B = 1, tai kopijuojama iš žemesniųjų 2 baitų vietos ir B = 0, bei AH padidinamas vienetu;

Pvz. Atminties\_žodis[2\_aukštesni\_baitai, 2\_žemesni\_baitai];

**LOxy** - Jei x=1, tuomet vygdomas pertraukimas, ir operacinė sistema rezervuoja y ląstelę, bei pažymi, kad ši ląstelė yra užimta;

Jei x=0, tuomet vygdomas pertraukimas, ir operacinė sistema atlaisvina y ląstelę, bei pažymi, kad ši ląstelė yra laisva, bet tik jeigu šią ląstelę užėmė ta pati mašina, kuri bando ją atlaisvinti.

**GTxx** - Adresu BAR\*10+x, kur BAR- bendrosios atminties bloko adresas, x- ląstelė iš kurios norima gauti duomenis, turinio kopijavimas į AR registrą. Vygdomas pertraukimas. Operacinė sistema sėkmingai įvygdo komandą tik tada, jei prieš tai virtuali mašina užsirezervuoja atititinkamą ląstelę.

**PTxx** – Registro AR turinio kopijavimas adresu BAR\*10+x, kur BAR- bendrosios atminties bloko adresas, x- ląstelė iš kurios norima gauti duomenis. Vygdomas pertraukimas. Operacinė sistema sėkmingai įvygdo komandą tik tada, jei prieš tai virtuali mašina užsirezervuoja atititinkamą ląstelę.

**Virtualios mašinos procesoriaus aritmetinės komandos:**

**AAxy** - prie registro AR reikšmės pridedama atminties ląstelės, kurios adresas x\*10 + y, reikšmė.

jei AR + [x\*10 + y] > 4bytes, tai C=1, S=0, Z=0 ir AR registrui priskiriam paskutinius 4bytes, o pirmas byte prarandamas.

jei AR + [x\*10 + y] <= 4bytes, tai C=0, S=0, Z=0 ir AR:=AR+[x\*10 + y].

jei AR + [x\*10 + y] = 10000,  tai C=1, S=0, Z=1 ir AR:=0.

**ABxy** - prie registro BR reikšmės pridedama atminties ląstelės, kurios adresas x\*10 + y, reikšmė.

jei BR + [x\*10 + y] > 4bytes, tai C=1, S=0, Z=0 ir AR registrui priskiriam paskutinius 4bytes, o pirmas byte prarandamas.

jei BR + [x\*10 + y] <= 4bytes, tai C=0, S=0, Z=0 ir AR:=AR+[x\*10 + y].

jei BR + [x\*10 + y] = 10000,  tai C=1, S=0, Z=1 ir BR:=0.

**SAxy** - iš registro AR reikšmės atimama atminties ląstelės, kurios adresas x\*10 + y, reikšmė.

jei AR - [x\*10 + y] < 0, tai C=1, S=1, Z=0 ir AR:= FFFF-[x\*10 + y]+AR+1.

jei AR - [x\*10 + y] > 0, tai C=0, S=0, Z=0 ir AR:= AR - [x\*10 + y].

jei AR - [x\*10 + y] = 0, tai C=0, S=0, Z=1 ir AR:= 0.

**SBxy** - iš registro BR reikšmės atimama atminties ląstelės, kurios adresas x\*10 + y, reikšmė.

jei BR - [x\*10 + y] < 0, tai C=1, S=1, Z=0 ir BR:= FFFF-[x\*10 + y]+BR+1.

jei BR - [x\*10 + y] > 0, tai C=0, S=0, Z=0 ir BR:= BR - [x\*10 + y].

jei BR - [x\*10 + y] = 0, tai C=0, S=0, Z=1 ir BR:= 0.

**CAxy** - jei AR - [x\*10 + y] = 0, tai C=0, S=0, Z=1.

jei AR - [x\*10 + y] > 0, tai C=0, S=0, Z=0.

jei AR - [x\*10 + y] < 0, tai C=1, S=1, Z=0.

**CBxy** - jei BR - [x\*10 + y] = 0, tai C=0, S=0, Z=1.

jei BR - [x\*10 + y] > 0, tai C=0, S=0, Z=0.

jei BR - [x\*10 + y] < 0, tai C=1, S=1, Z=0.

**Virtualios mašinos procesoriaus valdymo perdavimo komandos:**

**JMxy** - besąlyginis valdymo perdavimas adresu x\*10 + y.

IP = [x\*10 + y];

**JGxy** - sąlyginis valdymo perdavimas adresu x\*10 + y.

Valdymo perdavimas įvyksta, jei C = 0, S = 0, Z = 0.

IP = [x\*10 + y];

**JLxy** - sąlyginis valdymo perdavimas adresu x\*10 + y.

Valdymo perdavimas įvyksta, jei C = 1, S = 1, Z = 0.

IP = [x\*10 + y];

**JCxy** - sąlyginis valdymo perdavimas adresu x\*10 + y.

Valdymo perdavimas įvyksta, jei C = 1.

IP = [x\*10 + y];

**JZxy** - sąlyginis valdymo perdavimas adresu x\*10 + y.

Valdymo perdavimas įvyksta, jei Z = 1.

IP = [x\*10 + y];

**JNxy** - sąlyginis valdymo perdavimas adresu x\*10 + y.

Valdymo perdavimas įvyksta, jei   Z = 0.

IP = [x\*10 + y];

**LPxy** - sąlyginis valdymo perdavimas adresu x\*10 + y.

Valdymo perdavimas įvyksta, jei BR > 0.

IP = [x\*10 + y];

BR = BR - 1;

**HALT** - Vartotojo programos vykdymo pabaiga.

**Virtualios mašinos procesoriaus darbo su išoriniais įrenginiais komandos:**

**PDxy** - Išsiunčia išvedimui 16 žodžių srautą iš atminties ląstelių [x \*10 + i ], kur i = 0.. 15. Formalus užrašymas:

for (int i = 0; i < 15; i++) {

Buffer[i]:= [x\*10+i];

}

Čia Buffer – 16 žodžių masyvas, kur laikomas išvedimo rezultatas.

**GDxy** - iš įvedimo srauto perskaito 16 žodžių ir įrašo juos į lasteles [x \*10 + i] kur i = 0.. 15. Formalus užrašymas:

for (int i = 0; i < 15; i++) {

[x\*10+i]:= Buffer[i];

}

Čia Buffer – 16 žodžių masyvas, kur laikomas įvedimo rezultatas.

**LXON** - Vykdomas pertraukimas. Operacinė sistema pakeičia kanalų registro CHST[4] reikšmę į “1” ir įjungia lemputę.

**LXOF** - Vykdomas pertraukimas. Operacinė sistema pakeičia kanalų registro CHST[4] reikšmią į “0” ir išjungia lemputę.

**LXCH** -

jei CHST[4]:=1, tai z=1.

jei CHST[4]:=0, tai z=0.

**Virtualios mašinos procesoriaus loginės komandos:**

**XAxy** - Registro AR bei atminties ląstelės, kurios adresas x\*10 + y turinys konvertuojamas į dvejetainę (binarinę) sistemą. Tuomet lyginami atitinkami registro bei ląstelės bitai. Jei atitinkami bitai lygūs, tuomet į AR registro atitinkamą bito poziciją įrašomas 0, jei bitai skirtingi, į atitinkamą poziciją įrašomas1.

**XBxy** - Registro BR bei atminties ląstelės, kurios adresas x\*10 + y turinys konvertuojamas į dvejetainę (binarinę) sistemą. Tuomet lyginami atitinkami registro bei ląstelės bitai. Jei atitinkami bitai lygūs, tuomet į BR registro atitinkamą bito poziciją įrašomas 0, jei bitai skirtingi, į atitinkamą poziciją įrašomas 1.

**NAxy** -  Registro AR bei atminties ląstelės, kurios adresas x\*10 + y turinys konvertuojamas į dvejetainę (binarinę) sistemą. Tuomet lyginami atitinkami registro bei ląstelės bitai. Jei atitinkami bitai lygūs 1, tuomet į AR registro atitinkamą bito poziciją įrašomas 1, jei bitai skirtingi arba lygūs 0, tada į atitinkamą poziciją įrašomas 0.

**NBxy** -Registro BR bei atminties ląstelės, kurios adresas x\*10 + y turinys konvertuojamas į dvejetainę (binarinę) sistemą. Tuomet lyginami atitinkami registro bei ląstelės bitai. Jei atitinkami bitai lygūs 1, tuomet į BR registro atitinkamą bito poziciją įrašomas 1, jei bitai skirtingi arba lygūs 0, tada į atitinkamą poziciją įrašomas 0.

**OAxy** - Registro AR bei atminties ląstelės, kurios adresas x\*10 + y turinys konvertuojamas į dvejetainę (binarinę) sistemą. Tuomet lyginami atitinkami registro bei ląstelės bitai. Jei atitinkami bitai lygūs 1 arba yra skirtingi, tuomet į AR registro atitinkamą bito poziciją įrašomas 1, jei bitai lygūs 0, į atitinkamą poziciją įrašomas 0.

**OBxy** - Registro BR bei atminties ląstelės, kurios adresas x\*10 + y turinys konvertuojamas į dvejetainę (binarinę) sistemą. Tuomet lyginami atitinkami registro bei ląstelės bitai. Jei atitinkami bitai lygūs 1 arba yra skirtingi, tuomet į BR registro atitinkamą bito poziciją įrašomas 1, jei bitai lygūs 0, į atitinkamą poziciją įrašomas 0.

**5. Realios mašinos atmintys**

Mūsų reali mašina turi keturių rūšių atmintis:

***Supervizorinę***.Modelio atveju supervizorinė atmintis tampa tik sąvoka, kuri realiai nebus realizuojama. Supervizorinėje atmintyje turėjusiais būti komandas ir resursus valdys aukšto lygio kalbos procesorius;

***Vartotojo***. Vartotojo atmintis skirta virtualių mašinų atmintims bei puslapių lentlėms laikyti. Mes apibrėšime vartotojo atmitį taip: lentelės dydis – 4096 žodžių po 4 baitus. 16 žodžių laikysime vienu bloku. Taigi, vartotojo atmintis lygi 256 blokų, sunumeruotų nuo 00 iki FF.

***Išorinė***. Išorinė atmintis bus realizuojama failu kietajame diske. Mūsų atmintis turės 20 blokų po 16 žodžių, bei kiekvienas žodis bus sudarytas iš 4 baitų.

***Bendra***. Paskutinis Vartotojo atminties blokas, kurio adresą laiko BAR registras. Bendrosios atminties ląstelės gali būti rakinamos ir atrakinamos.

**6. Virtualios mašinos atmintis**

Kiekvienai virtualiai mašinai yra skiriama 16 vartotojo atminties blokų. Tuose 16 blokų turi tilpti užduoties programa. Kiekvienas virtualios atminties blokas turi virtualų ir realų adresą. Virtualiais adresais operuoja virtuali mašina, realiais – reali mašina. Ryšiai tarp virtualaus ir realaus adreso nusakomi puslapių lentelėmis.

**7. Puslapiavimo mechanizmas**

Virtualios mašinos kūrimo ir veiklos scenarijus:

* virtuali mašina reikalauja 16 blokų atminties savo reikmėms;
* išskiriama 16 atsitiktinių blokų virtualiai mašinai;
* puslapių lentelė (1 blokas) užpildomas išskirtų 16 blokų atsitiktiniais realiais adresais, bei pati puslapių lentelė yra atsitiktinai patalpinama realioje atmintyje;
* 4 baitų registro PTR reikšmei priskiriamas puslapių lentelės bloko realus adresas, kuris talpinamas i du paskutinius PTR baitus;
* į 4 baitų registro PTR a0 baitą įdedamas reikšmė “0”, į a1 įdedama reikšmė “F”.  a0, a1– PTR registro 1-asis ir antrasis baitai. 1-asis baitas nurodo vartotojo programos dydį puslapiais (0-F), ir yra įrašomas vartotojui įkėlus programą. 2-asis baitas nurodo maksimalų puslapių skaičių. Į 2-ajį baitą įdedama reikšmė “F”, nes programa gali užimti maksimaliai 16 blokų.
* virtuali mašina baigiama kurti;
* virtuali mašina gauna procesorių;
* virtuali mašina virtualų adresą paverčia realiu:
  + 1. Turime x y, kur x- virtualus bloko adresas, y- virtualus ląstelės adresas.
    2. Turime a2 a3 kur a2 – registro PTR 3-iasis baitas,  a3 – registro PTR 4-asis baitas.
    3. Iš realios atminties bloko, kurio adresas yra a2 a3 paimamas x-osios ląstelės turinys. Šis turinys yra realios atminties bloko adresas z.
    4. Realus bloko adresas z unikaliai atitinka virtualų bloko adresą x. y nurodo konkrečią bloko ląstelę.
* Paleidus programą, suskaičiuojama kiek yra komandų. Komandų skaičius dalinamas iš 16, ir suapvalinamas į didžiąją pusę. Tuomet šis skaičius įrašomas į PTR registro pirmąjį baitą.

Užimta arba negalima atmintis yra apsaugoma semaforų masyvu.

**8. Taimerio mechanizmas**

* Sukūrus virtualią mašiną, taimerio reikšmė nustatoma į 10.
* Vykdant kiekvieną komandą, taimerio reikšmė mažinama vienetu.
* Kai tamerio reikšmė pasiekia 0, vykdomas taimerio pertraukimas, bei reikšmė vėl atstatoma į 10.

**9. Pertraukimų mechanizmas**

Gali būti trijų tipų pertraukimai – programiniai, supervizorinai ir taimerio.

įvykus pertraukimui, virtualios mašinos registrų reikšmės yra išsaugomos, procesorius perjungiamas į supervizoriaus režimą, nustatomas pertraukimo pobūdis, kviečiama pertraukimą apdorojanti paprogramė. Vėliau valdymas grįžta į virtualią mašiną, procesorius perjungiamas į vartotojo režimą.

programinių pertraukimų registras yra **PI**:

* PI=1 – neteisingas adresas,
* PI=2 – neteisingas operacijos kodas,
* PI=3 – neteisingas priskyrimas,
* PI=4 – perpildymas;

supervizorinių pertraukimų registras yra **SI**:

* SI=1 – vykdoma komanda GD,
* SI=2 – vykdoma komanda PD,
* SI=3 – komanda HALT;
* SI=4 – vykdoma komanda LXON,
* SI=5 – vykdoma komanda LXOF,
* SI=6 – vykdoma komanda LO,
* SI=7 – vykdoma komanda GT,
* SI=8 – vykdoma komanda PT.

taimerio pertraukimų registras yra **TI**:

* TI=1 – vykdomas taimerio pertraukimas;

Pertraukimai gali būti aptikti tik vartotojo režime.

**10. Kanalų mechanizmas**

Kanalai užtikrina bendravimą tarp procesoriaus bei išorinių įrenginių. Mūsų sistemoje, kanalai bus 1 bito loginiai registrai. Jų bus 4:

1. Įvedimo įrenginio kanalo registras CHST[1].

2. Išvedimo įrenginio kanalo registras CHST[2].

3. Išorinės atminties kanalo registras CHST[3].

4. Lemputės kanalo registras CHST[4].

Jie nurodys, ar konkrečiu metu išorinis įrenginys yra naudojamas programos, taip užtikrinant, kad kita programa negalėtų naudoti šio įrenginio.

**11. Lemputė**

Lemputę reguliuoja kanalo registras CHST[4]. Jis užima 1 bitą ir gali įgauti dvi būsenas: 1 ir 0.

Jei CHST[4]=1, vadinasi lemputė yra įjungta, ir kitos programos negali į ją kreiptis.

Jei CHST[4]=0, vadinasi lemputė yra išjungta, ir kitos programos gali į ją kreiptis.

Egzistuoja virtualios mašinos komandos, galinčios lemputę įjungti bei išjungti, taip pat patikrinti jos būseną.

**12. Įvedimo įrenginys**

Įvedimo įrenginį reguliuoja kanalo registras CHST[1]. Jis užima 1 bitą ir gali įgauti dvi būsenas: 1 ir 0.

Kai vyksta įvedimo operacija, CHST[1] įgyja reikšmę 1, pasibaigus veiksmui CHST[1] įgyja reikšmę 0.

Egzistuoja virtualios mašinos komanda **GDxy** leidžianti įrašyti įvestį į atmintį.

Jei CHST[1] yra 1, vadinasi juo jau naudojasi tam tikra programa, ir kitos programos šiuo įrenginiu galės naudotis tik tada, kai juo baigs naudotis pirmoji programa ir CHST[1] įgys reikšmę 0.

**13. Išvedimo įrenginys**

Išvedimo įrenginį reguliuoja kanalo registras CHST[2]. Jis užima 1 bitą ir gali įgauti dvi būsenas: 1 ir 0.

Kai vyksta Išvedimo operacija, CHST[2] įgyja reikšmę 1, pasibaigus veiksmui CHST[2] įgyja reikšmę 0.

Egzistuoja virtualios mašinos komanda **PDxy** leidžianti išvesti duomenis iš atminties į ekraną (išvedimo įrenginį).

Jei CHST[2] yra 1, vadinasi juo jau naudojasi tam tikra programa, ir kitos programos šiuo įrenginiu galės naudotis tik tada, kai juo baigs naudotis pirmoji programa ir CHST[2] įgys reikšmę 0.

**14. Programos formatas**

Programos laikomos atskiruose failuose. Viena programa - vienas failas.

Programos dalis. Šiai daliai skirta 100 \* 4 = 400 baitų. Kad būtų paprasčiau vesti

programas reikėtų laikytis sistemos:

Prieš kiekvieną naują programos ar duomenų bloką, turi būti simbolių rinkinys $\*x\*,

kur x kinta intervale [0..F], ir x reikšmė reiškia bloko, į kurį bus rašomi duomenys, numerį. \* reiškia nenaudojamą baitą.

Tarpai, teksto išvedimui žymimi „-“.

Programa laikoma korektiška, jei užima nedaugiau kaip 200 eilučių.

Pavyzdžiui:

• $0000

• LA40

• AA41

• SA50

• PD50

• HALT

• $040

•0002

•0002

• $050

• 0000

Arba:

• LXON

• LXCH

• LXOF

• HALT

**15. Virtuali mašina operacinės sistemos kontekste**

Multiprograminėje operacinėje sistemoje centrinis procesorius yra perjungiamas iš vieno VM proceso į kitą VM procesą. Kiekvieną iš procesų vykdo tik kelias dešimtąsias ar šimtąsias sekundės dalis. Todėl net ir sekundės bėgyje procesorius sugeba aptarnauti keletą procesų, ir tai suteikia vartotojui lygiagretaus veikimo įspūdį.

**16. Virtualios mašinos loginių komponentų sąryšis su realios mašinos techninės įrangos komponentais**

Virtuali mašina susideda iš realios mašinos komponentų ir operuoja kaip ji. VM perima tik svarbiausius realios mašinos techninės įrangos komponentus, būtinus darbo atlikimui.