VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA



**VIRTUALIOS IR REALIOS MAŠINOS MODELIS**

Anastasija Kiseliova  
Evelina Bujytė  
MI, 3 kursas

2017 m.  
VILNIUS

TURINYS

[1. REALI MAŠINA 3](#_Toc475393366)

[1.1. Techninės įrangos komponentai sudarantys realią mašiną: 3](#_Toc475393367)

[1.2. Procesorius 3](#_Toc475393368)

[1.3. Realios mašinos procesorius turi tokius registrus: 4](#_Toc475393369)

[1.4. Atmintys 4](#_Toc475393370)

[1.5. Išvedimo ir įvedimo įrenginiai 4](#_Toc475393371)

[1.6. Puslapiavimo mechanizmas 4](#_Toc475393372)

[1.7. Duomenų perdavimo kanalai 4](#_Toc475393373)

[2. VIRTUALI MAŠINA 5](#_Toc475393374)

[2.1. Pagrindinė operacinės sistemos funkcija 5](#_Toc475393375)

[2.2. Modeliuojamos virtualios mašinos loginių komponentų aprašymas. 5](#_Toc475393376)

[2.2.1 Virtualios mašinos atmintis 5](#_Toc475393377)

[2.2.2 Virtualios mašinos procesorius. 6](#_Toc475393378)

[2.2.3 Virtualios mašinos komandų sistema. 6](#_Toc475393379)

[2.3. Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais mechanizmo aprašymas. 8](#_Toc475393380)

[2.4. Virtualios mašinos interpretuojamojo ar kompiliuojamo vykdomojo failo išeities teksto formatas. 8](#_Toc475393381)

[2.5. Modeliuojamos virtualios mašinos loginių komponentų sąryšio su realios mašinos techninės įrangos komponentais aprašymas. 9](#_Toc475393382)

# REALI MAŠINA

Centrinis Procesorius

1 Kanalas

2 Kanalas

3 Kanalas

Klaviatūra  
(įvedimo įrenginys)

Ekranas  
(išvedimo įrenginys)

Kietasis  
diskas  
(išorinė atmintis)

Supervizorinė atmintis

Puslapiavimo mechanizmas

000

255

Vartotojo atmintis

.  
.  
.  
.  
.

APL

IC  
R1  
R2  
C  
DS  
CS

MODE

TI

PI

SI

IOI

PTR

CHST[1]

CHST[2]

CHST[3]

## Techninės įrangos komponentai sudarantys realią mašiną:

* + - * Centrinis procesorius
      * Vartojo atmintis
      * Supervizorinė atmintis
      * Išorinė atmintis
      * Duomenų perdavimo kanalai
      * Įvedimo įrenginys - klaviatūra
      * Išvedimo įrenginys - ekranas
      * Puslapiavimo mechanizmas
  1. Centrinis procesorius  
     Procesoriaus paskirtis yra skaityti komandas iš atminties ir jas interpretuoti.  
     Procesorius gali būti naudojamas dviem rėžimais:
     + - Supervizoriaus
       - Vartotojo

Supervizoriaus rėžime procesorius vykdo komandas, kurios yra atsakingos už operacinės sistemos funkcionavimą, bet ne už vartotojo užduočių programas. Naudojantis supervizoriaus rėžimu darbas su ALP yra atliekamas naudojantis supervizoriaus atmintimi. Procesoriaus persijungia į supervizoriaus režimą pertraukimais arba sisteminiais kreipiniais.

## Realios mašinos procesorius turi tokius registrus:

* + - * R1, R2– 4 baitų bendrosios paskirties registrai
      * IC – 2 baitų komandų skaitiklis
      * C -– 1 baito loginis registras, gali įgyti reikšmes TRUE arba FALSE.
      * SF – 1 baito požymių registras
      * PTR – 4 baitų puslapių lentelės registras (aktyvus puslapių lentelės bloko numeris)
      * MODE – 1 baito registras, skirtas nustatyti mašinos režimą (supervizoriniu arba vartotojo)
      * TI – 2 baitų taimerio registras, kiekvienas atliktas komandos žingsnis padidina jį laiko vienetu
      * PI,SI – 2 baitų programinių ir supervizorinių pertraukimų registrai
      * IOI - 2 baitų įvedimo ir išvedimo pertraukimų registras
      * CHST[1] ... CHST[3] – kanalų būsenos registrai.

## Atmintys:

Vartotojo ir supervizorinė atmintys dalijasi realios mašinos atmintį.

* + - * Vartotojo Atmintis-pagrindinė atmintis, naudojama vartotojo; joje saugomos užduotys rezultatai, puslapio lentelės ir t.t.
      * Supervizoriaus atmintis -ją naudoja APL.
      * Išorinė atmintis-papildoma atmintis, šiuo atvėju kietasisi diskas.

## Išvedimo ir įvedimo įrenginiai

* + - * Klaviatūra - įvedimo įrenginys; naudojamas nuskaityti įvestas komandas
      * Ekranas- išvedimo įrenginys; naudojamas informacijos išvedimui ir atvaizdavimui ekrane

## Puslapiavimo mechanizmas

Puslapiavimo mechanizmas yra naudojamas virtualaus adreso atvaizdavimui į realų adresą atmintyje. Virtualiai mašinai išskiriama 16 atminties blokų. Kiekvienas lentelės puslapis (1 blokas) užpildomas realiais adresais. Naudojamas registras PTR( 4 baitų a0,a1,a2,a3), kuriame laikomi puslapių lentelės bloko realūs adresai. Naudojant PTR registro baitus a2 ir a3 , surandamas blokas. Pagal virtualaus adreso x1 ir x2 yra nustatomas puslapis. Bloko adresas kur yra x1 randamas pagal formulę [16\*(16\*a2 a3)+x1], o prie bloko adreso pridėjus x2 gauname virtualų adresą: 16\*[16\*(16\*a2 a3)+x1]x2.

## Duomenų perdavimo kanalai

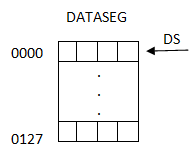
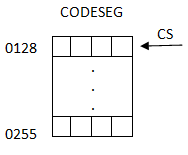
Skirti atminties ir įvedimo/išvedimo valdymui. Yra trys kanalai kiekvienas atsakingas už atitinkamus dalykus. Kanale vyksta tik rašymas arba skaitymas ir baigęs darbą jis informuoja centrinį procesorių ir sukelia pertraukimą padidindamas IOI registro reikšmę. Kanalų būsenos yra saugomos CHST[1] ... CHST[3] registruose.

# VIRTUALI MAŠINA

* 1. Pagrindinė operacinės sistemos funkcija – paslėpti visą jos sudėtingumą ir duoti programuotojui instrukcijų, su kuriomis jis galėtų dirbti, sąrašą. Todėl vienas iš operacinės sistemos tikslų yra paslėpti realią mašiną ir pateikti mums virtualią. Virtuali mašina - tai tarsi virtuali realios mašinos kopija, su kuria dirba vartotojas. VM yra lyg trapininkas tarp konkrečios mašinos ir jai taikomos programinės įrangos.

## Modeliuojamos virtualios mašinos loginių komponentų aprašymas.

* + 1. Virtualios mašinos atmintis susideda iš 16 blokų po 16 žodžių (iš viso 256 žodžiai), kurie yra po 4 B (32 bit). Atmintis turi 2 lygius segmentus: duomenų (DATASEG) – 8 blokai (128 žodžių), kodo (CODESEG) – 8 blokai (128 žodžių), į kuriuos bus įkeliamos atitinkamos programos dalys.  
       Kodo segmente esančios programos paskutinė komanda turi būti HALT.   
       Virtualios mašinos atmintis vizualiai atrodo taip:



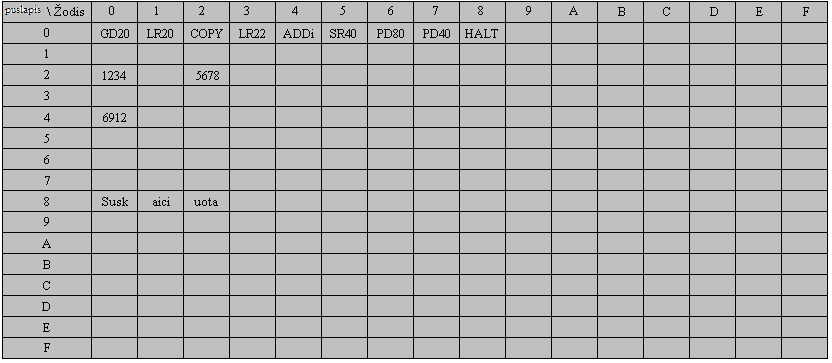
Kur,

DS – registras, kurio reikšmė yra rodyklė į duomenų segmentą atmintyje, o CS – registras, kurio reikšmė yra rodyklė į kodo segmentą atmintyje.

Puslapiavimo mechanizmas yra naudojamas virtualaus adreso atvaizdavimui į realų adresą atmintyje. Virtualiai mašinai išskiriama 16 atminties blokų. Kiekvienas lentelės puslapis (1 blokas) užpildomas realiais adresais. Naudojamas registras PTR( 4 baitų a0,a1,a2,a3), kuriame laikomi puslapių lentelės bloko realus adresai. Naudojant PTR registro baitus a2 ir a3 surandamas blokas. Pagal virtualaus adreso x1 ir x2 yra nustatomas puslapis. Bloko adresas kur yra x1 randamas pagal formulę [16\*(16\*a2 a3)+x1], o prie bloko adreso pridėjus x2 gauname virtualų adresą: 16\*[16\*(16\*a2 a3)+x1]x2.

Virtualiais adresais operuoja virtuali mašina, o realiais adresais – reali mašina. Ryšiai tarp realaus ir virtualaus adreso nusakomi per puslapiavimo mechanizmą.

Informacija apie kiekvieną procesą yra saugoma operacinės sistemos lentelėje, kuri yra realizuojama struktūrų linked list‘u - kiekviena struktūra vienam, dabar egzistuojančiam, procesui.



* + 1. Virtualios mašinos procesorius.  
       Virtualios mašinos procesoriaus registrai:
       - Komandų skaitliukas:
         1. **IC** – 2 baitų virtualios mašinos programos skaitiklis
       - Bendrosios paskirties registrai:
         1. **R1** – bendrosios paskirties registras
         2. **R2** – bendrosios paskirties registras
       - Požymių registrai:
         1. **C** – 1 baito loginis registras, požymius formuoja aritmetinės, o į juos reaguoja sąlyginio valdymo perdavimo komandos. Įgyjamos reikšmės: 0 – jeigu daugiau, 1 – jeigu lygu, 2 – jeigu mažiau
       - Segmentų registrai:
         1. **DS** – Data segment – rodyklė į duomenų segmentą atmintyje.
         2. **CS** – Code segment – rodyklė į kodo segmentą atmintyje.

### Virtualios mašinos komandų sistema.

Kiekvieną virtualios mašinos komandą sudaro 4B, tačiau priklausomai nuo komandos ne visi baitai turi būti užimti – jie gali būti ir tušti.

Komandos:

1. Duomenų persiuntimui iš atminties į registrus ir atvirkščiai:
   1. **LR** – Load Register – iš atminties baito x1x2 persiunčia į registrą R:  
      **LR x1x2** =) R:=[x1x2];
   2. **SR** – Save Register – iš registro R persiunčia į atminties baitą x1x2:   
      **SR x1x2** =) [x1x2]:=R;
2. Duomenų sukeitimui tarp registrų:
   1. **RR** – sukeičia registro R ir R2 reikšmes:  
      **RR** =) R:=R+R2, R2=R-R2, R=R-R2;
3. Aritmetinės komandos:
   1. **AD** – suma – prie esamos registro R reikšmės prideda reikšmę esančią x1x2 atminties baite, rezultatas patalpinamas registre R:   
      **AD x1x2** =) R:=R+[x1x2];
   2. **SB** – atimtis – iš esamos registro R reikšmės atimama reikšmė esanti x1x2 atminties baite, rezultatas patalpinamas registre R:   
      **SB x1x2** =) R:=R-[x1x2];
   3. **CR** – palyginimas – esamą registro R reikšmė yra lyginama su reikšme esančią x1x2 atminties baite, rezultatas patalpinamas registre C:   
      **CR x1x2**=)  
       if R>[x1x2] then C:=0;   
      if R=[x1x2] then C:=1;   
      if R<[x1x2] then C:=2;
4. Valdymo perdavimo:
   1. **JM** – besąlyginio valdymo perdavimas – valdymas perduodamas adresu 16\*x1+x2:  
      **JM x1x2**=) IC:=16\*x1+x2;
   2. **JA** – sąlyginio valdymo perdavimas (jeigu daugiau) – valdymas perduodamas jeigu C=0, valdymas perduodamas adresu 16\*x1+x2:   
      **JA x1x2**=) If C=0 then IC:= 16\*x1+x2;
   3. **JE** – sąlyginio valdymo perdavimas (jeigu lygu) – valdymas perduodamas jeigu C=1, valdymas perduodamas adresu 16\*x1+x2:   
      **JE x1x2**=) If C=1 then IC:= 16\*x1+x2;
   4. **JL** – sąlyginio valdymo perdavimas (jeigu mažiau) – valdymas perduodamas jeigu C=2, valdymas perduodamas adresu 16\*x1+x2:   
      **JL x1x2**=) If C=2 then IC:= 16\*x1+x2;
5. Darbo su bendra atminties sritimi (prieinama visoms vartotojo programoms; komandos leidžia į ją rašyti ir skaityti; sritis apsaugoma semaforais):
   1. **SM** – registro R įrašymas į bendrąją atmintį:   
      **SM x1x2** =)[16\*[16\*(16\*a2 a3)+x1]x2]:=R (pagal puslapiavimo mechanizmą);
   2. **LM** – iš bendrosios atminties įrašomas žodis į registrą R:   
      **LM x1x2** =) R:= [16\*[16\*(16\*a2 a3)+x1]x2] (pagal puslapiavimo mechanizmą);
6. Programos pabaigos:
   1. **HALT** – programos pabaigos komanda.
7. Įvedimo/Išvedimo:
   1. **GD** – įvedimas – iš įvedimo srauto paima 1 žodžio srautą ir jį įveda į atmintį pradedant atminties baitu 16\*x1+x2:   
      **GD x1x2**
   2. **PD** – išvedimas – iš atminties, pradedant atminties baitu 16\*x1+x2 paima 1 žodžio srautą ir jį išveda į ekraną:   
      **PD x1x2**

## Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais mechanizmo aprašymas.

VM duomenis skaito iš išorinės atminties (realizuotos failu kietajame diske), o rezultatą išveda į kompiuterio ekraną. Įvedimą/išvedimą kontroliuoja kanalų įrenginys.

* 1. Virtualios mašinos interpretuojamojo ar kompiliuojamo vykdomojo failo išeities teksto formatas. **Pavyzdžiui, kaip išskiriamas duomenų segmentas, kodo segmentas, kaip aprašomi duomenys ir t.t.)**

VM modelio įvedimo įrenginiui pateikiamas programos failas turi būti tokios struktūros:

DATASEG

.  
.  
.

CODESEG

.  
.  
.  
HALT

Atmintis yra išdėstyta nuosekliai: 128 žodžiai skirti DATASEG (nuo 0 iki 127) ir 128 žodžiai CODESEG (nuo 128 iki 255).

Duomenų segmento apraše galimi tokie atvejai:

DW - Išskiriamas vienas tuščias žodis skaitinei reikšmei.  
DW X - Išskiriamas vienas žodis ir į jį talpinama nurodyta skaitinė reikšmė.  
DB ssss - Išskiriamas vienas žodis ir į jį talpinami keturi nurodyti simboliai.  
DB nnnn - Tai rezervuota simbolinė konstanta, reiškianti \n.

Programa apskaičiuoja reiškinio „100 + 20 – 80“ reikšmę, bei ją išveda į ekraną.

000 | DATA  
001 | 100  
002 | 20  
003 | 80  
004 | Rezu  
005 | ltat  
009 | as y  
00A | ra:   
080 | CODE  
081 | LR 01  
082 | AD 02  
083 | SB 03  
084 | PD 04  
085 | PD 05  
086 | PD 06  
087 | PD 07  
088 | SR 10  
089 | PD 10  
08A | HALT

0FF |

## Modeliuojamos virtualios mašinos loginių komponentų sąryšio su realios mašinos techninės įrangos komponentais aprašymas.

Virtualiai mašinai atliekant komandas gali kilti pertraukimai. Jie apdorojami tik tada kai VM baigia vykdyti komandą.Tuomet reali mašina persijungia iš vartotojo režimo į supervizorinį.

Įvedimo/ Išvedimo veiksmas atliekamas supervizoriniu rėžimu, tam naudojama iniciavimo operacija StarIO – kuria nustatomi kanalai, jų panaudojimas ir tikrinamas užimtumas.

Norint iš supervizorinio rėžimo į vartotojo rėžimą darbo pratęsimui virtualioje mašinoje reikalinga pakrauti būsena tam naudojama operacija Slave(plr,c,r,ic) – kur registrų panaudojimas sutampa su realios mašinos.