

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

**Virtualios ir realios mašinos modelis**

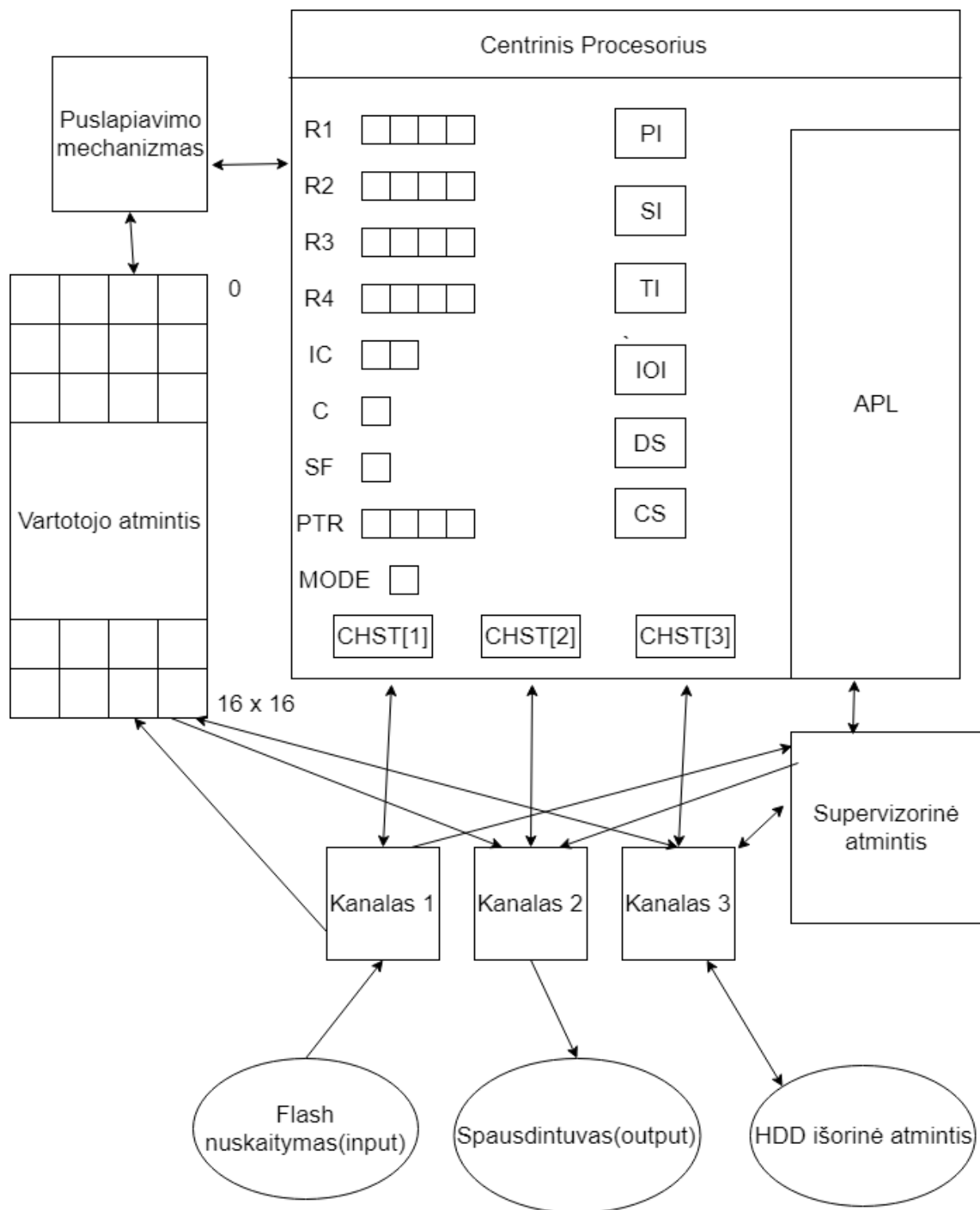
**Virtual and real machine model**

Atliko:	2 kurso 3 grupės studentai	
	Matas Savickis	(parašas)
	Justas Tvarijonas	(parašas)
	Greta Pyrantaitė	(parašas)
Darbo vadovas:	Mantas Grubliauskas Lekt.	(parašas)

## TURINYS

1. REALI MAŠINA .....	2
1.1. Techninės įrangos elementai sudarantys realią mašiną .....	3
1.2. Centrinis procesorius .....	3
1.3. Procesoriaus registrai .....	3
1.4. Atmintys .....	4
1.5. I/O .....	4
1.6. Puslapiavimo mechanizmas .....	4
1.7. Duomenų perdavimo kanalai .....	4
2. VIRTUALI MAŠINA .....	5
2.1. Funkcija.....	5
2.2. Modeliuojamos virtualios mašinos loginių komponentų aprašymas .....	5
2.2.1. Atmintis.....	5
2.2.2. Procesorius .....	6
2.2.3. Virtualios mašinos komandų sistema.....	7
2.3. Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais mechanizmo aprašymas.....	8
2.4. Virtualios mašinos interpretuojamojo ar kompiliuojamo vykdomojo failo išeities teksto formatas.	8
2.5. Modeliuojamos virtualios mašinos loginių komponentų sąryšio su realios mašinos techninės įrangos komponentais aprašymas.....	9
3. VIRTUALI MAŠINA OPERACINĖS SISTEMOS KONTEKSTE .....	10

# 1. Reali mašina



1 pav. Reali mašina

## **1.1. Techninės įrangos elementai sudarantys realią mašiną**

- Centrinis procesorius
- Vartotojo atmintis
- Supervizorinė atmintis
- Išorinė atmintis
- Duomenų perdavimo kanalai
- Įvedimo įrenginys - flash atmintinis
- Išvedimo įrenginys - spausdintuvas
- Puslapiavimo mechanizmas

## **1.2. Centrinis procesorius**

Centrinio procesoriaus paskirtis yra skaityti komandas iš atminties ir jas interpretuoti. Procesorius gali dirbti dviem režimais - supervizoriaus ir vartotojo. Supervizoriaus režime komandos yra apdorojamos aukšto lygio procesoriaus. Komandos vykdomos supervizoriaus režime yra skirtos operacinės sistemos funkcionavimui palaikyti. Procesorius persijungia į šį režimą pertaukimais arba sisteminiiais kreipiniais.

## **1.3. Procesoriaus registrai**

- R1, R2, R3, R4 - 4 baitų bendros paskirties registrai
- DS - registras rodantis duomenų segmento adresą
- CS - registras rodantis kodo segmento adresą
- IC - 2 baitų komandų skaitiklis
- C - 1 baito loginis registras (TRUE arba FALSE)
- SF - 1 baito požymių registras CF OF XX XXXZF - carry flag, overflow flag, zero flag
- PTR - 4 baitų puslapių lentelės registras(naudojamo puslapių lentelės numeris)
- MODE - registras, kurio reikšmė nusako procesoriaus darbo režimą(supervizorinis, vartotojo)
- PI - programinių pertraukimų registras
- SI - supervizorinių pertraukimų registras
- TI - taimerio registras
- CHST[1]...CHST[3] - kanalų būsenos registrai
- IOI - 2 baitų įvedimo ir išvedimo pertraukimų registras

## 1.4. Atmintys

Pagrindinę realios mašinos atmintį dalinasi vartotojo ir supervizorinė atmintis. Kaskart sukuriant naują virtualią mašiną panaudojama dalis realios mašinos atminties. Atminties dydis 16 blokų po 16 žodžių. Žodžio ilgis 4 baitai. Supervizorinę atmintį naudoja aukšto lygio procesorius, tačiau šios atminties savo rašomoje programoje nenaudosim. Taip pat yra išorinė atmintis kietojo disko pavidalu.

## 1.5. I/O

Komandų įvedimui naudojamas „flash atmintinių“ nuskaitymo įrenginys. Išvedimui naudojamas spausdintuvas.

## 1.6. Puslapiavimo mechanizmas

Puslapiavimo mechanizmas yra metodas skirtas virtualios atminties adreso parodymui į realios atminties adresą. Kiekvienai virtualiai mašinai skiriama 4 atminties blokai. Paskutinis lentelės blokas yra užpildomas realiais adresais. Naudojamas PTR(4 baitų a0,a1,a2,a3) registras kuriame laikomas puslapių lentelės bloko realūs adresai. Pagal a2 ir a3 baitus surandame bloką. Pagal virtualaus adreso x1 ir x2 yra nustatomas puslapis. Bloko adresas kur x1 randamas  $[4*(4*a2\ a3)+x1]$ , o prie bloko adreso pridėjus x2 gauname virtualų adresą:  $4*[4*(4*a2\ a3)+x1]x2$ .

## 1.7. Duomenų perdavimo kanalai

Skirti I/O ir atminties valdymui. Modelyje egzistuoja trys kanalai iš kurie kiekvienas atsakingas už skirtingus dalykus. Kanale vyksta arba rašymas arba skaitymas ir pabaigęs darbą jis informuoja centrinį procesorių ir sukelia pertraukimą padidindamas IOI registro reikšmę. Kanalų būsenos yra saugomos CHST[1]...CHST[3] registruose.

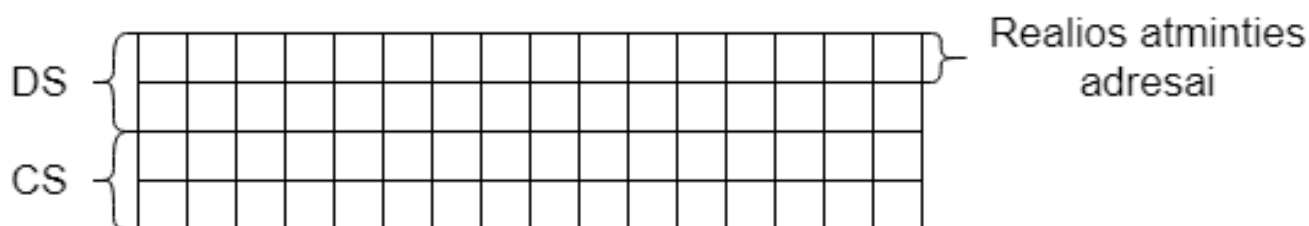
## 2. Virtuali mašina

### 2.1. Funkcija

Virtuali mašina yra skirta paslepti realios mašinos sudėtingumą ir suteikti vartotojui instrukcijų sąrašą su kuriuo jis galėtų dirbti. Todėl operacinės sistemos paskirtis ir yra paslėpti realią mašiną ir duoti mums virtualią. Virtuali mašina taip pat suteikia darbų pasidalijimą kurio dėk galima paleisti kelias virtualias mašinas ir tokiu būdu ant kiekvienos iš jų atlikti skirtingas užduotis.

### 2.2. Modeliuojamos virtualios mašinos loginių komponentų aprašymas

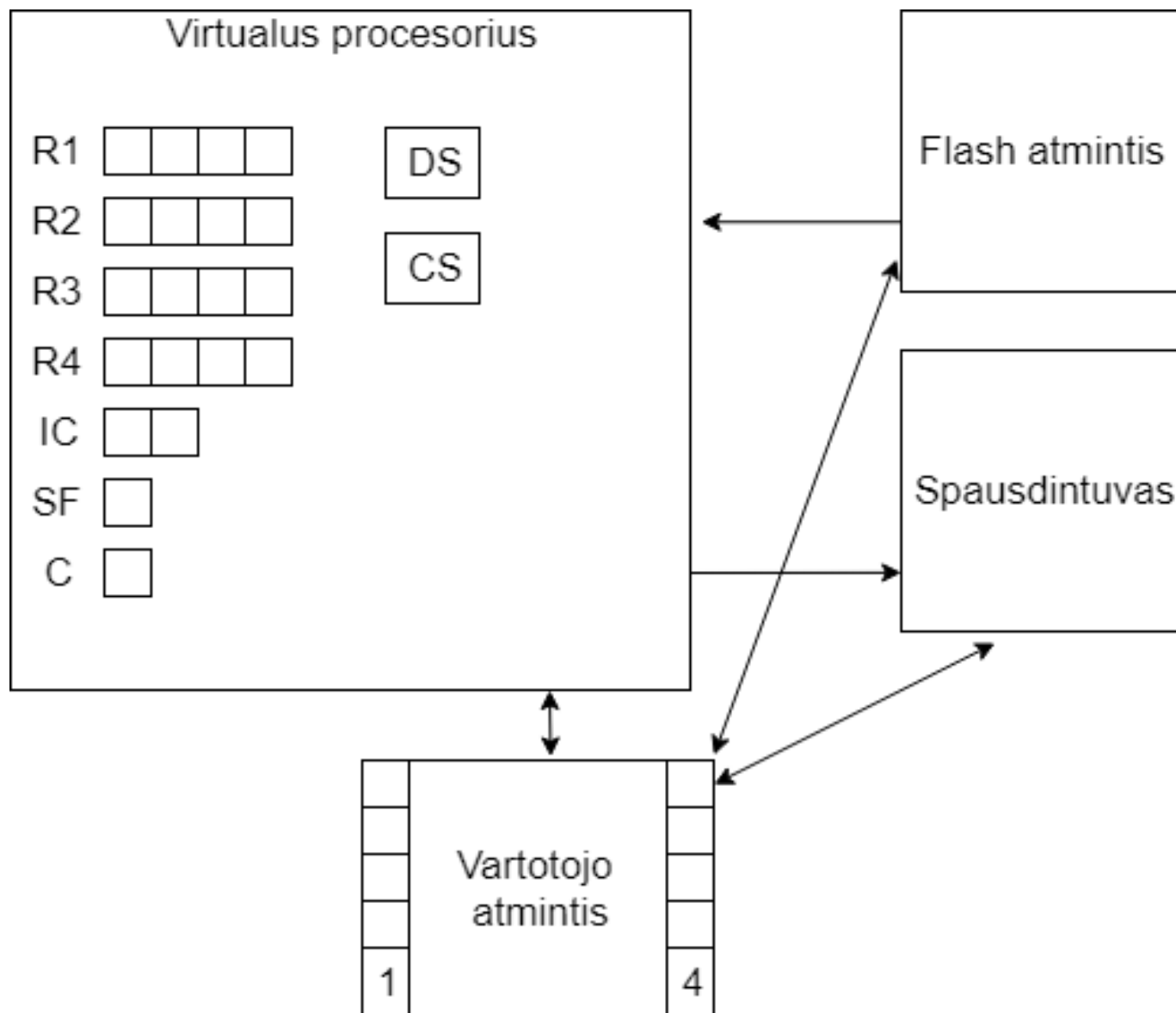
#### 2.2.1. Atmintis



2 pav. Virtualios mašinos atmintis

Virtualios mašinos atmintis susideda iš 4 blokų po 16 žodžių iš viso 64 žodžiai po 4 baitus arba 32 bitus. Pirmi du atminties blokai bus paskirti duomenų segmentui(DS), o paskutiniai du blokai kodo segmentui(CS). Norint pasinaudoti puslapiavimo mechanizmu pirmas blokas bus išskiriamas surašyti realiems adresams. Naudosime registrą PTR(4 baitai a0 a1 a2 a3) kuriame laikomas puslapių lentelės bloko realūs adresai. Per a2 ir a3 surandame bloką. Pagal x1 ir x2 nustatomas puslapis. Bloko adresas, kur yra x1 randamas pagal formulę  $[4*(4*a2\ a3)+x1]*x2$ , o prie bloko adreso pridėjus x2 gauname virtualų adresą  $4*[4*(4*a2\ a3)+x1]*x2$ . Efektyviai naudotis Virtualia atmintimi įvesime du papildomus registus DS ir CS. DS registro reikšmė bus rodyklė į duomenų segmento atmintį, o CS reikšmė rodyklė į kodo segmento atmintį.

### 2.2.2. Procesorius



3 pav. Virtualios mašinos procesorius

Registrai:

- R1...R4 - Bendros paskirties registrai
- IC - komandos skaitliukas, rodo sekančios komandos adresą
- SF - status flagas rodantis programos būseną
- C - loginis registras(TRUE arba FALSE)
- DS - rodo į duomenų segmentą atmintyje
- CS - rodo į kodo segmentą atmintyje

### 2.2.3. 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:

- (a) LR – Load Register – iš atminties baito  $x1x2$  persiunčia į registrą R:  $LR\ x1x2 \Rightarrow R := [x1x2]$ ;
- (b) SR – Save Register – iš registro R persiunčia į atminties baitą  $x1x2$ :  $SR\ x1x2 \Rightarrow [x1x2] := R$ ;

2. Duomenų sukeitimui tarp registrų:

- (a) RR – sukeičia registro R ir R2 reikšmes:  $RR \Rightarrow R := R + R2, R2 := R - R2, R := R - R2$ ;

3. Aritmetinės komandos:

- (a) AD – suma – prie esamos registro R reikšmės prideda reikšmę esančią  $x1x2$  atminties baite, rezultatas patalpinamas registre R:  $AD\ x1x2 \Rightarrow r1 := r1 + [x1x2]$ ;
- (b) SB – atimtis – iš esamos registro R reikšmės atimama reikšmė esanti  $x1x2$  atminties baite, rezultatas patalpinamas registre R:  $SB\ x1x2 \Rightarrow r1 := r1 - [x1x2]$ ;
- (c) CR – palyginimas – esamą registro R reikšmę yra lyginama su reikšme esančią  $x1x2$  atminties baite, rezultatas patalpinamas registre C:  $CR\ x1x2 \Rightarrow \text{if } r1 > [x1x2] \text{ then } cf := 0, zf := 0; \text{ if } r1 = [x1x2] \text{ then } zf := 1; \text{ if } r1 < [x1x2] \text{ then } cf := 1$ ;
- (d) MU  $x1x2$  – daugyba,  $r1 := r1 * [x1x2]$ .
- (e) DI  $x1x2$  – dalyba,  $r1 := r1 / [x1x2], r2 := r1 \% [x1x2]$ .

4. Valdymo perdavimo:

- (a) JU – besąlyginio valdymo perdavimas – valdymas perduodamas adresu  $16 * x1 + x2$ :  $JU\ x1x2 \Rightarrow IC := 16 * x1 + x2$ ;
- (b) JG – sąlyginio valdymo perdavimas (jeigu daugiau) – valdymas perduodamas jeigu  $C=0$ , valdymas perduodamas adresu  $16 * x1 + x2$ :  $JG\ x1x2 \Rightarrow \text{If } C=0 \text{ then } IC := 16 * x1 + x2$ ;
- (c) JE – sąlyginio valdymo perdavimas (jeigu lygu) – valdymas perduodamas jeigu  $C=1$ , valdymas perduodamas adresu  $16 * x1 + x2$ :  $JE\ x1x2 \Rightarrow \text{If } C=1 \text{ then } IC := 16 * x1 + x2$ ;
- (d) JL – sąlyginio valdymo perdavimas (jeigu mažiau) – valdymas perduodamas jeigu  $C=2$ , valdymas perduodamas adresu  $16 * x1 + x2$ :  $JL\ x1x2 \Rightarrow \text{If } C=2 \text{ then } IC := 16 * x1 + x2$ ;

5. Darbo su bendra atminties sritimi (prieinama visoms vartotojo programoms; komandos leidžia į ją rašyti ir skaityti; sritis apsaugoma semaforais):

- (a) SM – registro R įrašymas į bendrąją atmintį:  $SM\ x1x2 \Rightarrow [16 * [16 * (16 * a2\ a3) + x1]x2] := R$  (pagal puslapiavimo mechanizmą);
- (b) LM – iš bendrosios atminties įrašomas žodis į registrą R:  $LM\ x1x2 \Rightarrow R := [16 * [16 * (16 * a2\ a3) + x1]x2]$  (pagal puslapiavimo mechanizmą);

6. Programos pabaigos:



(a) HALT – programos pabaigos komanda.

7. Įvedimo/Išvedimo:

(a) GD – įvedimas – iš įvedimo srauto paima 1 žodžio srautą ir jį įveda į atmintį pradedant atminties baitu  $16 \cdot x_1 + x_2$ : GD  $x_1 x_2$

(b) PD – išvedimas – iš atminties, pradedant atminties baitu  $16 \cdot x_1 + x_2$  paima 1 žodžio srautą ir jį išveda į ekraną: PD  $x_1 x_2$

8. Loginės:

(a) AND –  $r_1 := r_1 \text{ and } r_2$

(b) XOR –  $r_1 := r_1 \text{ xor } r_2$

(c) OR –  $r_1 := r_1 \text{ or } r_2$ .

(d) NOT –  $r_1 := \text{not } r_1$ .

9. Operavimo failais:

(a) FC – uždaromas failas, kurio handleris yra  $r_1$ .

(b) FO  $x_1 x_2$  – atidaromas  $[x_1 x_2]$  pavadinimo failas. Handleris įrašomas į  $r_1$ .

(c) FR  $x_1 x_2$  –  $r_1$  – handleris,  $r_2$  – adresas, iš kur skaitome.  $16 \cdot x_1 + x_2$  – virtualios atminties vieta, į kurią įrašysime.

(d) FW  $x_1 x_2$  – handleris,  $r_2$  – adresas, iš kur skaitome.  $16 \cdot x_1 + x_2$  – virtualios atminties vieta, iš kurios rašysime į failą.

(e) FD – ištrinamas failas, kurio handleris yra  $r_1$ .

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

VM duomenis skaito iš flash atminties (realizuotos failu kietajame diske), o rezultatą išveda spausdintuvas. Įvedimą/išvedimą kontroliuoja kanalų įrenginys.

### 2.4. Virtualios mašinos interpretuojamojo ar kompiliuojamo vykdomojo failo išeities teksto formatas.

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 ?????????? papildyti !!!!!!!.

Programa apskaičiuoja reiškinių „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 |

## **2.5. 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 režimu, tam naudojama iniciavimo operacija StarIO – kuria nustatomi kanalai, jų panaudojimas ir tikrinamas užimtumas.

Norint iš supervizorinio režimo į vartotojo režimą darbo pratęsimui virtualioje mašinoje reikalinga pakrauti būseną tam naudojama operacija Slave(plr,c,r,ic) – kur registų panaudojimas sutampa su realios mašinos.

### **3. Virtuali mašina operacinės sistemos kontekste**

Operacinei sistemai turi būti pateikiamas užduočių rinkinys (programa). Tam reikalinga specifinė užduočių pateikimo kalba. Kiekviena užduotis suformuojama kaip failas. Užduotis sudaryta iš pateikiamų duomenų ir rezultatų. Vykdam užduotį ji yra išskaidoma į dalis: užduotis saugoma išorinėje atmintyje, kai ji paruošta vykdymui. Užduotis tiesioginės sąveikos su fiziniais įrenginiais neturi, tik su virtualiais.