

# Virtualios ir realios mašinos projektas

Atilko:  
Irmantas Varačisnkas,  
Karolis Šimaitis,  
Miglė Augustinaitė

# Turinys

<b>1</b>	<b>Projekto sąlygos</b>	<b>2</b>
<b>2</b>	<b>Realios mašinos modelis</b>	<b>3</b>
2.1	Realios mašinos brėžinys . . . . .	3
2.2	Realios mašinos registrai . . . . .	3
2.3	Taimerio mechanizmas . . . . .	4
2.4	Supervizoriniai pertraukimai . . . . .	4
2.5	Programiniai pertraukimai . . . . .	5
2.6	Realios mašinos režimai . . . . .	5
2.7	Puslapių transliacija . . . . .	5
2.8	Swappingas . . . . .	5
2.9	Įvedimas ir išvedimas . . . . .	5
2.10	Atminties įrenginiai . . . . .	6
2.11	Išorinė atmintis . . . . .	6
<b>3</b>	<b>Virtualios mašinos modelis</b>	<b>7</b>
3.1	Virtualios mašinos schema . . . . .	7
3.2	Virtualios mašinos samprata . . . . .	7
3.3	Virtualios mašinos loginiai komponentai . . . . .	7
3.3.1	Virtualios mašinos atmintis . . . . .	7
3.3.2	Virtualios mašinos procesorius . . . . .	7
3.3.3	Virtualios mašinos komandų sistema . . . . .	8
<b>4</b>	<b>Programos pavyzdys</b>	<b>10</b>

# 1 Projekto sąlygos

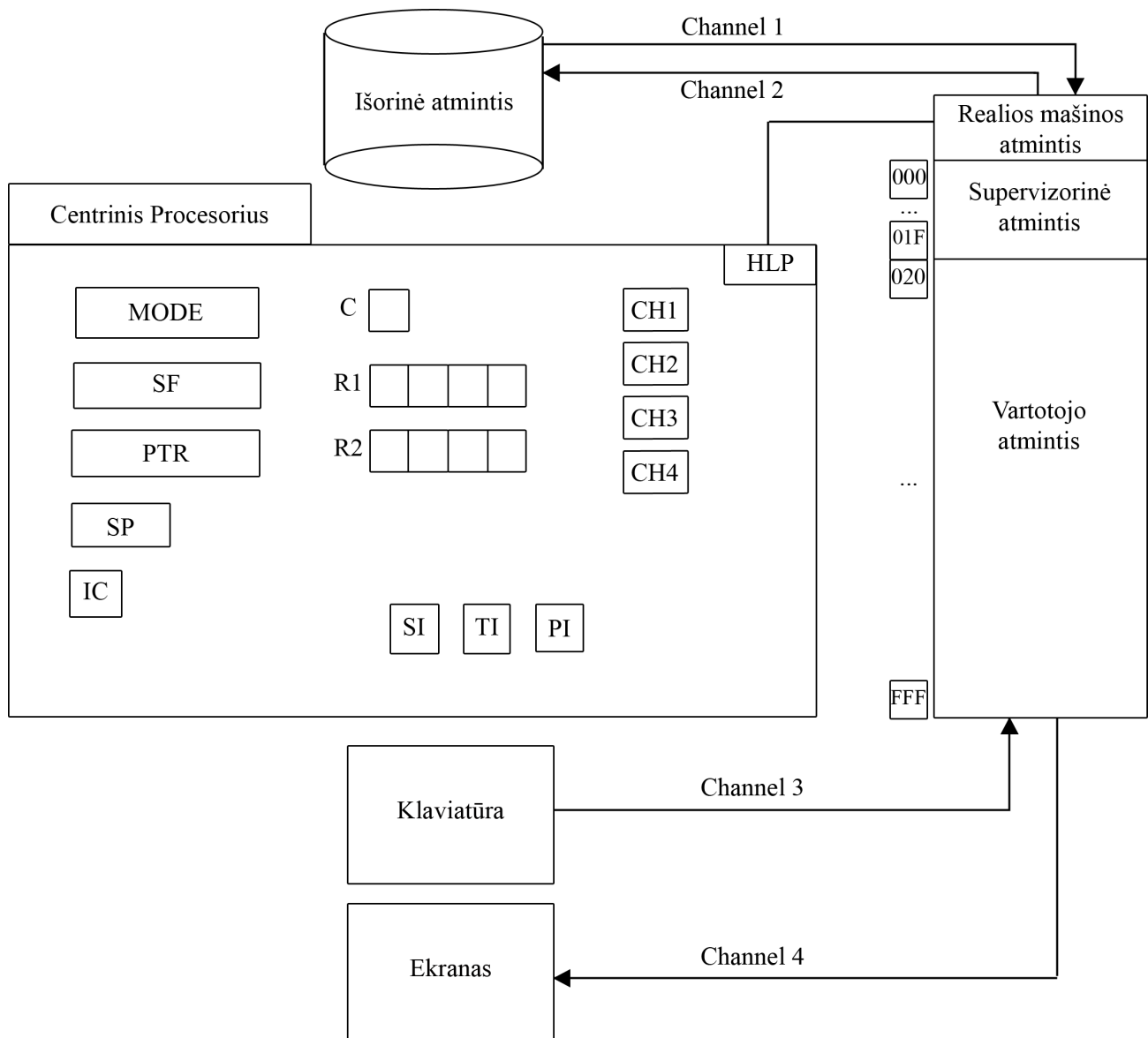
Projektuojama interaktyvi OS.

**Virtualios mašinos** procesoriaus komandos operuoja su duomenimis, esančiais registruose ir ar atmintyje. Yra komandos duomenų persiuntimui iš atminties į registrus ir atvirkščiai, aritmetinės (sudėties, atimties, daugybos, dalybos, palyginimo), sąlyginio ir besąlyginio valdymo perdavimo, įvedimo, išvedimo, darbo su failais (atidarymo, skaitymo, rašymo, uždarymo, sunaikinimo) ir programos pabaigos komandos. Registrai yra tokie: komandų skaitiklis, bent du bendrosios paskirties registrai, požymių registras (požymius formuoja aritmetinės, o į juos reaguoja sąlyginio valdymo perdavimo komandos). Atminties dydis yra 16 blokų po 16 žodžių (žodžio ilgį pasirinkite patys).

**Realios mašinos** procesorius gali dirbti dviem režimais: vartotojo ir supervizoriaus. Virtualios mašinos atmintis atvaizduojama į vartotojo atmintį naudojant puslapių transliaciją. Yra taimeris, kas tam tikrą laiko intervalą generuojantis pertraukimus. Įvedimui naudojama klaviatūra, išvedimui - ekranas. Yra išorinės atminties įrenginys - kietasis diskas. Vartotojas, dirbantis su sistema, programas paleidžia interaktyviai, surinkdamas atitinkamą komandą. Laikoma, kad vartotojo programos yra realios mašinos kietajame diske, į kurią jos patalpinamos „išorinėmis“, modelio, o ne projektuojamos OS, priemonėmis.

## 2 Realios mašinos modelis

### 2.1 Realios mašinos brėžinys



### 2.2 Realios mašinos registrai

1. HLP - bet kuris aukšto lygio kalbos procesorius. Vartotojo režime HLP vykdo užduoties programą.
2. MODE - realios mašinos režimo registras. Dydis - 1 baitas. Jei reikšmė 0, dirbama supervizoriaus režimu, jei reikšmė nėra 0, tada dirbama vartotojo režimu.
3. SF - požymių registras. Dydis - 1 baitas. Parodo procesoriaus būseną po aritmetinio veiksmo.  
Požymių registro struktūra: X X X X X CF ZF OF

- X - nenaudojamas.

- CF - carry flag. Rezultatas netilpo į skaičiaus be ženklo režius.
  - ZF - zero flag. Rezultatas yra nulis.
  - OF - overflow flag. Rezultatas netilpo į skaičiaus su ženklu režius.
4. PTR - puslapių lentelės registras. Dydis - 2 baitai. Vyresnysis baitas saugo puslapių lentelės bloko numerį, jaunesnysis baitas saugo puslapių lentelės dydį.
  5. SP - steko rodyklė. Dydis - 1 baitas. Rodo į virtualios mašinos steko viršūnę.
  6. IC - instrukcijų skaitliukas. Dydis - 1 baitas. Rodo virtualios mašinos einamąją instrukciją.
  7. C - loginis trigeris. Dydis - 1 baitas. 0 yra false, visa kita yra true.
  8. R1, R2 - bendros paskirties registrai. Dydis - po 4 baitus. Skirti atlikti komandoms.
  9. Kanalių registrai. Dydis - po 1 baitą.
    - CH1 - registras rodantis ar yra atliekamas persiuntimas iš išorinės atminties į realią atmintį.
    - CH2 - registras rodantis ar yra atliekamas persiuntimas iš realios atminties į išorinę atmintį.
    - CH3 - registras rodantis ar atliekamas įvedimas iš klaviatūros.
    - CH4 - registras rodantis ar atliekamas išvedimas į ekraną.
  10. SI - supervizoriaus pertraukimų registras. Dydis - 1 baitas.
  11. PI - programinių pertraukimų registras. Dydis - 1 baitas.
  12. TI - taimerio registras. Dydis - 1 baitas.

## 2.3 Taimerio mechanizmas

Po kiekvienos įvykdytos komandos taimerio registras yra sumažinamas vienetu. Pradinę taimerio reikšmę gali nustatyti vartotojas, tačiau numatytoji reikšmė yra 10.

Kai šio registro reikšmė pasiekia nulį, iškviečiamas supervizorinis taimerio pertraukimas. Įvykdžius taimerio pertraukimo apdorojimo procedūrą, taimerio registro reikšmė vėl tampa 10.

## 2.4 Supervizoriniai pertraukimai

Supervizoriniai pertraukimai yra iškviečiami tuomet, kai SI registras nėra lygus nuliui arba TI registras yra lygus nuliui.

Pertraukimai pagal SI registro reikšmę:

- 0 - jokio pertraukimo.
- 1 - Full stack. Kyla, kai SP reikšmė tampa F. Naudojama swapingo mechanizmui.
- 2 - Empty stack. Kyla, kai stekas tampa tuščiu. Naudojama swapingo mechanizmui.
- 3 - HALT. Programa baigė darbą.

## 2.5 Programiniai pertraukimai

Programiniai pertraukimai yra išskviečiami tuomet, kai PI registras nėra lygus nuliui. Pertraukimai pagal PI registro reikšmę:

- 0 - jokio pertraukimo.
- 1 - Undefined operation code. Neteisingas operacijos kodas.
- 2 - Undefined address. Neteisingas adresas.
- 3 - Incorrect assignment. Neteisingas priskyrimas.
- 4 - Input. Įvedimas iš klaviatūros.
- 5 - Output. Išvedimas į ekraną.

## 2.6 Realios mašinos režimai

Reali mašina gali vykdyti darbą dvejais režimais: supervizoriaus arba vartotojo.

Supervizoriaus režimu programa bus vykdoma nuo pradžios iki galo, ją nutraukti gali tik pertraukimai. Visi realios mašinos resursai yra prieinami.

Vartotojo režimu programa gali vykti kelios programos vienu metu, po kiekvieno veiksmo taimeris mažinamas ir kai jis pasiekia nulį, toliau procesoriaus darbą nustato prioritetų automatas. Prieinama atmintis yra tik ta, kurią išskyrė procesorius.

## 2.7 Puslapių transliacija

Puslapių transliavimo mechanizmas naudojamas, tam, kad galėtumė susieti virtualią ir realią atmintis. registre PTR bus laikomas puslapiavimo lentelės bloko adresas, kurio kiekviename žodyje bus laikomas virtualios mašinos puslapio takelio adresas.  $a_0$  ir  $a_1$  naudojami naudojamas gauti puslapių lentelės adresą, naudojama formulė yra tokia:

$$10 * a_0 + a_1$$

Dviženklis adresas  $x_0$ ,  $x_1$  realiojoje mašinoje apskaičiuojamas taip:

$$10 * [10 * (10 * a_0 + a_1) + x_0] + x_1$$

## 2.8 Swappingas

Kai steke pritrūksta vietos, dalis steko turinio yra perkeliamas į išorinę atmintį.

Kai stekas tampa tuščias, į jį grąžinami duomenys iš išorinės atminties, jei jų yra.

Kiekvienai programai yra sukuriamas laikinas failas, kur ir bus dedamas stekas, jei to prireiks. Programai pasibaigus šis failas bus ištrinamas.

## 2.9 Įvedimas ir išvedimas

Kadangi projektuojama OS yra interaktyvi, vartotojas gali įvesti komandas ir/ar duomenis ir taip reguliuoti realios mašinos darbą ir matyti šių komandų rezultatus.

Įvedimo įrenginys - klaviatūra. Suvedama komanda arba jos duomenys ir paspaudžiamas mygtukas ENTER, kad būtų paleidžiamas arba pratęsiamas jos vykdymas.

Išvedimo įrenginys - ekranas. Įvykdytų komandų rezultatai gali būti išvedami į ekraną.

## 2.10 Atminties įrenginiai

**Išorinė atmintis** - kietasis diskas. Jame duomenys bus išsaugoti net ir išjungus realią mašiną.

**Realios mašinos atmintis** - atmintis esanti realioje mašinoje. Ji suskirstyta į 16 blokų po 16 žodžių, vieno žodžio dydis yra 32 bitai. Ją toliau skirstome į supervizorinę ir vartotojo atmintį:

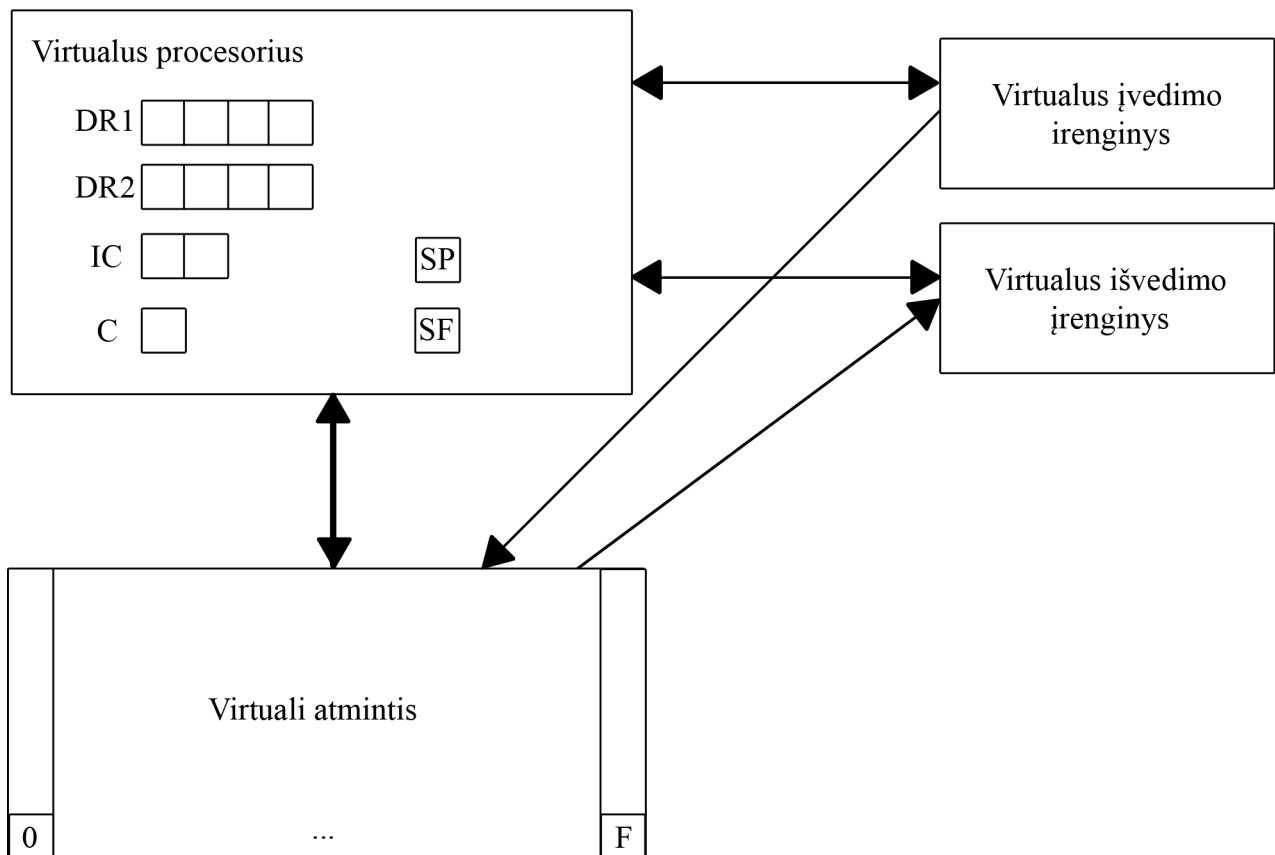
- **Supervizorinė atmintis** - 2 blokai išskirti realios mašinos atminties pradžioje. Jie nėra prieinami vartotojui.
- **Vartotojo atmintis** - Visa likusi realios mašinos atmintis. Ją galima skirstyti virtualioms mašinoms.

## 2.11 Išorinė atmintis

Išorinė atmintis bus realizuojama failu kietajame diske. Darbą su ja vykdys HLP.

## 3 Virtualios mašinos modelis

### 3.1 Virtualios mašinos schema



### 3.2 Virtualios mašinos samprata

**Virtuali mašina** - tai tarsi realios mašinos kopija. Virtuali mašina yra sudaryta iš realios mašinos komponentų, tokių kaip procesorius, atmintis, įvedimo/išvedimo įrenginiai. Jiems yra suteikiama paprastesnė nei reali vartotojo sąsaja.

### 3.3 Virtualios mašinos loginiai komponentai

#### 3.3.1 Virtualios mašinos atmintis

Virtualiai mašinai skirta 16 blokų po 16 žodžių, vieno žodžio dydis yra 32 bitai. Vienas blokas yra skirtas virtualios mašinos stekui.

#### 3.3.2 Virtualios mašinos procesorius

Virtualios mašinos registrai:

1. Duomenų registrai:

- DR1 - Data Register. Dydis - 4 baitai. Naudojamas duomenų pakrovimui į jį iš atminties ir iš jo į atmintį. Taip pat gali būti naudojamas operacijose.
- DR2 - Data Register. Dydis - 4 baitai. Naudojamas duomenų pakrovimui į jį iš atminties ir iš jo į atmintį. Taip pat gali būti naudojamas operacijose.



## 2. Segmentų registrai:

- CS - Code Segment. Rodyklė rodanti į kodo segmentą atmintyje.
- DS - Data Segment. Rodyklė rodanti į duomenų segmentą atmintyje.
- ST - Stack Segment. Rodyklė rodanti į steko segmentą atmintyje.

## 3. Nuorodų registrai:

- IC - Instruction Counter. Saugoma einamosios komandos žodžio indeksas.
- SP - Stack Pointer. Saugomas steko viršūnės žodžio indeksas.

## 4. Loginiai registrai:

- SF - Status Flag. Rodo procesoriaus būseną po aritmetinio veiksmo.

### 3.3.3 Virtualios mašinos komandų sistema

Vieną komandą sudaro 4 baitai, tačiau nebūtinai visi baita privalo būti naudojami.

#### 1. Aritmetinės komandos

- ADD - sudeda du viršutinius steko elementus, sumažina steko virūnės rodyklę vienetu ir įdeda rezultatą į steką. Gali pakeisti visus flag'us.
- SUB - atima viršutinį steko elementą iš prieš paskutinio steko elemento, sumažina steko virūnės rodyklę vienetu ir įdeda rezultatą į steką. Gali pakeisti visus flag'us.
- MUL - sudaugina du viršutinius steko elementus, sumažina steko virūnės rodyklę vienetu ir įdeda rezultatą į steką. Gali pakeisti visus flag'us.
- DIV - padalina antrą nuo viršaus steko elementą iš viršutinio steko elemento, sumažina steko virūnės rodyklę vienetu ir įdeda rezultatą. Gali pakeisti visus flag'us.

#### 2. Loginės komandos

- AND - įvykdo AND operaciją tarp viršutinių steko elementų, sumažina steko virūnės rodyklę vienetu ir įdeda rezultatą. Gali pakeisti ZF.
- OR - įvykdo OR operaciją tarp viršutinių steko elementų, sumažina steko virūnės rodyklę vienetu ir įdeda rezultatą. Gali pakeisti ZF.
- XOR - įvykdo XOR operaciją tarp viršutinių steko elementų, sumažina steko virūnės rodyklę vienetu ir įdeda rezultatą. Gali pakeisti ZF.
- NOT - įvykdo NOT operaciją viršutiniam steko elementui, sumažina steko virūnės rodyklę vienetu ir įdeda rezultatą. Gali pakeisti ZF.

#### 3. Palyginimo komandos

- CMP - palygina du viršutinius steko elementus ir nustato SF. Jei elementai lygūs, tada  $ZF = 1$ . Jei viršutinis elementas didesnis, tada  $CF = 0$  ir  $ZF = 0$ . Jei viršutinis elementas mažesnis,  $CF = 1$ .

#### 4. Darbo su duomenimis komandos

- LWxy - į DR1 įdedamas žodis esantis duomenų segmente adresu  $10 * x + y$ .

- SWxy - į duomenų segmentą, adresu  $10 * x + y$  įdedama DR1 reikšmė.
- PRNT - iškviečiamas pertraukimas, spausdinantis eilutę į ekraną. Eilutės adresas nurodytas steko viršūnėje, eilutės ilgis nurodytas registre DR1
- PRNS - iškviečiamas pertraukimas, spausdinantis steko viršūnę kaip skaičių į ekraną.

#### 5. Darbo su steku komandos

- PUSH - steko viršūnės rodyklė padidinama vienu ir į steko viršūnę idedama DR1 reikšmė.
- POP - į DR1 įdedamas žodis esantis steko viršūnėje ir steko viršūnės rodyklė sumažinama vienetu.

#### 6. Valdymo perdavimo komandos

- JMxy - besąlyginio valdymo perdavimo komanda. Valdymas perduodamas kodo segmento žodžiui  $10 * x + y$ , t.y.  $PC = 10 * x + y$ .
- JExy - sąlyginio valdymo perdavimo komanda. Jei  $ZF = 1$  (elementai yra lygūs), tada valdymas perduodamas kodo segmento žodžiui  $10 * x + y$ .
- JAxy - sąlyginio valdymo perdavimo komanda. Jei  $CF = 0$  ir  $ZF = 0$  (viršūnės elementas yra didesnis), tada valdymas perduodamas kodo segmento žodžiui  $10 * x + y$ .
- JLxy - sąlyginio valdymo perdavimo komanda. Jei  $CF = 1$  (viršūnės elementas yra mažesnis), tada valdymas perduodamas kodo segmento žodžiui  $10 * x + y$ .

#### 7. Failų rašymo, skaitymo komandos

- FR - DR1 registre bus saugoma į kurią vietą duomenų segmente įdėti įvestus duomenis.
- FW - DR1 registre bus saugoma kokio ilgio eilutė bus įrašyta į failą. DR2 registre bus saugomas žinutės adresas.

#### 8. Darbo pabaigos komanda

- HALT - parodo programos pabaigą.

Kodo struktūra turi būti tokia:

```
DATASEG
...
CODESEG
...
HALT
```

DATASEG viduje galima įdėti duomenis į duomenų segmentą naudojant DW komandą.

## 4 Programos pavyzdys

Mūsų programos pavyzdys atliks aritmetinius veiksmus ir juos išspausdins į ekraną.  
Programa skaičiuos šį reiškinį:  $45^2 + 13 * 4 - 3$

```
DATASEG
DW 45
DW 13
DW 4
DW 3
CODESEG
LW01
PUSH
PUSH
MUL
LW02
PUSH
LW03
PUSH
MUL
ADD
LW04
PUSH
SUB
PRNS
HALT
```