Matematikos ir informatikos fakultetas

Programų sistemų II kursas, 2 grupė

**Virtualios ir realios mašinos projektas**

Darbą atliko:

Zorian Ivanec

Dominykas Kiauleikis

Tomas Bareikis

Vilnius

2012 m. vasaro 28 d.

# Turinys

[Turinys 2](#_Toc321849481)

[1. Realios mašinos aprašas 3](#_Toc321849482)

[1.1. Užduotis 3](#_Toc321849483)

[1.2. Techninės įrangos komponentai, kurie sudaro realią mašiną 3](#_Toc321849484)

[1.3. Realios mašinos techninės įrangos komponentų detalizavimas 3](#_Toc321849485)

[1.3.1. Procesorius: 3](#_Toc321849486)

[1.3.2. Procesoriaus registrai: 3](#_Toc321849487)

[1.3.3. Operatyvioji atmintis: 4](#_Toc321849488)

[1.3.4. Išorinė atmintis: 4](#_Toc321849489)

[1.4. Realios mašinos techninės įrangos komponentų išsidėstymo vienas kito atžvilgiu ir tarpusavio sąveikos schema. 5](#_Toc321849490)

[2. Virtualios mašinos aprašas 6](#_Toc321849491)

[2.1. Modeliuojamos virtualios mašinos sampratos aprašymas 6](#_Toc321849492)

[2.2. Modeliuojamos virtualios mašinos loginių komponentų aprašymas 6](#_Toc321849493)

[2.2.1. Virtualios mašinos atmintis 6](#_Toc321849494)

[2.2.2. Virtualios mašinos procesorius 7](#_Toc321849495)

[2.2.3. Virtualios mašinos komandų sistema 7](#_Toc321849496)

[2.3. Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais mechanizmo aprašymas 10](#_Toc321849497)

[2.4. Virtualios mašinos interpretuojamojo ar kompiliuojamo vykdomojo failo išeities teksto formatas 10](#_Toc321849498)

[2.5. Modeliuojamos virtualios mašinos loginių komponentų sąryšio su realios mašinos techninės įrangos komponentais aprašymas 10](#_Toc321849499)

[3. Virtuali mašina visos operacinės sistemos kontekste 10](#_Toc321849500)

[4. Naudota literatūra 10](#_Toc321849501)

# Realios mašinos aprašas

## Užduotis

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, palyginimo), sąlyginio ir besąlyginio valdymo perdavimo, įvedimo, išvedimo, darbo su bendra atminties sritimi (prieinama visoms vartotojo programoms; komandos leidžia į ją rašyti ir skaityti; sritis apsaugoma semaforais) 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 kurį jos patalpinamos „išorinėmis“, modelio, o ne projektuojamos OS, priemonėmis.

## Techninės įrangos komponentai, kurie sudaro realią mašiną

2 procesoriai,

Operatyvioji atmintis,

Išorinė atmintis,

Duomenų perdavimo kanalai,

Įvedimo įrenginys - klaviatūra,

Išvedimo įrenginys - ekranas,

Išorinės atminties įrenginys - kietasis diskas,

Papildomi įrenginiai-2 garsiakalbiai.

## Realios mašinos techninės įrangos komponentų detalizavimas

### Procesorius:

Procesorius gali dirbti dviem režimais – supervizoriaus arba vartotojo. Yra registras MODE, kuris nusako procesoriaus darbo režimą.

Procesoriuje yra aukšto lygio kalbos procesorius, kuris betarpiškai apdoroja supervizoriaus režime komandas, o vartotojo režime vykdo užduoties programą.

### Procesoriaus registrai:

**PLR** – 4 baitų puslapių lentelės registras skirtas puslapiavimo mechanizmui

**R1** – 4 baitų bendro naudojimo registras skirtas aritmetiniams skaičiavimams ir jų manipuliacijoms.

**R2** – 4 baitų bendro naudojimo registras skirtas aritmetiniams skaičiavimams ir jų manipuliacijoms.

**IC** – 2 baitų virtualios mašinos programos skaitiklis. Pačioje pradžioje jis rodo adresu 00.

**C** – 1 baito požymių registras. (registro įgaunamos reikšmės aprašytos virtualios mašinos apraše)

**S** – 2 baitų semaforo registras, kurio kiekvienas bitas nurodo bendros atminties bloką(pirmojo bloko užimtumą parodo pirmasis bitas, antrojo – antrasis ir t.t.). Kai bito reikšmė lygi 0, bendrąja atmintimi, kurią nusako bitas gali naudotis procesas. Jei to bito reikšmė lygi 1, tuomet procesas negali gauti atitinkamo bloko valdymo ir turi laukti tol, kol semaforo reikiamo bito reikšmė tampa lygi 0.

**MODE** – registras, kurio reikšmė nusako procesoriaus darbo režimą.

S – Jei dirbama supervizoriaus rėžimu. Įjungiama jei kyla pertraukimas.

U – Jei dirbama vartotojo rėžimu. Kai vykdomas programos kodas.

**PI** – programinių pertraukimų registras.

1 – Jei suveikė atminties apsauga.

2 – Jei operacijos kodas neegzistuoja.

**SI** – supervizorinių pertraukimų registras.

1 – Jei kviečiama DG

2 – Jei kviečiama DP

3 – Jei kviečiamos papildomo įrenginio komandos.

4 – Jei kviečiamos komandos dirbančios su bendrąja atminties sritimi

5 - HALT

**CH[i]** – registrai, kuriuose nusakomas kanalų užimtumas.

1 – jei i-tasis kanalas nelaisvas.

0 – jei i-tasis kanalas laisvas.

**TIMER** – taimerio registras. Šis registras yra atsakingas už tai, kad nesusidarytų ciklai tarp virtualių mašinų. Pradedant virtualios mašinos užduoties vykdymą TI registro reikšmė bus nustatoma tam tikrai reikšmei (N). Įvykdžius eilinę instrukciją TI reikšmė bus mažinama priklausomai nuo to per kiek ši instrukcija yra atliekama. Kuomet TI reikšmė yra lygi nuliui įvyksta pertraukimas.

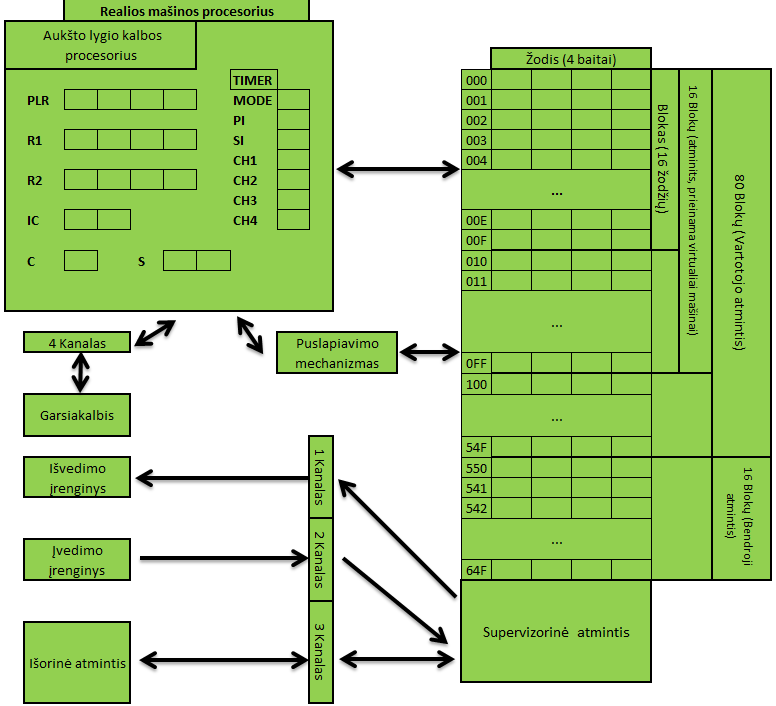
### Operatyvioji atmintis:

Realioje mašinoje yra atmintis suskirstyta į supervizorinę ir vartotojo atmintis. Vartotojo atmintis užima 64 blokai po 16 žodžių, iš kurių 16 blokų yra bendros atminties. Bendra atmintis nustatoma adresu 550+ (10h \* x1 + x2), kur x1, x2 yra adresai(šešioliktaine sistema).

### Išorinė atmintis:

Kietasis diskas.

## Realios mašinos techninės įrangos komponentų išsidėstymo vienas kito atžvilgiu ir tarpusavio sąveikos schema.

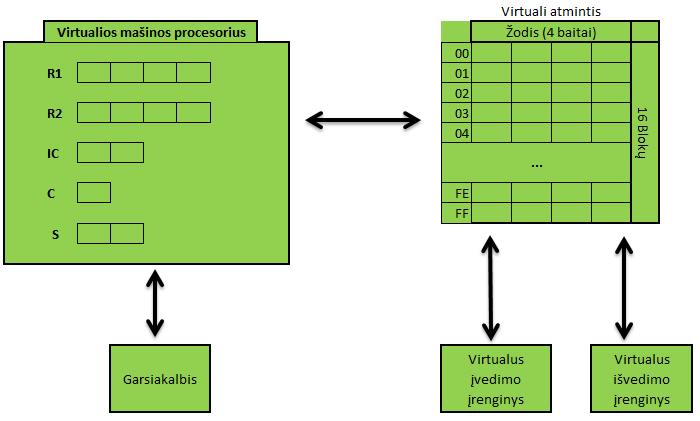


# Virtualios mašinos aprašas

## Modeliuojamos virtualios mašinos sampratos aprašymas

Virtuali mašina yra paprastesnis realios mašinos modelis, kurios operacijos yra apdorojamos MOS. Virtuali mašina tampa tarsi tarpininku tarp konkrečios mašinos ir jai taikomos programinės įrangos.

## Modeliuojamos virtualios mašinos loginių komponentų aprašymas



### Virtualios mašinos atmintis

Vartotojo atmintis skirta virtualių mašinų atmintims bei puslapių lentelėms laikyti. Virtualiai mašinai skiriama 16 blokų po 16 žodžių. Žodis 4 baitų dydžio.

Tuose blokuose turi tilpti užduoties programa. Kiekvienas virtualios atminties blokas turi virtualų ir realų adresą. Virtualius adresus naudoja virtuali mašina, realius – reali mašina. Ryšiai tarp virtualaus ir realaus adreso nusakomi puslapių lentelės registru, kuris yra 4 baitų dydžio. Jo baitus pažymėsime a0 a1 a2 a3. a0 baite nurodomas vartotojo programos dydis žodžiais, a1 - maksimalus puslapių skaičius, a2 a3 nusako ryšį tarp realios ir virtualios atminties.

Ryšys nusakomas tokiu būdu: Realus adresas = 10h \* [10h \* (10h \* a2 + a3) + x1] + x2. Virtualios mašinos registrų reikšmės sutampa su realios mašinos atitinkamų registrų reikšmemis

### Virtualios mašinos procesorius

Kiekviena virtuali mašina turi virtualų procesorių. Virtualaus procesoriaus paskirtis - vykdyti programą, kuri yra virtualioje atmintyje.

Virtualus procesorius turi šiuos registrus:

Duomenų registrai:

**R1** – 4 baitų bendro naudojimo registras;

**R2** – 4 baitų bendro naudojimo registras;

Nuorodų registras:

**IC** – 2 baitų virtualios mašinos programos skaitiklis;

Loginis registras:

**C** – 1 baito požymio registras.

Formuojamas komandų C1 ir C2 ir visų aritmetinių komandų bei loginių komandų (jos neformuoja OF). Kitos komandos reikšmės nekeičia.

Pirmasis bitas: zero flag (ZF):

ZF = 1, jei atliekant CMP operaciją registro ir reikšmės, esančios adresu xx skirtumas lygus 0. Arba jei atlikus atirmetinę operaciją, jos rezultatas lygus nuliui.

ZF = 0 – priešingu atveju.

Antrasis bitas: sign flag (SF)

SF = 1, jei atlikus aritmetinę operaciją su registro ir atminties xx reikšmėmis ženko bitas yra 1.

SF = 0 – priešingu atveju.

Trečiasis bitas: overflow flag (OF)

OF = 0, jei registro ir reikšmės, esančios adresu xx ženklo bitai yra vienodi, o jų skirtumo ženklo bitas skiriasi.

OF = 1 – priešingu atveju.

### Virtualios mašinos komandų sistema

Atmintyje esantis 4 baitų žodis gali būti traktuojamas kaip duomenys arba komanda. Pirmieji du arba trys baitai (vyresnieji) laikomi operacijų kodu, likę naudojami kaip operandai, paprastai tai bus atminties adresas.

Šiame komandų aprašyme 2 baitų adresas bus žymimas– „xx“, puslapio numeris „x“. Atmintyje skaičiai saugomi sveikaisiais skaičiais su ženklu, šešioliktainėje sistemoje.

#### Aritmetinės komandos:

**A1 xx** - Prie registro R1 reikšmės prideda adreso xx reikšmę.

R1 = R1 + [xx];

**A2 xx** - Prie registro R2 reikšmės prideda adreso xx reikšmę.

R2 = R2 + [xx];

**B1 xx** - Iš registro R1 reikšmės atimama adreso xx reikšmė.

R1 = R1 - [xx];

**B2 xx** - Iš registro R1 reikšmės atimama adreso xx reikšmė.

R2 = R2 - [xx];

**MU xx** - Registro R1 reikšmė sudauginama su adreso xx reikšme.

R1 = R1 \* [xx];

**DI xx** - Registro R1 reikšmė padalinama iš adreso xx reikšmės. Liekana išsaugoma R2 registre.

R2 = R1 mod [xx]; R1 = R1 div [xx];

#### Logikos operacijų bitinės komandos:

**XR xx** - Su registro R1 bitais ir adreso xx reikšmės bitais atliekama loginė “dalyba moduliu 2” (xor) operacija. Rezultatas išsaugomas R1 registre.

R1 = R1 xor [xx];

**AN xx** - Su registro R1 bitais ir adreso xx reikšmės bitais atliekama konjunkcija. Rezultatas išsaugomas R1 registre.

R1 = R1 & [xx];

**OR xx** - Su registro R1 bitais ir adreso xx reikšmės bitais atliekama disjunkcija. Rezultatas išsaugomas R1 registre.

R1 = R1 | [xx];

#### Palyginimo komandos:

**C1 xx** - Su registro R1 reikšme palyginama adreso xx reikšmė. Nustatoma atitinkama požymių registro reikšmė.

**C2 xx** - Su registro R2 reikšme palyginama adreso xx reikšmė. Nustatoma atitinkama požymių registro reikšmė.

#### Darbo su duomenimis komandos:

Jeigu bandoma rašyti arba skaityti iš neleisto naudoti bendrosios atminties bloko, tada kyla pertraukimas. PI = 1.

**L1 xx** - Į registrą R1 užkrauna adreso xx reikšmę.

R1 = [xx];

**L2 xx** - Į registrą R2 užkrauna adreso xx reikšmę.

R2 = [xx];

**S1 xx** - Registro R1 reikšmę išsaugo adresu xx.

[xx] = R1;

**S2 xx** Registro R2 reikšmę išsaugo adresu xx.

[xx] = R2;

**LCK x** - Programa prašo leisti naudoti bendrosios atminties srities x-ąjį bloką. Gavus leidimą, nustatoma x-ojo S registro bito reikšmė lygi 1.

**X1 xx** - Į registrą R1 užkrauna bendrosios atminties srities adreso xx reikšmę.

R1 = [xx];

**X2 xx** - Į registrą R2 užkrauna bendrosios atminties srities adreso xx reikšmę.

R2 = [xx];

**Z1 xx** - Registro R1 reikšmę išsaugo bendrosios atminties srities adresu xx.

[xx] = R1;

**Z2 xx** - Registro R2 reikšmę išsaugo bendrosios atminties srities adresu xx.

[xx] = R2;

**ULC x** - Programa atlaisvina bendrosios atminties srities x-ąjį bloką. Nustatoma x-ojo S registro bito reikšmė lygi 0.

#### Valdymo perdavimo komandos:

**JP xx** – (Jump) Valdymas perduodamas adresui xx.

IC = xx;

**JE xx** – (Jump if equal) Valdymas perduodamas adresui xx, jei ZF = 1.

IC = xx;

**JG xx** – (Jump if greater) Valdymas perduodamas adresui xx, jei ZF = 0 ir SF = OF.

IC = xx;

**JL xx** – (Jump if less) Valdymas perduodamas adresui xx, jei SF <> OF.

IC = xx;

**JX xx** - Valdymas perduodamas adresui, kuris yra xx reikšmė.

IC = [xx];

**LO xx** - Ciklinė valdymo perdavimo komanda. R2 reikšmė sumažinama vienetu. Jei R2 reikšmė didesnė nei 0 perduodamas valdymas adresu xx.

R2 = R2 - 1; IF R2 > 0 THEN IC = xx;

#### Įvedimo/Išvedimo komandos:

**DGT x** – Pertraukimų registras SI nustatomas SI = 1 ir valdymas perduodamas OS, duomenų kopijavimui iš įvedimo įrenginio į puslapį x.

**DPT x** – Pertraukimų registras SI nustatomas SI = 2 ir valdymas perduodamas OS, duomenų kopijavimui į išvedimo įrenginį iš puslapio x.

#### Papildomo įrenginio valdymo komandos:

Šios komandos nustato pertraukimų registrą SI = 3 ir perduoda valdymą OS, komandų apdorojimui.

**GGR1** - Nustatyti garsiakalbio grojimo trukmę iš registro R1.

**GGR2** - Nustatyti garsiakalbio grojimo trukmę iš registro R2.

**GNR1** - Garsiakalbiu groti natą kurios dažnis saugomas registre R1

**GNR2** - Garsiakalbiu groti natą kurios dažnis saugomas registre R2.

#### Programos pabaigos komanda:

**HALT** - Baigti programos darbą. Registras SI nustatomas SI = 4.

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

Virtuali mašina, aptikusi įvedimo komandą apdoroja jos operandą kaip adresą ir atitinkamai iš įvedimo buferio kopijuoja visą puslapį (16 žodžių) į nurodytą atminties vietą. Buferis tampa paruoštas kopijavimui kai yra pilnai užpildytas arba vartotojas nuspaudžia enter mygtuką, tuomet likusi buferio dalis užpildoma simboliais, kurių ASCII kodas lygus nuliui. Išvedimo komandos atveju pagal nurodytą adresą visas puslapis kopijuojamas į išvedimo įrenginį. Ekrane informacija atvaizduojama simbolių eilute ASCII formatu, kuri yra arba visas puslapis arba iki pirmojo simbolio, kurio ASCII kodas lygus nuliui.

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

Programos paleidžiamos interaktyviai, surenkant jų failų pavadinimus kaip komandas. Programos saugomos realios mašinos kietajame diske, į kurį jos patalpinamos „išorinėmis“, modelio, o ne projektuojamos OS, priemonėmis.

Programos failas turi būti tokios struktūros:

**@Memory x**

**@Code**

**....**

**.**

**.**

**....**

**@CodeEnd**

Čia x – programai skiriamas atminties blokų skaičius šešioliktainiu formatu.

Programos kodas rašomas tarp @Code ir @CodeEnd. Vienoje eilutėje gali būti tik viena komanda.

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

Virtualios mašinos loginiai komponentai tiesiogiai susiję su realios mašinos techniniais įrangos komponentais. Virtualiajame mašinos procesoriuje esantys registrai tiesiogiai atvaizduoja realios mašinos registrus.

# Virtuali mašina visos operacinės sistemos kontekste

Virtuali mašina visos operacinės sistemos kontekste įsivaizduojama kaip operacinės sistemos servisas/programa, kuri turi savo atmintį ir yra vykdoma lygiagrečiai su kitomis virtualiomis mašinomis.

# Naudota literatūra

[http://uosis.mif.vu.lt/~evaldasc/os/Siaulys%20Magistrinis.pdf](http://uosis.mif.vu.lt/%7Eevaldasc/os/Siaulys%20Magistrinis.pdf)

<http://wiki.mifsa.lt/Operacin%C4%97s_sistemos>

[http://uosis.mif.vu.lt/~julius/2010Rud/KompArch/Planas.html](http://uosis.mif.vu.lt/%7Ejulius/2010Rud/KompArch/Planas.html)