

VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

|  |
| --- |
| OS Projektas |
| Reali ir virtuali mašinos |

Ignas Mikaliūans,

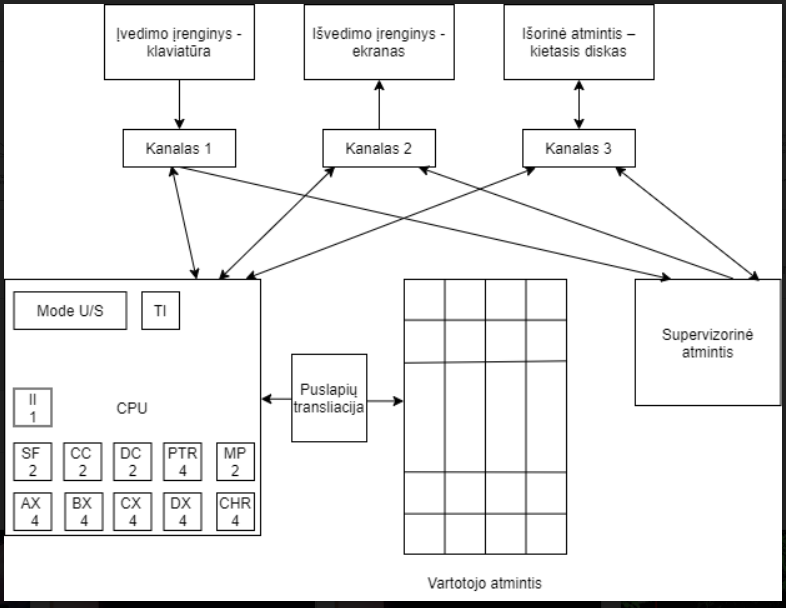
Juozapas Vytautas Norvaišas,

VU, MIF, 3k., INFO, 3 grupė.

**Užduoties sąlyga (A - 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),
  + 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
  + Supervizoriaus.
* Virtualios mašinos atmintis atvaizduojama į vartotojo atmintį naudojant puslapių transliaciją.
* Yra taimeris, kas tam tikrą laiko intervalą generuojantis pertraukimus.
* Input, Output
  + 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.

**Realios mašinos aprašas**

****

**Reali mašina** — tai kompiuteris, kuriam yra kuriama operacinė sistema. Realią mašiną sudaro centrinis procesorius, supervizoriaus, vartotojo ir išorinė atmintys, įvedimo ir išvedimo įrenginiai, atminties puslapiavimo mechanizmas. Atmintis bei įvedimo/išvedimo įrenginiai valdomi kanalais. Puslapiavimo mechanizmas išlaiko sąryšius tarp realių ir virtualių adresų. Puslapiavimo mechanizmas realizuotas keturių baitų registru procesoriuje.

**Centrinis procesorius —** atsakingas už komandų skaitymą iš atminties bei jų vykdymą. Procesorius gali dirbti dviem rėžimais: vartotojo ir supervizoriniu. Jei reali mašina dirba vartotojo rėžimu, tai ji faktiškai sutampa su virtualios mašinos dalimis. Vartotojo rėžime procesorius vykdo užduočių programas. Supervizoriaus rėžime komandos atliekamos aukšto lygio procesoriumi, kuris turi priėjimą prie virtualios mašinos atminties per puslapiavimo mechanizmą. Supervizoriaus rėžime procesorius vykdo komandas, kurios yra atsakingos už operacinės sistemos funkcionavimą, bet ne už vartotojo užduočių programas. Procesoriaus persijungia į supervizoriaus režimą pertraukimais arba sisteminiais kreipiniais.

**Centrinis procesorius turi:**

· **PTR** – puslapiavimo registras, naudojamas atvaizdavimui tarp virtualios ir realios mašinos. 4 baitų registras.

· **AX, BX, CX, DX** – 4 baitų registrai operacijų darbui.

· **CC** – 2 baitų virtualios mašinos komandų skaitliukas.

· **DC** – 2 baitų virtualios mašinos duomenų skaitliukas.

· **MODE** – registras, kurio reikšmė nusako procesoriaus darbo režimą (vartotojas ar supervizorius).

· **II** - pertraukimų registras.

II = 1, Komanda PRR

II = 2, Komanda PRS

II = 3, Komanda WGD

II = 4, Komanda RGD

II = 5, Komanda LGD

II = 6, Komanda UGD

II = 7, Komanda GET

EI = 1, Užimtas semiforas bendros atminties

EI = 2, Bloga komanda

· **SF** – Status Flag, rodo paskutinės aritmetinės/loginės operacijos rezultato požymius (ženklą), naudojamas ir procesoriaus darbinio rėžimo reguliavimui.

· **TI** – taimerio pertraukimo registras, nustato pertraukimą, kai pasiekiama tam tikra reikšmė. 2 baitų registras. Skirtas užduotims suderinti. TI = 0, Taimerio skaitliukas lygus nuliui

Yra sakoma, kad užduotis negali trukti ilgiau nei tam tikrą laiko momentą. Susitarsime, jog išvedimo / įvedimo operacijos reikalauja 3 laiko momentų, o kitos 1 ar 2. Taigi kai VM pradeda darbą, speciali supervizorinės atminties ląstelė TI yra nustatoma tam tikrai reikšmei. Įvykdžius instrukcija, TI reikšmė yra mažinama priklausomai nuo to kiek laiko momentų reikia instrukcijai. Kai TI tampa lygus nuliui, mikrokomanda aptinka taimerio pertraukimą.

· **CHR** – 4 baitų kanalų valdymo registras.

· **MP** – 2 baitų užpildytos bendros atminties registras semaforas

**Pertraukimų mechanizmas —** įgalina procesorių vykdyti keletą programų ir įvedimo – išvedimo operacijų vienu metu. Procesorius aptarnauja kiekviena operacija, kai atsiranda poreikis. Kiekvienas pertraukimas turi savo identifikaciją (sistema turi turėti galimybę atskirti pertraukimų tipus). Tik sistemai aptikus pertraukimus, yra nutraukiamas vartotojo programos vykdymas. Tai atliekama pasinaudojant pertraukimų vektorių lentele (kur yra nuorodos į pertraukimus apdorojančias programas). Valdymas yra perduodamas pertraukimą apdorosiančiai programai. Įvykdžiusi savo darbą, pertraukimo apdorojimo programa grąžina valdymą operacinei sistemai pertraukimo vietoje. Ką daryti toliau sprendžia OS.

**Realios mašinos atmintis.** Vartotojo ir supervizorinė atmintys dalijasi realios mašinos atmintį.

**Supervizorinė atmintis.** Apibrėžiama tokio dydžio, kokio reikia. Joje yra operacinės sistemos programos komandos, vykdomos supervizoriaus rėžime, semaforai – apsaugantys bendrą atmintį, ir programos interpretuojančios virtualios mašinios komandas. Supervizoriaus rėžime laikas neskaičiuojamas.Supervizorinė atmintis realiai nebus realizuota.

**Vartotojo atmintis** – susideda iš 2048 keturių baitų ilgio žodžių, nuosekliai adresuojamų nuo 000h iki 7FFh. Atmintis suskirstyta į 128 blokus po 16 žodžių. 96 blokai adresuojamos atminties skiriami VM. 4 blokai – puslapių lentelėms. Paskutiniai keturi atminties blokai nuo 7BFh iki 7FFh yra skiriami bendrajai atminčiai, kurią gali naudotis visos VM.

**Išorinė atmintis.** Išorinė atmintis bus realizuota failu kietajame diske. Išorinės atminties dydis nebus ribojamas, bet praktiškai priklausys nuo kietojo disko talpos.

**Įvedimo ir išvedimo įrenginiai.** Įvedimo/išvedimo įrenginiai atitinkamai yra naudojami nuskaityti vartotojo įvestas komandas bei išvesti rezultatą. Įvedimui naudojama klaviatūra, išvedimui - ekranas.

**Duomenų perdavimo kanalai** – specialus registras, valdantys įvedimo-išvedimo ir išorinės atminties įrenginių darbą. Jie reikalingi tam, kad nuo palyginti lėtų skaitymo ir rašymo operacijų būtų išlaisvintas centrinis procesorius. Kanalai ir centrinis procesorius dirba lygiagrečiai (aparatūrinis paralelizmas). Kanalai duomenis tvarko aukštesniame loginiame lygyje, nei įvedimo-išvedimo įrenginiai. Skirtingai nuo pastarųjų jie gali operuoti duomenų blokų įvairaus ilgio masyvais. Pirmasis kanalas priima duomenis iš vartotojo ir perduoda juos į supervizorinę atmintį. Panašiai dirba ir antrasis kanalas, tvarkantis duomenų srautą, einantį iš supervizoriaus atminties į ekraną. Trečiasis kanalas atsakingas už apsikeitimą duomenimis tarp supervizorinės atminties ir išorinės atminties įrenginio – kietojo disko. Duomenys gali judėti tiek viena (rašymas), tiek kita (skaitymas) kryptimi. Kanalų ir centrinio procesoriaus darbo sinchronizacijai ir yra skirti įvedimo-išvedimo pertraukimai. Kanalas, baigęs darbą, informuoja apie tai centrinį procesorių, atitinkamai pakeisdamas pertraukimo registro II reikšmę.

**Atminties puslapiavimo mechanizmas.** Virtualios mašinos blokai vartotojo atmintyje gali būti „išmėtomi“ bet kokia tvarka, taigi reikiamu momentu būtų sunku nustatyti, kur realioje atmintyje rasti virtualios mašinos atmintį. Šiai problemai spręsti naudojamas puslapiavimo mechanizmas. Pradėjus darbą virtualiai mašinai yra padaroma puslapių lentelė, kurioje nurodama, kuris realios atminties blokas atitinka virtualios atminties bloką. Puslapių lentelei yra skiriamas vienas realios atminties blokas. Puslapio lenteles žodžio pozicija rodo virtualios mašinos bloko numerį, o jame esanti reikšmė, to virtualios mašinos bloko adresą vartotojo atmintyje. PTR registre yra saugomas einamosios virtualios mašinos puslapiavimo lentelės vartotojo atmintyje adresas. Taip puslapiavimo mechanizmas, naudodamasis PTR registro reikšme išsprendžia šią problemą (susieja virtualią atmintį su realia).

**Veikimo principas:** PTR yra 4 baitų ir simboliškai žymėsime taip a**0** a**1** a**2** a**3** . O x**1** x**2** virtualius adresas.

· a**0** – nenaudojamas.

· a**1** – nenaudojamas.

· 16\*a**2** +a**3** – puslapių lentelės bloko numeris vartotojo atmintyje.

· 16\*(16\*a**2** +a**3** ) – puslapių lentelės bloko adresas.

· 16\*(16\*a**2** +a**3** ) + x**1** – bloko x**1** adresas puslapių lentelėje. Jame saugomas bloko numeris į kurį atvaizduotas yra x1 blokas VM.

· 16\*[16\*(16\*a**2** +a**3** ) + x**1** ] – VM bloko x**1** realus bloko adresas.

· 16\*[16\*(16\*a**2** +a**3** ) + x**1** ] + x**2** – realus adresas atitinkantis virtualų adresą x**1** x**2**.

**Taimerio mechanizmas.** Šis mechanizmas atsakingas už geresnį užduočių išlygiagretinimą. Yra laikoma kad ta pati užduotis negali būti vykdoma ilgiau nei N laiko momentų. Laikysim kad įvedimo/išvedimo instrukcijos atliekamos per 3 taktus, visos kitos per 1 taktą. Dabar apie veikimo principą. Pradedant virtualios mašinos užduoties vykdymą TI registro reikšmė nustatoma tam tikrai reikšmei. Tarkim N = 10. Įvykdžius eilinę instrukciją TI reikšmė mažinama priklausomai nuo to per kiek taktų ši instrukcija yra atliekama. Kuomet TI reikšmė yra lygi nuliui, mikrokomanda aptinka taimerio pertraukimą.

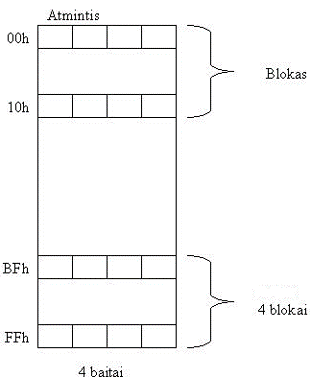
**Semaforas** S tai sveikas neneigiamas skaičius, su kuriuo atliekamos operacijos P(S) ir V(S), kur P ir V nauji primityvai. Operacijos pasižymi savybėmis:

P(S), V(S) – nedalomos operacijos, t.y. jų valdymo negalima pertraukti ir jų vykdymo metu negalima kreiptis į semaforą S;

* V(S): S:S+1; (didinama semaforo reikšmė)
* P(S): S:S-1; (sumažinama jei S>0)
* Jei S=0, tai procesas P, kuris vykdo operaciją P(S), laukia, kol sumažinimas vienetu bus galimas. Šiuo atvėju P(S) yra pertraukiamas.
* Jei keletas procesų vienu metu iškviečia V(S) ir/ar P(S) su vienu semaforu, tai užklausimai vykdomi nuosekliai, kokia nors iš anksto nežinoma tvarka.
* Jei keletas procesų laukia operacijos P(S) įvykdymo, S – ta pats, tai reikšmei tapus teigiamai (kai kažkuris procesas įvykdė operaciją V(S)), kažkuris iš laukiančių procesų bus pradėtas vykdyti.

Pagal prasmę operacija P atitinka perėjimo išškvietimą, o V – kito proceso aktyvaciją.

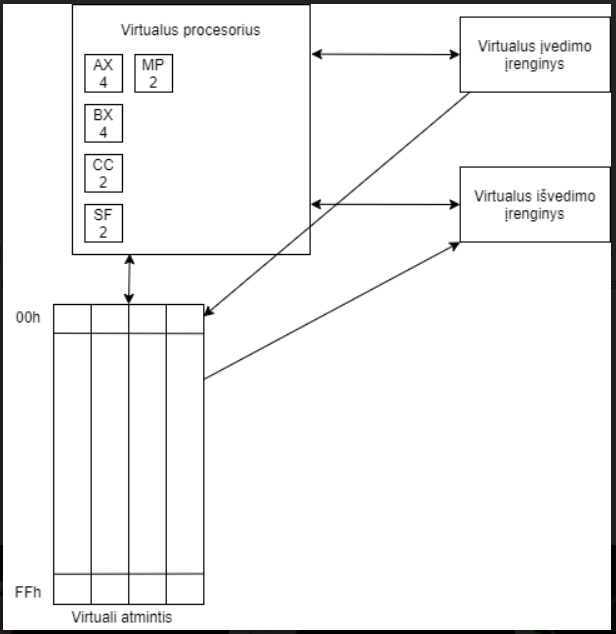
**Virtualios mašinos aprašas**

Virtuali mašina – tai tarsi virtuali realios mašinos kopija. Virtuali reiškia netikra. Mes surenkame reikalingas realios mašinos komponentes, tokias kaip procesorių, atmintį, įvedimo/išvedimo įrenginius ir suteikiame jiems paprastesnę vartotojo sąsają.

Tarkime, turime 100h žodžius atminties. Kiekvienas žodis susideda iš 4 baitų. Tuomet žodžiai bus adresuojami nuo 00h iki FFh. Virtualios mašinos atmintis suskirstoma į 16 blokų po 16 žodžių.

**Virtualios mašinos modelis**

Virtualios mašinos modelis panašus į realios mašinos modelį, tačiau virtualios mašinos komponentams yra suteikiama paprastesnė nei reali vartotojo sąsaja.



**Procesoriaus komandos**

Atmintyje esantis 4 baitų žodis gali būti traktuojamas kaip duomenys arba komanda. Pirmieji du baitai (vyresnieji) laikomi operacijų kodu, likę naudojami kaip operandai, paprastai tai bus atminties adresas. Mikroprogramos, interpretuojančios virtualaus procesoriaus komandas bus vykdomos HLP. Pačios mikroprogramos turi būti tik skaitymui pažymėtoje supervizorinės atminties dalyje.

**Virtualios mašinos procesoriaus komandos:**

**1) Aritmetinės komandos:**

**ADD –** sudėti pasirinktą darbinį registrą ir atminties ląstelę ar kitą darbinį registrą. Reikšmė priskiriama pirmajam darbiniam registrui.

**ADD x1 x2 :**

**x1 := x1 + x2**

**SUB –** atima iš darbinio registro atminties ląstelę ar kitą darbinį registrą. Reikšmė priskiriama pirmajam darbiniui registrui.

**SUB x1 x2 :**

**x1 := x1 - x2**

**2) Palyginimo komandos:**

**MOR–** Dviejų registrų palyginimas ( jei x1 daugiau už x2 ).

**MOR x1 x2 :**

**If x1 > x2 then SF[0] := 1 else SF[0] : = 0**

**EQL –** Dviejų registrų palyginimas (jei R lygus D)

**EQL x1 x2 :**

**If x1 == x2 then SF[0] := 1 else SF[0] : = 0**

**3) Darbui su atmintimi**

**MOV –** Kopijuoja duomenis esančius pirmame registre/atminties ląstelėje į antra registrą/ atminties ląstelę.

**MOV x1 x2 :**

**x2 := x1**

**LEA –** Į darbinį registrą įkrauna kintamojo efektyvųjį adresą.

**LEA x1 x2 :**

**x1 := [x2]**

**4) Įvedimo išvedimo komandos**

**GET -** Nuskaito pasirinktą kiekį įvesties duomenų(žodžiais) iš vartotojo į atminties vietą ir gražina adresą į pasirinktą darbinį registrą, iššaukiant interuptą: 7 kuris perduoda valdymą OS.

**GET x1 x2 :**

**Read (x1, [ b ], b = 16 \* x2)**

**PRR -** Išspausdina pasirinktą registro reikšmę, iššaukiant interuptą: 1 kuris perduoda valdymą OS.

**PRR x1 :**

**Print ( [ x1 ])**

**PRS -** Išspausdina pasirinktą atminties bloką, iššaukiant interuptą: 2 kuris perduoda valdymą OS.

**PRS x1 , x2 :**

**Print ( [ x1 , x2])**

**5) Darbo su bendra atminties sritimi komandos**

**WGD -** Į bendrą atminty nurodytu adresu įrašoma pasirinkto darbinio registro reikšmė, iššaukiant interuptą: 3 kuris perduoda valdymą OS.

**WGD x1x2**

**[ x2 ] := x1**

**RGD -** Iš bendros atminties nurodytu adresu nuskaitomas žodis į pasirinktą darbinį registrą, iššaukiant interuptą: 4 kuris perduoda valdymą OS.

**RGD x1x2**

**x1 := [ x2 ]**

**6) Semaforų komandos**

**LGD -** Užblokuojama pasirinkta bendros atminties ląstelė. Keičiamas MP registro pasirinktas bitas, iššaukiant interuptą: 5 kuris perduoda valdymą OS. **LGD x1 :**

**MP[x1] := 1**

**UGD -** Atblokuojama pasirinkta bendros atminties ląstelė. Keičiamas MP registro pasirinktas bitas, iššaukiant interuptą: 6 kuris perduoda valdymą OS.

**UGD x1 :**

**MP[x1] := 0**

**7) Sąlyginiai ir besalygynis valdymo perdavimai**

**JMP -** Besąlyginio valdymo perdavimo komanda. Valdymas perduodamas kodo sričiai nurodytu adresu. Ilgis 2

**JMP x1 :**

**CC = x1**

**JEZ -** Sąlyginio valdymo perdavimo komanda. Valdymas perduodamas kodo sričiai nurodytu adresu jei pasirinkto registro reikšmė lygi nuliui.

**JMP x1 x2 :**

**If x2 == 0 then CC = x1**

**JNZ -** Sąlyginio valdymo perdavimo komanda. Valdymas perduodamas kodo sričiai nurodytu adresu jei pasirinkto registro reikšmė nėra lygi nuliui.

**JMP x1 x2 :**

**If x2 != 0 then CC = x1**

**JGZ -** Sąlyginio valdymo perdavimo komanda. Valdymas perduodamas kodo sričiai nurodytu adresu jei pasirinkto registro reikšmė didesnė nuliui.

**JGP x1 x2 :**

**If x2 > 0 then CC = x1**

**JLZ -** Sąlyginio valdymo perdavimo komanda. Valdymas perduodamas kodo sričiai nurodytu adresu jei pasirinkto registro reikšmė mažesnė nuliui.

**JMP x1 x2 :**

**If x2 < 0 then CC = x1**

**8) Besąlyginio programos sustabdymo komanda**

**END –** programos vykdymo pabaiga, programos sustojimo taško komanda.

|  |  |  |  |
| --- | --- | --- | --- |
| M | O | V | (NULL) |
| A | X | 1 | 2 |
| A | D | D | (NULL) |
| A | X | 3 | 7 |
| S | U | B | (NULL) |
| A | X | 1 | 5 |
| P | R | R | (NULL) |
| A | X | (NULL) | (NULL) |

12 + 37 - 15 ?= 34

MOV AX, 12

ADD AX, 37

SUB AX, 15

PRR AX