VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ KATEDRA

Operacinės Sistemos pirmasis laboratorinis darbas

Virtuali mašinos „VMjCubeOS“ ir

Realios mašinos „jCubeOS“ projektas

Atliko: PS 3 kurso, 3 grupės studentas:

Jokūbas Rusakevičius (parašas)

Vilnius  
2018

# Anotacija

Darbas yra atliktas Vilniaus Universiteto Matematikos ir Informatikos fakulteto Programų sistemų 3 kurso 3 grupės studento Jokūbo Rusakevičiaus [jokubas.rusakeviciu@mif.stud.vu.lt](mailto:jokubas.rusakeviciu@mif.stud.vu.lt) kaip Operacinių sistemų kurso laboratorinis darbas. Šiame darbe bus aprašoma reali ir virtuali mašinos, kalbama apie jų komponentus bei jų sąveiką.

# Turinys

[Įvadas 4](#_Toc514273932)

[Dokumento paskirtis 4](#_Toc514273933)

[Darbo pagrindas 4](#_Toc514273934)

[1. Realios mašinos „jCubeOS“ projektas 5](#_Toc514273935)

[1.1. Techninės įrangos komponentai 5](#_Toc514273936)

[1.1.1. Procesorius 5](#_Toc514273937)

[1.1.2. Realios mašinos atmintys 6](#_Toc514273938)

[1.1.3. Duomenų perdavimo kanalų įrenginys 7](#_Toc514273939)

[1.1.4. Įvedimo ir išvedimo įrenginiai 8](#_Toc514273940)

[1.2. Komponentų išsidėstymas 8](#_Toc514273941)

[2. Virtualios mašinos „VMjCubeOS“ aprašas 9](#_Toc514273942)

[2.1. Virtualios mašinos samprata 9](#_Toc514273943)

[2.2. Virtualios mašinos komponentų aprašymas 9](#_Toc514273944)

[2.2.1. Virtuali atmintis 9](#_Toc514273945)

[2.2.2. Virtualus procesorius 9](#_Toc514273946)

[2.2.3. Virtualios mašinos komandų sistema 10](#_Toc514273947)

[2.3. Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais mechanizmas 12](#_Toc514273948)

[2.4. Virtualios mašinos interpretuojamojo vykdomojo failo išeities teksto formatas 12](#_Toc514273949)

[2.5. Modeliuojamos virtualios mašinos loginių komponentų sąryšio su realios mašinos techninės įrangos komponentai aprašymas 13](#_Toc514273950)

[3. Virtuali mašina operacinės sistemos kontekste 13](#_Toc514273951)

# 

# 1. Realios mašinos „jCubeOS“ projektas

Šiame skyriuje bus aprašoma realios mašinos projektas ir jo komponentai.

## 1.1. Techninės įrangos komponentai

### 1.1.1. Procesorius

Procesoriauspaskirtis – skaityti komandą iš atminties ar registrų ir ją vykdyti. Procesorius gali dirbti dviem rėžimais: vartotojo ir supervizoriaus. Vartotojo rėžime vykdomos virtualios mašinos komandos, o supervizoriaus rėžime – operacinės sistemos. Laiko tarpas, per kurį procesorius atlieka vieną komandą laikomas vienu laiko vienetu. Procesorius turi šiuos registrus:

***IC*** 2 baitų virtualios mašinos programos skaitiklis. Skirtas nurodyti vykdomos virtualios mašinos komandos adresą atmintyje.

***R1*** 4 baitų bendrosios paskirties registras.

***R2***4 baitų bendrosios paskirties registras.

***PTR*** 4 baitų *a0a­1a2a3* puslapių lentelės registras, kur *ai* yra baitai. Baitas *a0* nurodo kiek blokų užima vartotojo programa, *a1* – maksimalus puslapių skaičius, *a2* \* 10h + *a3* žymi puslapių lentelės adresą.

***MODE*** 1 baito registras, kurio reikšmė nusako procesoriaus darbo režimą („N“ – naudotojo, „S“ - supervizoriaus).

***SF*** 1 baitų loginis požymių registras. Skirtas saugoti aritmetinių operacijų loginių reikšmių sekai. Lentelėje pateikta informacija apie logines reikšmes.

|  |  |  |  |
| --- | --- | --- | --- |
| Bitas | Trumpinys | Reikšmė | Paaiškinimas |
| 0 | CF | Pernešimo požymis | „1“ tada, kai aritmetinių operacijų rezultatas netelpa į žodį |
| 1 | ZF | Nulio požymis | Parodo ar paskutinės operacijos rezultatas yra nulis |
| 2 | SF | Ženklo požymis | Parodo, koks yra paskutinės operacijos rezultato ženklas („1“ – jei neigiamas) |

lentelė. SF registro loginės reikšmės

***PI*** 1 baito programinių pertraukimų registras. (Fiksuoja pertraukimus sukeltus aptikus neegzistuojančių operacijų kodus, ar „išėjus“ už numatytų adreso ribų, ar pan.)

|  |  |  |
| --- | --- | --- |
| Registro Reikšmė | Pertraukimo Reikšmė | Paaiškinimas |
| 1 | Neteisingas adresas | Adresas nėra šešioliktainio formato. |
| 2 | Neteisingas operacijos kodas | Komandų interpretatorius nerado atitikmens duotajai komandai. |
| 3 | Neteisingi argumentai aritmetinei komandai | Registrų ar kitų operandų, su kuriais atliekamos aritmetinės komandos reikšmės nėra skaitinės (ne šešioliktainiai skaičiai). |
| 4 | Neteisingas priskyrimas |  |
| 5 | Perpildymas (overflow) | Sunaudota visa sistemos atmintis. |
| 6 | Nepavyko atidaryti failo | Nurodytas neteisingas pavadinimas arba failas jau atidarytas rašymo režime. |
| 7 | Nepavyko nuskaityti iš failo | Nėra atidaryto failo. |
| 8 | Nepavyko įrašyti į failą | Nėra atidaryto failo. Failas atidarytas skaitymo režime. |
| 9 | Nepavyko uždaryti failo | Iš failo šiuo metu yra skaitoma arba į jį rašoma, arba failas jau yra uždarytas. |
| 10 | Nepavyko ištrinti failo | Failas tokiu pavadinimu neegzistuoja. Arba neturima privilegijų trinti tokio formato, tipo failų ar to failo vietoje esančių failų, arba failas yra šiuo metu naudojamas kitos programos. |
|  |  |  |
| 11 | Atidarytu failu limitas | Atmintyje yra saugomi atidarytų failų pavadinimai ir pan. Todėl yra limitas kiek failų galima vienu metu būti atidarius. |
| 12 | Netinkamas failo pavadinimas | Failo pavadinimas netelpa jam priskirtame bloke, yra per ilgas. |

***SI*** 1 baito supervizorinių pertraukimų registras. (Pertraukimas įvyksta kai virtuali mašina negali pati atlikti veiksmo (pavyzdžiui, darbas su įvedimu ir išvedimu) ir kreipiasi į operacinės sistemos servisą).

|  |  |  |
| --- | --- | --- |
| Registro Reikšmė | Pertraukimo Reikšmė | Paaiškinimas |
| 1 | Darbas su įvedimu (blokas) | Iššaukta duomenų bloko nuskaitymo iš įvedimo įrenginio komanda. |
| 2 | Darbas su įvedimu (žodis) | Iššaukta duomenų žodžio nuskaitymo iš įvedimo įrenginio komanda. |
| 3 | Darbas su išvedimu (blokas) | Iššaukta duomenų bloko įrašymo į išvedimo įrenginį komanda. |
| 4 | Darbas su išvedimu (žodis) | Iššaukta duomenų žodžio įrašymo į išvedimo įrenginį komanda. |
| 5 | Darbas su failais (Atidarymas) | Iššaukta Failo atidarymo komanda. |
| 6 | Darbas su failais (Uždarymas) | Iššaukta Failo uždarymo komanda. |
| 7 | Darbas su failais (Ištrynimas) | Iššaukta Failo ištrynimo komanda. |
| 8 | Darbas su failais (Bloko Rašymas) | Iššaukta į Failą bloko įvedimo komanda. |
| 9 | Darbas su failais (Pasirinktas Rašymas) | Iššaukta į Failą pasirinkto kiekio duomenų įvedimo komanda. |
| 10 | Darbas su failais (Bloko Skaitymas) | Iššaukta iš Failo bloko nuskaitymo komanda. |
| 11 | Darbas su failais (Pasirinktas Skaitymas) | Iššaukta iš Failo pasirinkto kiekio duomenų nuskaitymo komanda. |
| 12 | Darbo pabaiga | Iššaukta darbo pabaigos komanda HALT |

***TI*** 1 baito registras taimerio pertraukimas fiksuoti.

Taip pat procesoriuje yra taimeris – laiko matavimo įrenginys, kuris vienodais laiko intervalais generuoja pertraukimus.

Procesorius dirba su simboliniais duomenimis bei sveikaisiais skaičiais su ženklu, užrašytais simboliniu formatu.

### 1.1.2. Realios mašinos atmintys

Atmintis – įrenginys informacijai saugoti. „jCubeOS“ reali mašina turi trijų rūšių atmintis: supervizorinę, vartotojo ir išorinę. Vartotojo ir supervizorinės atmintys dalijasi realios mašinos atmintį, kurios dydis 128 blokai po 16 žodžių (iš viso 2048 žodžiai), kur žodžio ilgis yra 4 baitai, o baitas susideda iš simbolio arba vieno skaitmens šešioliktainio skaičiaus (0..F). Vartotojo atmintis bus atminties pradžioje. Visą likusią dalį užims supervizorinė atmintis.

#### 1.1.2.1. Vartotojo atmintis

Vartotojo atmintis skirta virtualių mašinų atmintims bei puslapių lentelėms laikyti. Vartotojo atminties dydis yra 96 blokai. Atmintis numeruojama nuo 0. Visi virtualios mašinos puslapiai yra užkraunami prieš pradedant virtualios mašinos darbą.

#### 1.1.2.2. Supervizorinė atmintis

Supervizorinės atminties dydis yra 32 blokai. Atmintis numeruojama nuo 0.

Paskutiniai 8 blokai yra rezervuoti failų tvarkyklei. Ten pateikiama informacija apie atidarytus failus: jų naudojimo būseną, jų pavadinimus bei atidarymo režimus („R“ – skaitymui, „W“ – rašymui). Vienas blokas yra skirtas vienam failui. Pirmas žodis žymi failo naudojimo statusą. Antras skaitymo rašymo režimą („R“ arba „W“) . Visi likę skirti atidaryto failo pavadinimui saugoti.

#### 1.1.2.3. Išorinė atmintis

Išorinė atmintis realizuota failu kietajame diske. Schematiškai išorinę atmintį galima pavaizduoti analogiškai atminčiai, tik ji nėra dalijama į kitas atmintis. Atminties dydis kietajame diske yra 128 blokai. Išorinės atminties dydis yra lengvai keičiamas.

Taip pat reali mašina gali sukurti, atidaryti skaitymui, atidaryti rašymui (visa faile esanti informacija yra ištrinama) ir ištrinti failus. Iš failo skaitoma ir į jį rašoma 1 bloko dydžio ilgio įrašais. (Vienas apsikeitimas trunka tris laiko vienetus.)

### 1.1.3. Duomenų perdavimo kanalų įrenginys

Kanalų įrenginys leidžia dirbti su atmintimis. Priklausomai nuo nustatytų registrų kanalų įrenginys gali vykdyti apsikeitimą duomenimis visomis galimomis kryptimis. Veiksmai su kanalų įrenginiu atliekami tik supervizoriaus režime.

Kanalų įrenginio registrai:

***SB*** – Bloko, iš kurio kopijuosime numeris.

***DB*** – Bloko, į kurį kopijuosime numeris.

***ST –*** Objekto, iš kurio kopijuosime, numeris

1. Vartotojo atmintis;
2. Supervizorinė atmintis;
3. Išorinė atmintis;
4. Įvedimo srautas;
5. Įvedimo srautas (žodis);
6. Registras *R1* (žodis);
7. Registras *R2* (žodis);
8. Failas.

***DT*** – Objekto, į kurį kopijuosime, numeris

1. Vartotojo atmintis;
2. Supervizorinė atmintis;
3. Išorinė atmintis;
4. Išvedimo srautas;
5. Išvedimo srautas(žodis);
6. Registras *R1* (žodis);
7. Registras *R2* (žodis);
8. Failas.

Žodinės komandos galimos tik tarp žodinių, o blokinės tik tarp blokų. (5, 6, 7 naudojamos tik tarpusavyje).

Kanalų įrenginys turi komandą ***XCHG***, tačiau neturi procesoriaus, kuris galėtų ją įvykdyti. Šią komandą vykdo centrinis realios mašinos procesorius.

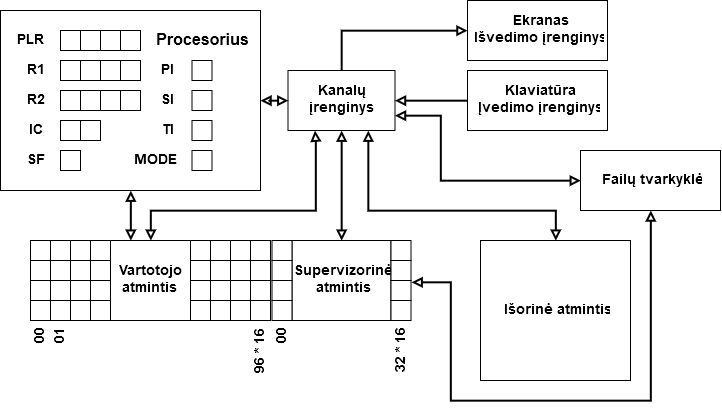
Procesas, norėdamas pasinaudoti kanalų įrenginiu, turi nustatyti kanalų įrenginio registrus ir tada įvykdyti komandą ***XCHG***.

### 1.1.4. Įvedimo ir išvedimo įrenginiai

Reali mašina „jCubeOS“ turi vieną nuoseklų įvedimo įrenginį – klaviatūrą ir vieną nuoseklų išvedimo įrenginį – ekraną. Tiek duomenų nuskaitymas, tiek jų įrašymas vyksta 1 bloko arba 1 žodžio dydžio įrašais. Taip pat mašina gali skaityti duomenis iš ir rašyti duomenis į atidarytus failus naudodama failų tvarkyklę.

## 1.2. Komponentų išsidėstymas

Šiame skyriuje yra pateikiama realios mašinos techninės įrangos komponentų išsidėstymo vienas kito atžvilgiu ir tarpusavio sąveikos schema.



pav. Realios mašinos tarpusavio sąveikos schema

# 2. Virtualios mašinos „VMjCubeOS“ aprašas

Šiame skyriuje bus aprašoma virtualios mašinos samprata, jos komponentai, bendravimo su įvedimo/išvedimo įrenginiais mechanizmas

## 2.1. Virtualios mašinos samprata

Virtuali mašina – tai tarsi realios mašinos kopija. Virtualioje mašinoje surenkami reikalingi komponentai, tokie kaip procesorius, atmintis, įvedimo/išvedimo įrenginiai ir suteikiama jiems paprastesnį vartotojo sąsaja. Tuo pačiu palengvinamas programavimo procesas – sudėtingas ar vartotojui nepatogios sąsajos virtualioje mašinoj yra aprašomos supaprastintai. Virtuali mašina realizuoja realios mašinos komandas paprastesniu, lengviau suprantamu būdu vykdant virtualios mašinos komandas kaip realios mašinos komandas ar jų rinkinius. Tai leidžia pasiekti realią mašiną bei mašininiu kodu parašytą virtualios mašinos programą sėkmingai įvykdyti realioje mašinoje.

## 2.2. Virtualios mašinos komponentų aprašymas

Šiame skyriuje aprašomi virtualios mašinos loginiai komponentai.

### 2.2.1. Virtuali atmintis

Atmintis virtualiai mašinai yra paskiriama jos inicijavimo metu, o jos dydis yra 16 blokų po 16 žodžių (256 žodžiai). Virtuali mašina dirba su 4 baitų ilgio žodžiais.

### 2.2.2. Virtualus procesorius

Virtualus procesorius yra paprastesnis nei realus procesorius ir turi tik pagrindinius registrus:

***IC*** 2 baitų virtualios mašinos programos skaitiklis. Skirtas nurodyti vykdomos komandos adresą virtualioje atmintyje.

***R1*** 4 baitų bendrosios paskirties registras.

***R2***4 baitų bendrosios paskirties registras.

***SF*** 1 baitų loginis požymių registras. Skirtas saugoti aritmetinių operacijų loginių reikšmių sekai. Lentelėje pateikta informacija apie logines reikšmes.

|  |  |  |  |
| --- | --- | --- | --- |
| Bitas | Trumpinys | Reikšmė | Paaiškinimas |
| 0 | CF | Pernešimo požymis | „1“ tada, kai aritmetinių operacijų rezultatas netelpa į žodį |
| 1 | ZF | Nulio požymis | Parodo ar paskutinės operacijos rezultatas yra nulis |
| 2 | SF | Ženklo požymis | Parodo, koks yra paskutinės operacijos rezultato ženklas („1“ – jei neigiamas) |

lentelė. SF registro loginės reikšmės

### 2.2.3. Virtualios mašinos komandų sistema

Visos komandos yra 4 baitų (vieno žodžio) ilgio. Toliau pateikiamos virtualios mašinos procesoriaus komandos su paaiškinimais.

#### 2.2.3.1. Duomenų persiuntimo komandos

|  |  |  |  |
| --- | --- | --- | --- |
| ***L*** | ***r*** | ***x*** | ***y*** |

Atminties ląstelės, kurios adresas *x* \* 10h + *y* (*x* ir *y* yra šešioliktainiai skaitmenys (0..F)), turinio kopijavimas į registrą *r* (jei *r* = 1, tai *R1* (*R1* := [*x* \* 10h + *y*]); *r* = 2, tai *R2* (*R2* := [*x* \* 10h + *y*])).

|  |  |  |  |
| --- | --- | --- | --- |
| ***S*** | ***r*** | ***x*** | ***y*** |

Registro *r* (jei *r* = 1, tai *R1; r* = 2, tai *R2*) turinio kopijavimas į atminties ląstelę, kurios adresas *x* \* 10h + *y* ([*x* \* 10h + *y*] := *R1*) arba ([*x* \* 10h + *y*] := *R2*).

#### 2.2.3.2. Aritmetinės komandos

|  |  |  |  |
| --- | --- | --- | --- |
| ***I*** | ***N*** | ***C*** | ***r*** |

Registro *r* (jei *r* = 1, tai *R1; r* = 2, tai *R2*) reikšmė padidinama vienetu (*r* := *r* + *1*). Formuoja *SF* požymius.

|  |  |  |  |
| --- | --- | --- | --- |
| ***D*** | ***E*** | ***C*** | ***r*** |

Registro *r* (jei *r* = 1, tai *R1; r* = 2, tai *R2*) reikšmė sumažinama vienetu (*r* := *r* - *1*). Formuoja *SF* požymius.

|  |  |  |  |
| --- | --- | --- | --- |
| ***A*** | ***D*** | ***D*** |  |

Prie registro *R1* reikšmės pridedama registro *R2* reikšmė (*R1* := *R1* + *R2*). Formuoja *SF* požymius.

|  |  |  |  |
| --- | --- | --- | --- |
| ***A*** | ***D*** | ***x*** | ***y*** |

Sudedamos registrų *R1* ir *R2* reikšmės bei rezultatas išsaugojamas atminties ląstelėje, kurios adresas *x* \* 10h + *y* ([*x* \* 10h + *y*] := *R1* + *R2*).

|  |  |  |  |
| --- | --- | --- | --- |
| ***S*** | ***U*** | ***B*** |  |

Iš registro *R1* reikšmės atimama registro *R2* reikšmė (*R1* := *R1* – *R2*). Formuoja *SF* požymius.

|  |  |  |  |
| --- | --- | --- | --- |
| ***S*** | ***B*** | ***x*** | ***y*** |

Iš registro *R1* atimama registro *R2* reikšmė ir rezultatas išsaugojamas atminties ląstelėje, kurios adresas *x* \* 10h + *y* ([*x* \* 10h + *y*] := *R1* – *R2*). Formuoja *SF* požymius.

|  |  |  |  |
| --- | --- | --- | --- |
| ***M*** | ***U*** | ***L*** |  |

Registro *R1* reikšmė padauginama iš registro *R2* reikšmės ir rezultatas įrašomas į *R1* registrą. (*R1* := *R1* \* *R2*). Formuoja *SF* požymius.

|  |  |  |  |
| --- | --- | --- | --- |
| ***M*** | ***L*** | ***x*** | ***y*** |

Registro *R1* reikšmė padauginama iš registro *R2* reikšmės ir rezultatas įrašomas į atminties ląstelę, kurios adresas *x* \* 10h + *y* ([*x* \* 10h + *y*] := *R1* \* *R2*). Formuoja *SF* požymius.

|  |  |  |  |
| --- | --- | --- | --- |
| ***D*** | ***I*** | ***V*** |  |

Registro *R1* reikšmė padalinama iš registro *R2* reikšmės ir rezultatas įrašomas į registrą *R1*, o liekana į registrą *R2*. (*R1* := *R1* / *R2*; *R2* := liek. (*R1* / *R2*)). Formuoja *SF* požymius.

|  |  |  |  |
| --- | --- | --- | --- |
| ***C*** | ***M*** | ***P*** |  |

Palygina registrus *R1* ir *R2*. Formuoja *SF* požymius (*R1* > *R2*, tai *ZF* = 0 ir *SF* = 0; *R1* = *R2*, tai *ZF* = 1; *R1* < *R2*, tai *ZF* = 0 ir *SF* = 1).

|  |  |  |  |
| --- | --- | --- | --- |
| ***C*** | ***r*** | ***x*** | ***y*** |

Palygina registrą *r* (jei *r* = 1, tai *R1*; *r* = 2, tai *R2*) ir atminties žodį adresu *x* \*10h + *y*. Formuoja *SF* požymius (*r* > *xy*, tai *ZF* = 0 ir *SF* = 0; *r* = *xy*, tai *ZF* = 1; *r* < *xy*, tai *ZF* = 0 ir *SF* = 1).

|  |  |  |  |
| --- | --- | --- | --- |
| ***X*** | ***O*** | ***R*** |  |

Atliekama registrų tarp *R1* ir *R2* reikšmių loginė operacija XOR ir rezultatas įrašomas į registrą *R1*. Formuoja *SF* požymius (*R1* = 0, tai *ZF* = 1; *R1* < 0, tai *ZF* = 0 ir *SF* = 1; *R1* > 0, tai *ZF* = 0 ir *SF* = 0).

|  |  |  |  |
| --- | --- | --- | --- |
| ***A*** | ***N*** | ***D*** |  |

Atliekama registrų tarp *R1* ir *R2* reikšmių loginė operacija AND ir rezultatas įrašomas į registrą *R1*. Formuoja *SF* požymius (*R1* = 0, tai *ZF* = 1; *R1* < 0, tai *ZF* = 0 ir *SF* = 1; *R1* > 0, tai *ZF* = 0 ir *SF* = 0).

|  |  |  |  |
| --- | --- | --- | --- |
| ***O*** | ***R*** |  |  |

Atliekama registrų tarp *R1* ir *R2* reikšmių loginė operacija OR ir rezultatas įrašomas į registrą *R1*. Formuoja *SF* požymius (*R1* = 0, tai *ZF* = 1; *R1* < 0, tai *ZF* = 0 ir *SF* = 1; *R1* > 0, tai *ZF* = 0 ir *SF* = 0).

|  |  |  |  |
| --- | --- | --- | --- |
| ***N*** | ***O*** | ***T*** |  |

Atliekama NOT operacija *R1* registro reikšmei. Formuoja *SF* požymius (*R1* = 0, tai *ZF* = 1; *R1* < 0, tai *ZF* = 0 ir *SF* = 1; *R1* > 0, tai *ZF* = 0 ir *SF* = 0).

#### 2.2.3.3. Valdymo perdavimo komandos

|  |  |  |  |
| --- | --- | --- | --- |
| ***G*** | ***O*** | ***x*** | ***y*** |

Besąlygiškas valdymo perdavimas kodo segmento komandai adresu *x* \* 10h + *y* (*IC* := *x* \* 10h + *y)*

|  |  |  |  |
| --- | --- | --- | --- |
| ***J*** | ***G*** | ***x*** | ***y*** |

Jei *ZF* = 0 ir *SF* = 0, tai valdymas perduodamas kodo segmento komandai adresu *x* \* 10h + *y.*

|  |  |  |  |
| --- | --- | --- | --- |
| ***J*** | ***L*** | ***x*** | ***y*** |

Jei *SF* = 1, tai valdymas perduodamas kodo segmento komandai adresu *x* \* 10h + *y.*

|  |  |  |  |
| --- | --- | --- | --- |
| ***J*** | ***C*** | ***x*** | ***y*** |

Jei *CF* = 1, tai valdymas perduodamas kodo segmento komandai adresu *x* \* 10h + *y.*

|  |  |  |  |
| --- | --- | --- | --- |
| ***J*** | ***Z*** | ***x*** | ***y*** |

Jei *ZF* = 1, tai valdymas perduodamas kodo segmento komandai adresu *x* \* 10h + *y.*

|  |  |  |  |
| --- | --- | --- | --- |
| ***J*** | ***N*** | ***x*** | ***y*** |

Jei *ZF* = 0, tai valdymas perduodamas kodo segmento komandai adresu *x* \* 10h + *y.*

#### 2.2.3.4. Įvedimo ir išvedimo komandos

|  |  |  |  |
| --- | --- | --- | --- |
| ***G*** | ***D*** | ***B*** | ***x*** |

Iš įvedimo srauto perskaito bloką (16) žodžių ir įrašo juos į ląsteles [*x* \*10h + *i* ], kur *i* = 0..F (0..15).

|  |  |  |  |
| --- | --- | --- | --- |
| ***G*** | ***D*** | ***r*** |  |

Iš įvedimo srauto perskaito vieną žodį ir įrašo jo reikšmę į registrą *r* (jei *r* = 1, tai *R1*; *r* = 2, tai *R2*).

|  |  |  |  |
| --- | --- | --- | --- |
| ***P*** | ***D*** | ***B*** | ***x*** |

Išsiunčia išvedimui bloką žodžių srautą iš atminties ląstelių [*x* \* 10h + *i* ], kur *i* = 0..F.

|  |  |  |  |
| --- | --- | --- | --- |
| ***P*** | ***D*** | ***r*** |  |

Išsiunčia išvedimui vieną žodį iš registro *r* (jei *r* = 1, tai *R1; r* = 2, tai *R2*).

#### 2.2.3.5. Darbo su failais komandos

|  |  |  |  |
| --- | --- | --- | --- |
| ***F*** | ***O*** | ***w*** | ***x*** |

Atidaro failą, kurio pavadinimas yra saugomas bloke adresu [*x* \* 10h + *i* ], kur *i* = 0..F, režimu *w* (jei *w* = *R*, tai skaitymo rėžimu; *w* = *W*, tai rašymo rėžimu). Atidaryto failo *id* išsaugo į *R1*.

|  |  |  |  |
| --- | --- | --- | --- |
| ***F*** | ***G*** | ***x*** | ***y*** |

Nuskaito bloką (16 žodžių) iš failo abstrakcijos, kurios *id* = *x* ir įrašo į atminties ląsteles adresu [*y* \* 10h + *i* ], kur *i* = 0..F.

|  |  |  |  |
| --- | --- | --- | --- |
| ***F*** | ***R*** | ***x*** | ***y*** |

Nuskaito *R2* registre esančia skaitinę reikšmę *r* žodžių iš failo abstrakcijos, kurios *id* yra nurodytas *R1* registre ir įrašo į atminties ląsteles adresu [*x* \* 10h + *y* + *i* ], kur *i* = 0..*r*.

|  |  |  |  |
| --- | --- | --- | --- |
| ***F*** | ***P*** | ***x*** | ***y*** |

Išsiunčia bloką iš atminties ląstelių [*y* \* 10h + *i* ], kur *i* = 0..F į failo abstrakciją, kurios *id* = *x*.

|  |  |  |  |
| --- | --- | --- | --- |
| ***F*** | ***W*** | ***x*** | ***y*** |

Išsiunčia *R2* registre esančią skaitinę reikšmę *r* žodžių iš atminties ląstelių [*x* \* 10h + *y +i*], kur *i* = 0..*r* į failo abstrakciją, kurios *id* yra nurodytas *R1* registre.

|  |  |  |  |
| --- | --- | --- | --- |
| ***F*** | ***C*** | ***x*** |  |

Uždaromas failas, kurio *id* = *x*.

|  |  |  |  |
| --- | --- | --- | --- |
| ***F*** | ***C*** | ***R*** |  |

Uždaromas failas, kurio *id* yra *R1*.

|  |  |  |  |
| --- | --- | --- | --- |
| ***F*** | ***D*** | ***x*** |  |

Pašalina failą, kurio pavadinimas saugomas bloke atmintyje adresu *x* \* 10h + *i*, kur *i* = 0..F.

#### 2.2.3.6. Programos pabaigos komandos

|  |  |  |  |
| --- | --- | --- | --- |
| ***H*** | ***A*** | ***L*** | ***T*** |

Vartotojo programos vykdymo pabaiga. Užbaigia programos darbą.

## 2.3. Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais mechanizmas

Virtualios mašinos įvedimo ir išvedimo įrenginiai yra pasiekiami naudojant *GDBx* ir *PDBx* arba *GDr* ir *PDr* komandas. Taip pat galima įvesti ar išvesti duomenis naudojant failus. Vienu metu gali būti atidarytas tiek failų kiek yra skirta atminties takelių supervizorinėje atmintyje failų tvarkyklei. Failai atidaromi su komanda *FOwx*, kur *w* atidarymo rėžimas („R“ skaitymas; „W“ rašymas), o *x* virtualios atminties bloko pirmo žodžio adresas. Skaitoma iš failo ir į faila naudojant *FGxy* ir *FPxy* arba *FRxy* ir *FWxy* komandas.

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

Vykdomojo failo bendras pavidalas susideda iš dviejų dalių:

* Kodo segmento;
* Duomenų segmento.

### 2.4.1. Kodo segmentas

Kodo segmento pradžia žymi eilutė „$CODE\n“ (\n – naujos eilutės simbolis). Visos komandos rašomos po šios žymės. Kiekviena komanda yra rašoma iš naujos eilutės. Visos komandos yra iki 4 baitų arba vieno žodžio ilgio. Komandos rašomos didžiosiomis raidėmis. Prieš komandą gali būti nurodyta žymė (žymės sintaksė: <žymė>: <komanda>), kuri po to gali būti naudojama vietoj betarpiško argumento (sintaksė: <komanda> $<žymė>. Kodo segmentas užbaigiamas prasidėjus duomenų segmentui.

### 2.4.2. Duomenų segmentas

Duomenų segmento pradžią žymi raktinis žodis „$DATA“. Toliau esančiose eilutėse pateikiamas duomenų segmento turinys. Naujos eilutės simboliai yra ignoruojami. Jei eilutė prasideda [<adresas>]: tai toliau esantys simboliai rašomi į duomenų segmentą nuo nurodyto <adresas> adreso. Duomenų segmentas užbaigiamas su raktiniu žodžiu „$END“, kuris žymi vykdomojo failo pabaigą.

### 2.4.3. Programos pavyzdys:

#### 2.4.3.1. Pavyzdys 1

* CODE
  + PDB4 - Išvedimo įrenginys išveda bloką duomenų pradedant adresu 4 \* 10h;
  + GD1 - Į *R1* registrą įrašoma reikšmė iš įvesties įrenginio;
  + PDB5 - Išvedimo įrenginys išveda bloką duomenų pradedant adresu 5 \* 10h;
  + GD2 - Į *R2* registrą įrašoma reikšmė iš įvesties įrenginio;
  + ADD - Sudedamos registrų reikšmės ir patalpinamos *R1;*
  + S168 - *R1* reikšmė patalpinama adresu 68;
  + PDB6 - Išvedimo įrenginys išveda bloką duomenų pradedant adresu 6 \* 10h;
  + HALT - Programa baigia darbą.
* $DATA
  + [40]:Įveskite pirmą skaičių:
  + [50]:Įveskite antrą skaičių:
  + [60]:Pirmo ir antro skaičių suma:
* $END

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

Virtuali mašina savo komandų vykdymui naudoja realios mašinos procesorių. Vienintelis skirtumas yra tai, kad jai neleidžiama pasiekti, kai kurių jai nereikalingų registrų. Taip pat virtuali mašina mato tik jai skirtą visos naudotojo atminties dalį bei gali pasiekti tik tuos failus, kurie yra jai paskirti.

# 3. Virtuali mašina operacinės sistemos kontekste

Virtuali mašina yra mašina skirta baigtinių naudotojo programų vykdymui. Ji yra supaprastintas, tuo pačiu ir apribotas realios mašinos modelis. Operacinės sistemos paskirtis yra pasirūpinti virtualių mašinų inicijavimu, resursų tarp jų skirstymu bei baigusių vykdyti programas mašinų sunaikinimu.