VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
INFORMATIKOS KATEDRA

Operacinių sistemų pirmoji užduotis

Virtualios ir realios mašinos projektas

Atliko: 3 kurso 1 grupės studentas
Dominykas Marma (parašas)

Darbo vadovas:
Mantas Grubliauskis (parašas)

Vilnius
2023

Turinys

1. Užduoties aparašymas

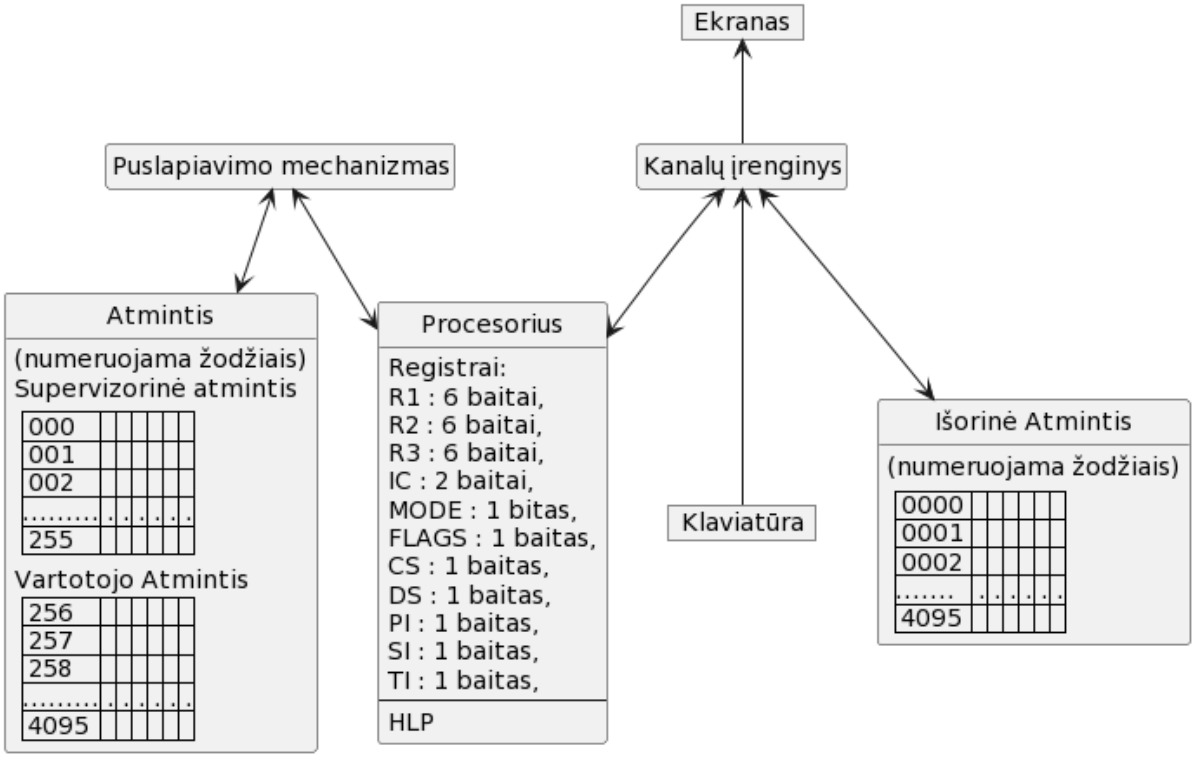
Virtualios mašinos procesoriaus komandos operuoja su duomenimis, esančiais registruose ir ar atmintyje. Yra komandos duomenų persiuntimui iš atminties į registrus ir atvirkščiai, aritmetinės (sudėties, atimties, daugybos, dalybos, palyginimo), sąlyginio ir besąlyginio valdymo perdavimo, įvedimo, išvedimo, 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 kurį jos patalpinamos „išorinėmis", modelio, o ne projektuojamos OS, priemonėmis.

2. Realios mašinos modelis

Reali mašina - kompiuterio modelis, turintis jam būdingas sudėtines dalis. Šiame darbe bus modeliuojama vieną procesorių turinti reali mašina. Jos modelis yra pateikiamas 1 paveiksliuke. Tolesnėse sekcijose bus plačiau aprašoma kiekviena šios mašinos dalis



1 pav. Realios mašinos modelis

2.1. Procesorius

Procesorius turi registrus:

- IC - komandų skaitiklis - 2 baitai.
- R1, R2, R3 - bendrosios paskirties registrai iš 6 baitų (vienas žodis).
- CS - kodo segmento registras, 2 baitai.
- DS - duomenų segmento registras, 2 baitai.
- MODE - 1 bito registras, kuris nurodo procesoriaus darbo režimą (0 - vartotojas, 1 - supervizorius)
- FLAGS - 1 baito požymių registras, kurio reikšmė yra pakeičiama po aritmetinės ar palyginimo operacijos. Jis susideda iš šių bitų (numeruojami nuo jauniausio pradedant nuo skaičiaus 1):

  - (1-asis bitas) CF - pernešimo požymis (carry flag).
  - (2-asis bitas) ZF - nulio požymis (zero flag).
  - (3-asis bitas) SF - ženklo požymis (sign flag).
  - (4-asis bitas) OF - perpildymo požymis (overflow flag).
  - 5-asis, 6-asis, 7-asis ir 8-asis bitai nenaudojamas.
- PTR - puslapiavimo registras - 2 baitai
- PI - 1 baito programinių pertraukimų registras.
- TI - 1 baito laikrodžio pertraukimas registras.
- SI - 1 baito sisteminių pertraukimų registras.

Be registrų procesorius dar turi Aukšto lygio kalbos procesorių - HLP. Jis atsakingas už komandų apdorojimą.

2.2. Vartotojo atmintis

Realios mašinos atminties dydis yra 256 blokai po 16 žodžių. Vieno žodžio ilgis yra 6 baitai.

2.2.1. Puslapiavimas

Norint virtualios mašinos blokų numerius paversti į realios mašinos blokų numerius yra taikomas puslapiavimo mechanizmas. Tam tikslui, kuriant virtualią mašiną, yra išskiriamas dar vienas blokas, kuriame saugama lentelė, atliekanti pervertimą.

Kai PTR reikšmė yra a_0a_1, kur a_0, a_1 yra šešioliktainiai skaičiai. Tada puslapių lentelės bloko adresas yra $16 * a_0 + a_1$

Virtualaus adreso x_0x_1 ($x_0,\ x_1$ - šešioliktainiai skaičiai) realus adresas = $16 * [16 * (16 * a_0 + a_1) + x_0] + x_1$.

2.3. Supervizorinė atmintis

Pirmi 16 realios mašinos blokų yra skiriami supervizorinei atminčiai.

2.4. Pertraukimai

Pertraukimą iškviečia vartotojas. Tada yra pakeičiama registro MODE reikšmė į 1 (supervizoriaus režimas) ir šiame režime yra vykdomas pertraukimas.

Programiniai pertraukimai PI gali kilti, bandant įvykdyti kokį nors neleistiną veiksmą. Šie veiksmai yra:

- PI = 1 - dalyba iš nulio.
- PI = 2 - netinkamas adresas.
- PI = 3 - neleistinas operacijos kodas

Sisteminiai pertraukimai SI:

- SI = 1 - komanda HALT**
- SI = 2 - komanda OPENF*
- SI = 3 - komanda CLOSEF
- SI = 4 - komanda WRITEF
- SI = 5 - komanda READF*
- SI = 6 - komanda DELETF
- SI = 7 - komanda OUTSIM
- SI = 8 - komanda OUTNUM
- SI = 9 - komanda INPONE
- SI = 10 - komanda INPBLK

TI yra atsakingas už taimerio mechanizmą.

2.4.1. Taimerio mechanizmas

Realioje mašinoje yra taimeris. Jis skirtas tam, kad viena užduotis nebūtų vykdoma daugiau nei 10 laiko momentų. Tam tikslui po kiekvienos operacijos yra sumažina TI reikšmė. Taip pat skirtingos operacijos užtrunka kitokį laiko skaičių: išvedimo į kanalų įrenginį ar įvedimo į jį užima 5 laiko vienetus, o visos kitos - 1. Kai TI reikšmė pasiekia 0 yra sugeneruojamas taimerio pertraukimas.

2.5. Kanalų įrenginys

Realioje mašinoje yra įvedimo ir išvedimo įrenginys - atitinkamai klaviatūra ir ekranas. Jie su procesorium apsikeičia duomenimis tik per kanalų įrenginį. Įvydžius komandą EXCHGE yra perkeliamas norimi baitai. Tam, kad būtų tiksliai specifikuota, tai, kas bus perkeliama yra naudojami šie kanalų įrenginio registrai:

- SB - 2 baitų registras, kuriame saugomas bloko, iš kurio bus kopijuojama numeris. Registro reikšmė saugoma kaip šešioliktainis skaičius.

- SW - 2 baitų registras, nuo kurio baito takelyje bus kopijuojama. Pirmame baite kaip šešioliktainis skaičius nurodyta, nuo kurio žodžio kopijuojama. Antrame - saugoma nuo kurio baito tame žodyje bus kopijuojama (tarp 0 ir 5 imtinai).
- DB - 2 baitų registras, kuriame saugas bloko, į kurį bus kopijuojama numeris. Registro reikšmė saugoma kaip šešioliktainis skaičius.
- DW - 2 baitų registras, nuo kurio baito takelyje bus kopijuojama. Pirmame baite kaip šešioliktainis skaičius nurodyta, nuo kurio žodžio kopijuojama. Antrame - saugoma nuo kurio baito tame žodyje bus kopijuojama (tarp 0 ir 5 imtinai).
- BC - 2 baitų registras, kuriame nurodoma, kiek baitų bus kopijuojama. Registro Registro reikšmė saugoma kaip šešioliktainis skaičius.
- ST - 1 baito registras, kuriame nurodytas tipas objekto, kuris bus kopijuotas. ST esančio skaičiaus reikšmės:
    1. Vartotojo atmintis
    2. Supervizorinė atmintis
    3. Išorinė atmintis
    4. Klaviatūra
- DT - 1 baito registras, kuriame nurodytas tipas objekto, į kurį bus kopijuojama. ST esančio skaičiaus reikšmės:
    1. Vartotojo atmintis
    2. Supervizorinė atmintis
    3. Išorinė atmintis
    4. Ekranas

2.6. Išorinė Atmintis

Išorinė atmints yra realizuojama kietuoju disku. Jame yra 256 blokai po 16 žodžių.
Komunikacija su išorine atmintimi yra vykdoma per kanalų įrenginį.

2.7. Klaviatūra

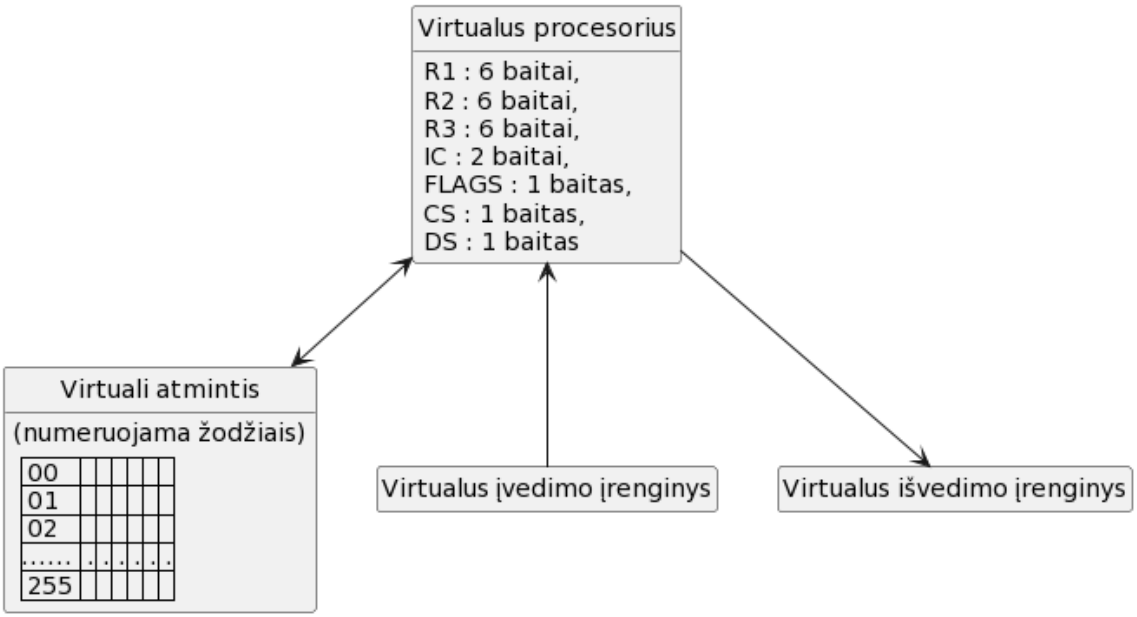
Kiekvienas klaviatūros mygtuko paspaudimai yra operacinės sistemos saugomi iki programa juos pasiima. Pasiimimas vyksta per kanalų įrenginį.

2.8. Ekranas

Į ekraną yra išvedama tiek simbolių. Programa norėdama tai atlikti turi nustatyti registrų reikšmes ir iškviesti atitinkamą sisteminį pertraukimą.

3. Virtualios mašinos modelis

Virtuali mašina - operacinės sistemos konstruktas, kurio pagalba kiekvienos programos veikimas yra izoliajamos nuo visų kitų, o darbui su bendrais įrenginiais pasitelkiama operacinė sistemą. Reali mašina paleidžia virtualią mašiną, o ši vykdo komandas. Šios mašinos modelis yra pateikiamas 2 paveiksliuke.



2 pav. Virtualios mašinos modelis

3.1. Registrai

Virtuali mašina turi priėjimą tik prie šių realios mašinos registrų:

- R1, R2, R3 (bendros paskirties registrai)
- CS (kodo segmento registras)
- DS (duomenų segmento registras)
- FLAGS (požymių registras)
- IC (programos skaitiklis)

3.2. Atmintis

Virtuali mašina turi priėjimą prie 16 blokų. Ji gauna jų virtualius numerius. Dėl to kiekviena virtuali mašina mato, kad jos blokai yra pirmi šešiolika - 0, 1, ..., 14, 15.

3.3. Komandos

Visos komandos yra aprašomos vienu žodžiu - 6 baitais. Visose komandose simbolis * reiškia, kad šis baitas nėra naudojamas.

3.3.1. Aritmetinės

- ADDmxy - jei m = r, tai operacija sudeda skaičius esančius registruose Rx, Ry ir rezultatą patalpina į Rx. Jei m=c, tai operaciją atlieka su registru Rx ir konstanta y. Antru atveju y yra šešioliktainis skaičius.
- SUBmxy - jei m = r, tai operacija atima iš skaičiaus esančio registre Rx, skaičių iš registro Ry ir rezultatą patalpina į Rx. Jei m=c, tai operaciją atlieka su registru Rx ir konstanta y. Antru atveju y yra šešioliktainis skaičius.
- MULmxy - jei m = r, tai operacija sudaugina skaičius esančius registruose Rx, Ry ir rezultatą patalpina į Rx. Jei m=c, tai operaciją atlieka su registru Rx ir konstanta y. Antru atveju y yra šešioliktainis skaičius.
- DIVmxy - jei m = r, tai operacija padalina skaičių esantį registre Rx, iš skaičiaus esančio registre Ry ir rezultatą patalpina į Rx. Jei m=c, tai operaciją atlieka su registru Rx ir konstanta y. Antru atveju y yra šešioliktainis skaičius.
- CMPmxy - jei m = r, tai operacija palygina skaičius esančius registruose Rx ir Ry bei pagal Rx-Ry reikšmę nustato registro FLAGS reikšmę. Jei m=c, tai operaciją atlieka su registru Rx ir konstanta y. Antru atveju y yra šešioliktainis skaičius.

Po kiekvienos aritmetinės operacijos procesorius padina IC reikšmę. Taip pat po kiekvienos aritmetinės operacijos pagal rezultatą (o palyginimo atveju - lyginamųjų skaičių skirtumą) yra nustatomos registro FLAGS reikšmės.

- CF: Nustatomas 1, jei yra pernešimas ar skolinimasis iš vyriausiojo bito, kitu atveju - 0.
- ZF: Jei rezultatas yra 0, tai nustatomas ZF = 1, kitu atveju ZF = 0.
- SF: Jei vyriausias bitas yra 1, tai nustatoma SF = 1, kitu atveju SF = 0.
- OF: Nustatomas 1, jei rezultatas per didelis teigiamas ar permažas neigiamas, kad tilpų į rezultatą nenaudojant ženklo bitu, kitu atveju - 0.

3.3.2. Duomenų judėjimo

- MVmrxy - Perkelia duomenis. m reikšmė nurodo perkelimo režimą. x ir y specifikuoja vietas, kur juda duomenys. Po šios komandos yra padidinama IC reikšmė. Čia x ir y yra šešioliktainiai skaičiai.

1. m = 'r'. Šiuo atveju perkėlimas yra iš registro į registrą. x nurodo pirmo registro (šaltinio) numerį, y - tikslo. Šiuo atveju r reikšmė nenaudojama
2. m = 'o'. Šiuo atveju perkėlimas yra iš registro į atmintį. Atminties vieta yra gaunama pagal formulę $10_{16} \cdot x + y$, o registro numeris - pagal r reikšmę
3. m = 'i'. Šiuo atveju perkėlimas yra iš atminties į registrą. Atminties vieta yra gaunama pagal formulę $10_{16} \cdot x + y$, o registro numeris - pagal r reikšmę
4. m = 'c'. Šiuo atveju į registrą yra perkeliama konstanta. Registro numerį nurodo r, o konstanta yra $10_{16} \cdot x + y$, kur x, y - šešioliktainiai skaičiai. Konstanta pakeičia registro reikšmę.

Registrai yra numeruojami taip:

1. R1
2. R2
3. R3
4. CS
5. DS

3.3.3. Valdymo

Norint tinkamai naudotis perėjimais, prieš juos reikia panaudoti CMP operaciją. Komandose naudojami x, y, z yra šešioliktainiai skaičiai

- JMP*xy - besąlyginis perėjimas. Pakeičia IC := $10_{16} * \text{x} + \text{y}$
- JE**xy - pereiti, jeigu lygu. Jei ZF=1, tai IC := $10_{16} * \text{x} + \text{y}$, kitu atveju IC := IC + 1.
- JNE*xy - pereiti, jei nelygu. Jei ZF=0, tai IC := $10_{16} * \text{x} + \text{y}$, kitu atveju IC := IC + 1.
- JL**xy - pereiti, jeigu mažiau (su ženklu). Jei SF≠OF, tai IC := $10_{16} * \text{x} + \text{y}$, kitu atveju IC := IC + 1.
- JLE*xy - pereiti, jeigu mažiau arba lygu (su ženklu). Jei ZF=1 arba SF≠OF, tai IC := $10_{16} * \text{x} + \text{y}$, kitu atveju IC := IC + 1.
- JG**xy - pereiti, jeigu daugiau (su ženklu). Jei ZF=0 arba SF=OF, tai IC := $10_{16} * \text{x} + \text{y}$, kitu atveju IC := IC + 1.
- JGE*xy - pereiti jeigu daugiau arba lygu (su ženklu). Jei SF=OF, tai IC := $10_{16} * \text{x} + \text{y}$, kitu atveju IC := IC + 1.

- JB**xy - pereiti, jei žemiau (be ženklo). Jei CF=1, tai IC := $10_{16} * x + y$, kitu atveju IC := IC + 1.
- JBE*xy - pereiti, jei žemiau ar lygu (be ženklo). Jei CF=1 arba ZF=1, tai IC := $10_{16} * x + y$, kitu atveju IC := IC + 1.
- JA**xy - pereiti, jei aukščiau (be ženklo). Jei CF=0 ir ZF = 0, tai IC := $10_{16} * x + y$, kitu atveju IC := IC + 1.
- JAE*xy - pereiti, jei aukščiau arba lygu (be ženklo). Jei CF = 0, tai IC := $10_{16} * x + y$, kitu atveju IC := IC + 1.

Po valdymo komandų registro FLAGS reikšmė nėra keičiama.

3.3.4. Darbas su failais

Visos šios operacijos iškviečia sisteminius pertraukimus. Procesorius informaciją apie pertraukimo tipą gauna iš SI reikšmės.

- OPENF* - atidaro failą, kurio pavadinimas yra žodis, įrašytas adresu esančiu R1 reikšmėje. Operacijos pabaigoje į R2 yra įrašomas failo numeris. Jeigu failas yra nerandamas - sukuria naują failą tokiu pavadinimu.
- CLOSEF - uždaro failą, kurio numeris yra įrašytas R2. Jeigu norimas failas nėra rastas, ar jau buvo uždarytas yra nustatoma R1 = 0, kitu atveju R1 bus uždaryto failo numeris.
- WRITEF - rašo simbolius į failą, kurio numeris nurodytas R2. Rašoma nuo baito, kuris nurodytas R1 iki tol, kol parašo R3 baitų arba sutinkamas nulinis baitas.
- READF* - skaito simbolius iš failo, kurio numeris yra nurodytas R2. Rašymas į atmintį pradedamas nuo to baito, kurį nurodo R1 reikšmė. Perskaitomi baitai iki failo galo, bet nedaugiau negu buvo nurodyta reikšmėje R3. Jeigu buvo perskaityta mažiau, negu buvo nurodyta R3 reikšmėje, tai ji atitinkamai pakeičiama.
- DELETF - sunaikiną failą, kurio numeris nurodomas R2 reikmšmėje. Jeigu norimas failas nebeegzistuoja arba nepavyksta jo sunaikinti - operacijos pabaigoje įrašo į R2 reikšmę 0, kitu atveju nekeičia R2 reikšmės. Sunaikinus failą jo nereikia uždarinėti.

Po kiekvienos iš šių komandų procesorius padidina registro IC reikšmę.

Naują eilutę darbe su failais bei įvedime ir išvedime žymi du simboliai

n.

Jeigu WRITEF nepavyko parašyti visų simbolių, tai yra nustatomas CF = 1, o priešingu atveju CF reikšmė yra išvaloma. Kitos valdymo komandos nekeičia registro FLAGS reikšmės.

3.3.5. Įvedimas ir išvedimas

Įvedimo ir išvedimo komandos, kaip ir darbo su failais, iškviečia sisteminį pertraukimą.

- OUTSIM - išveda baitus nuo R1 baitais nurodyto adreso ir parašo tiek simbolių, kokia reikšmė dabar yra registre R3.
- OUTNUM - išveda R1 reikšmę kaip skaičių.
- INPLIN - perskaito vieną eilutę iš įvedimo srauto ir patalpina į atmintį, nuo adreso, nurodyto registre R1, tačiau nedaugiau simbolių nei nurodyta registre R3. Kol eilutė nėra gaunama, programa užsiblokuoja.
- INPNUM - perskaito skaičių iš įvedimo srauto ir patalpine registre R1.

3.3.6. Programos pabaigos

- HALT** - iškviečia sisteminį pertraukimą, kuris baigia programos darbą.

3.4. Programos struktūra

Maksimalus programos dydis yra 256 žodžiai, t.y. 16 blokų. Failo pavadinimas gali būti tik vieno žodžio.

Programos pradžią žymi žodis $PROG$. Po jo yra pavadinimas, o žymė —— (šeši '-' ženklai) - pavadinimo pabaigą. Ši dalis užima pirmąjį takelį. Taip pat pabaigos žymė turi būti viename žodyje.

Prieš kiekvieną naują programos ar duomenų bloką turi būti žymė. Žymės yra vienos iš dviejų tipų: pozicinės ir vardinės.

Pozicinės yra aprašomas formatu $$x$$$, kur x yra šešioliktainis skaičius, nurodantis takelio numerį (numeruojama nuo 0), į kurį bus rašomi duomenys.

Vardinės yra dvi žymės: .CODES ir .DATAS, kurios atitinkamai reiškia kodo segmentą ir duomenų segmentą. Jos taip pat atinkama pozicines žymes $$1$$$ ir $$8$$$. Programoje šių žymių gali ir nebūti.

Programos darbo pradžioje yra nustatomos tokios reikšmės: IC = 10_{16}, CS = 10_{16}, DS = 80_{16}.

Programos pabaiga žymi žodis $FINS$.

3.5. Failo struktūra

Failo maksimalus dydis yra neribojamas, tačiau failo pavadinimas yra vienas žodis.

Failo pradžią žymi žodis $FILE$. Po jo yra failo pavadinimas, o žymė —— (šeši '-' ženklai) žymi pavadinimo pabaigą.

Po jų seka failas. Jis yra baigiamas žodžiu $FINS$

3.6. Išorinio disko pavyzdys

Išorinėje atmintyje duomenys saugomi failais ir programomis. Laikoma, kad visa atmintis, kuri netenkina aprašytos struktūros yra ignoruojama.

Žemiau nurodomas galimas kietojo disko pavyzdys su trimis programomis:

1. ”ciklas” - programa vykdanti amžinąjį ciklą.
2. ”sudeti” - iš vartotojo įvesties gauna du skaičius ir išveda jų sumą.
3. ”jungti” - iš vartotojo įvesties gauna dviejų failų pavadinimus, skaito juose esantį tekstą ir sujungtą failą ”sujungtas” (pradžioje pirmo, tada antro). Jei trečiasis failas pradžioje buvo netuščias, tai jį ištrina.

```
$PROG$
$PROG$
ciklas
------
.CODES
JMP*10
.DATAS
$FINS$
$PROG$
sudeti
------
.CODES
MVrr51
MULc16
MVc318
OUTSIM
INPNUM
MVrr12
MVc190
MULc16
MVc318
OUTSIM
INPNUM
ADDr12
MVrr12
MVc1F0
```

```
MVc30B
MVrr21
OUTNUM
HALT**
.DATAS
iveski
te pir
ma ska
iciu\n
$$9$$$
iveski
te ant
ra ska
iciu\n
$$A$$$
gauta
suma:
$FINS$
$PROG$
jungti
------
.CODES
MVc102
MVo1C2
MVc1A0
OPENF*
DELETF
OPENF*
MVo2C0
MVrr51
MULc16
MVc31B
OUTSIM
MVc190
MULc16
MVc310
INPLIN
DIVc16
OPENF*
```

```
MVo2C1
MVc1A0
MULc16
MVc360
MVi2C1
READF*
MVi2C0
WRITEF
CMPc30
JNE*22
MVi2C1
CLOSEF
MVi1C2
SUBc11
MVo1C2
CMPc10
JNE*17
MVi2C0
CLOSEF
HALT**
.DATAS
Iveski
te fai
lo pav
adinim
a\n
$$A$$$
sujung
tas
$FINS$
$FILE$
pirmas
------
pirmoj
o fail
o eilu
tes, k
urios
reikal
```

```
ingos
jungim
ui.
$FINS$
$FILE$
antras
------
antroj
o fail
o eilu
tes, k
urios
reikal
ingos
jungim
ui.
$FINS$
$PROG$
aritme
------
.CODES
ADDc1A
ADDc27
MULr12
DIVc13
.DATAS
$FINS$
$PROG$
outtes
------
.CODES
MVc115
OUTNUM
MVc110
MULc16
MVc312
OUTSIM
HALT**
.DATAS
```

```
$FINS$
$FILE$
sujung
------
pirmoj
o fail
o eilu
tes, k
urios
reikal
ingos
jungim
ui.
antroj
o fail
o eilu
tes, k
urios
reikal
ingos
jungim
ui.
$FINS$
```

3.7. Virtuali mašina operacinės sistemos kontekste

Operacinė sistema paleidžia virtualią mašiną kiekvienai programai. Jei operacinė sistema yra multiprograminė, tai joje vienu metu veikia tik viena programa, bet vykdomoji programa nuolatos keičiasi.

Virtualias mašinas kuria operacinė sistemą. Jos yra sukuriamos, kai vartotojas įveda programą LOAD prg, kur prg yra programos pavadinimas kietąjame diske. Tada programa yra perkeliama į supervizorinę atmintį, joje yra patikrinama ar programoje nėra neegzistuojančių komandų. Jei yra - įkėlimas nutraukiamas, jei nėra - programa perkeliama į vartotojo atmintį.

Pažingsniui kūrimo scenarijus būtų toks:

1. Pradedama kurti virtuali mašina.
2. Ji reikalauja 16 takelių.
3. Išskiriamas papildomas takelis puslapių lentelei, kuris užpildomas adresais.

4. Nustatoma registro PTR reikšmė su puslapių lentelės adresu.

5. Virtuali mašina baigiama kurti ir gauna procesorių.