

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS

Projektas

Virtualios bei realios mašinos apibrėžimas ir
realizacija

Dovilė Patiejūnaitė ir Paulius Matijošaitis

Bioinformatika III k.

VILNIUS
2018

TURINYS

1 ĮVADAS	1
2 REALI MAŠINA	2
2.1 PROCESORIUS.....	3
2.1.1 REGISTRAI.....	3
2.2 OPERATYVI ATMINTIS	4
2.3 KOMANDOS.....	4
2.4 KLAIDŲ KODAI	5
3 VIRTUALI MAŠINA	6
3.1 PROCESORIUS.....	6
3.2 OPERATYVI ATMINTIS	6
3.3 KOMANDOS.....	6
3.4 KLAIDŲ KODAI	8
4. PERTRAUKIMŲ MECHANIZMAS	8
5 ĮVEDIMO/ IŠVEDIMO ĮRENGINIAI.....	10
6 PUSLAPIAVIMO MECHANIZMAS.....	11
7. PROGRAMOS FAILO FORMATAS	12

1 ĮVADAS

Šio rašto darbo paskirtis yra pateikti virtualios mašinos ir ją palaikančios realios mašinos modelius. Šie modeliai yra skirti panaudoti šių mašinų realizacijai ir šioms mašinoms skirtos operacinės sistemos projektavimui ir realizacijai.

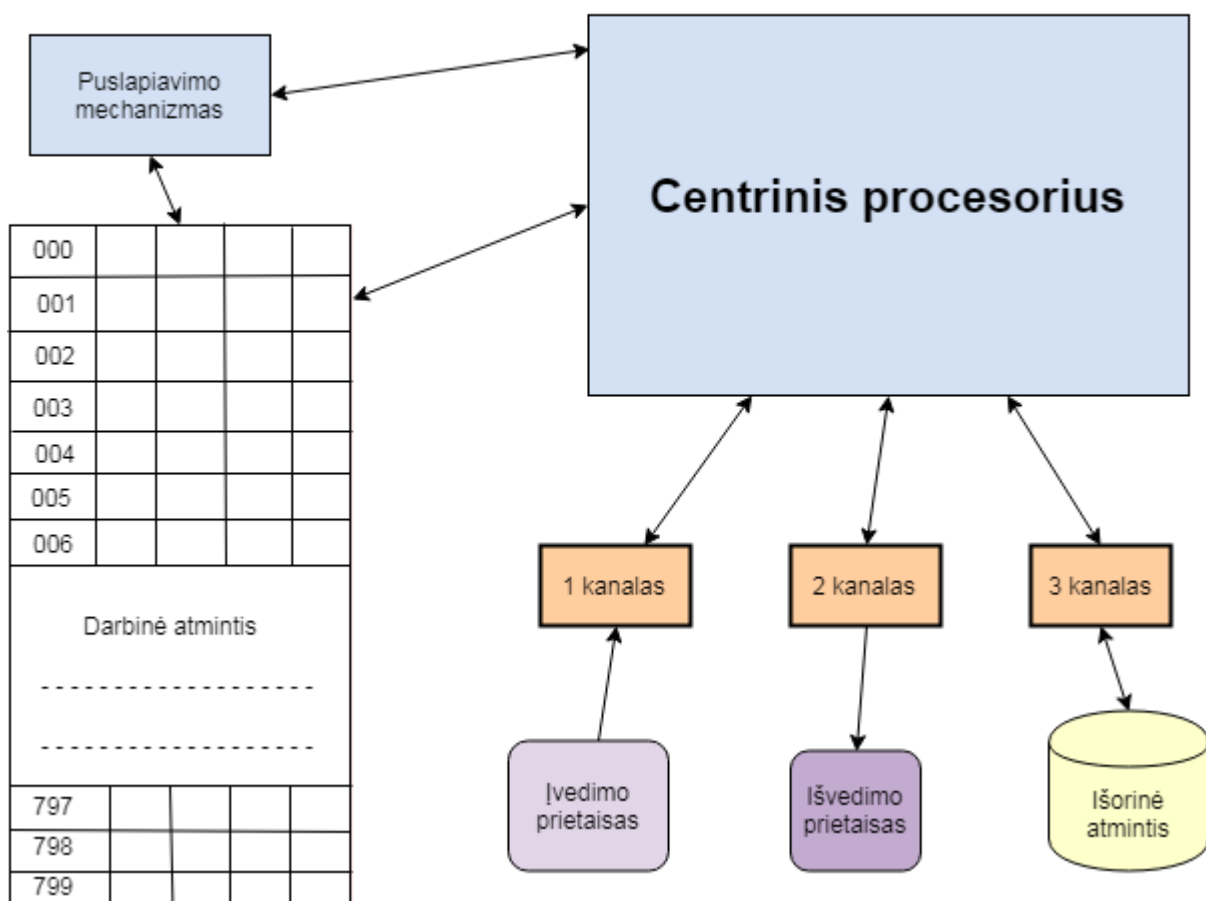
Reali mašina (RM) – tai virtualus kompiuteris, kuris turi tik žemo lygio komandas. RM turi pasižymėti dviem režimais: vartotojo ir supervizoriaus. Jei RM dirba vartotojo režimu, tai ji faktiškai sutampa su virtualios mašinos (VM) dalimis.

VM turi kelias realios mašinos dalis, tačiau VM yra suteikta daug paprastesnė nei reali vartotojo sąsaja, kas palengviną programavimą. Taigi virtuali mašina dirba su operacinės sistemos pateiktais virtualiais resursais, kurie daugelį savybių perima iš savo realių analogų. Kiekviena VM atlieka vieną paskirtą užduotį.

2 REALI MAŠINA

Operacinė sistema – programa, kuri modeliuoja kelių virtualių mašinų darbą realioje mašinoje. Realioji mašina – tai kompiuteris. Toks kompiuteris (toliau kalbama apie mūsų realiąją mašiną), kuris turi savo procesorių, darbinę ir išorinę atmintį, įvedimo ir išvedimo prietaisus, 2 režimus, kurie apibūdina mašinos būseną – supervizorinę ar vartotojo.

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



2.1 PROCESORIUS

Procesoriaus paskirtis – skaityti komandą iš atminties ir ją vykdyti. Procesorius yra pajėgus dirbti dviem režimais – supervizoriaus ir vartotojo.

2.1.1 REGISTRAI

Procesoriuje esantys registrai:

- **PTR** – 4 baitų puslapių lentelės registras, naudojamas atvaizdavimui tarp VM ir RM.
PTR registras nuo vyriausio baito turi 4-ias reikšmes a0, a1, a2, a3. a0
a0,a1 – puslapių lentelės dydis, a2 ,a3 – bloko numeris, kuriame saugoma lentelė.
- **R** – 8 baitų bendrojo naudojimo registras. Galimos reikšmės – tik sveikieji skaičiai.
- **CS** – 4 baitų registras, saugantis kodo segmento pradžią.
- **SS** – 4 baitų registras, saugantis steko segmento pradžią.
- **IP** - 4 baitų registras, rodantis poslinkį nuo kodo segmento pradžios, komandos vykdymo metu rodo į sekančios vykdomos komandos pradžią.
- **SP** – 4 baitų registras, rodantis į steko viršūnę.
- **IR** – 1 baito registras, kuriame saugomi pertraukimų kodai. Galimi tik sveikieji skaičiai.
- **MODE** – registras, nusakantis procesoriaus režimą(vartotojas(0) arba supervizorius(1))
- **TI** – taimerio registras, nustato pertraukimą, kai pasiekiamas 0. Palaipsniui mažėja atliekant komandą vartotojo režime.
- **CF** – 1 baito registras, saugantis reikšmes 0 arba 1. Įgija reikšmę 1, kai aritmetinių komandų rezultatas netelpa į R registrą(perpildymas).
- **ZF** - 1 baito registras, saugantis reikšmes 0 arba 1. Įgija reikšmę 1, jeigu tikrinamas rezultatas yra 0.
- **SF** - 1 baito registras, saugantis reikšmes 0 arba 1. Įgija reikšmę 1, kai rezultatas neigiamas.
- **ERR** – 2 baitų registras, saugantis virtualios mašinos klaidų kodus.
- **RS** – 8 baitų registras, skirtas eilutinėms komandoms.
- **P** – 8 baitų pagalbinis registras, skirtas papildomoms operacijoms.
- **RMB** – 4 baitų registras, galimos reikšmės tik sveiki skaičiai, saugo realios mašinos steko viršūnės reikšmę.
- **RE** – 2 baitų registras, saugantis realios mašinos klaidų kodus.
- **CHST(1), CHST(2), CHST(3)** – 1 baito kanalų būsenos(stabdymo) registrai. 1 kanalas skirtas įvedimo įrenginiui, 2 – išvedimo įrenginiui ir 3 – išoriniai atminčiai.

2.2 OPERATYVI ATMINTIS

Reali mašina turi dviejų rūšių atmintį: darbinę ir išorinę.

Viso darbinės atminties dydis yra 800 žodžių po 8 baitus. Darbinė atmintis yra suskaidyta į segmentus. Pertraukimų vektorių lentelės segmentas – nuo 0-9 (10 žodžių atmintyje), puslapių lentelės segmentas yra nuo 10 iki 59, kodo segmentas – nuo 60 iki 699 (640 žodžių) ir steko segmentas – nuo 700-799(100 žodžių).

Kodo segmente yra laikoma virtualių mašinų atmintis ir realios mašinos komandos. Šio segmento dydis yra lygus 64 blokai, nes vienas blokas bus sudarytas iš 10 žodžių.

Išorinė atmintis bus realizuota failu kietajame diske. Laikysime, kad kietojo disko atmintis 2000 žodžių.

2.3 KOMANDOS

Visos virtualios mašinos komandos yra ir realios mašinos komandos. (skyrelis 3.3). Visos procesoriaus komandos susideda iš tokios struktūros: 4 baitus sudaro instrukcija ir likusius 4 baitus sudaro parametrai.

Toliau pateikiamos komandos, galimos tik supervizoriaus režime(parametrai bus žymimi *):

PR ***** - išveda į ekraną žodį, esantį adresu *****.

WRT ***** - iš įvesties rašo žodį į atmintį adresu *****.

SETM – nustato mašinos režimą į vartotojo. RMB registras nustatomas dabartine SP reikšmę. Kodo segmento registro reikšmė nustatoma į 0000. Pagal tai ar grįžtama į prieš tai buvusią virtualią mašiną, ar buvo sukurta nauja, nustatomos likusios registrų reikšmės ir išskiriama steko atmintis.

INT * - iškviečiamas pertraukimas rankiniu būdu, * – pertraukimo numeris. Į registrą IR įdedamas *. (IR:=*)

IRET – grįžimo į virtualią mašiną komanda. Iš steko grąžinamos registrų reikšmės tokia tvarka: (CF, SF, ZF), ERR – negrąžinamas, SP, SS, RS, P, R, IP, PTR.

SCH * – *=1,2,3. Priklausomai nuo kanalo, pakeičia CHST(*) reikšmę į 1(1-užimtas, 0-laisvas).

CLCH * – I=1,2,3. Nustatoma CHST(*) kanalo reikšmę į 0 – laisvas.

SETT ** - Nustato taimerio reikšmę į **. TI:=**.

INC ** - padidina registro reikšmę vienetu. Galimi registrai: TI, SP, IP.

DEC ** - sumažina registro reikšmę vienetu. Galimi registrai: TI, SP, IP.

REDH***** – vieno bloko atminties iš kietojo disko skaitymas į atmintį prasidedančią adresu *****.

WRTH***** – vieno bloko atminties prasidedančios adresu ***** rašymas į kietąjį diską.

SPTR***** - nustato registro PTR reikšmę.

VMCR – sukuria naują virtualią mašiną.

PRMB***** - įrašo į atmintį RMB registrą.

MIV ***** - 1 žvaigždutė reiškia pertraukimo numerį, o likusios - IP adresą, ties kuriuo pradėsime vykdyti pertraukimą.

LIV * - įrašo pertraukimo vykdymo adresą į R registrą.

WTI ***** - įrašo į atmintį timer registro reikšmę.

WPTR***** - įrašo PTR reikšmę į atmintį.

PRER – išveda virtualios mašinos klaidos kodą.

PRRE – išveda realios mašinos klaidos kodą.

2.4 KLAIDŲ KODAI

1. Nėra laisvos atminties
2. Komandos INC arba DEC registrai nėra nei TI, nei SP ar IP
3. Neteisingas adresas
4. Komandos INC ar DEC registras nėra nei TI, nei SP ar IP
5. Netelpa duomenys į registro režius
6. Registras neegzistuoja INC arba DEC komandai
7. Kanalas neegzistuoja
8. Neteisinga registro reikšmė
- 9.
10. Registre yra ne skaitinė reikšmė
11. Neteisinga įvestis iš klaviatūros
12. Dalyba iš nulio
13. Neteisingi komandos parametrai
14. Vykdyta POP komanda, tačiau stekas tuščias

15. Bandoma rašyti, bet kanalas užimtas
16. Netinkamas duomenų rašyti į atmintį dydis

3 VIRTUALI MAŠINA

Viena virtuali mašina vykdo vieną užduotį. Virtualios mašinos paskirtis vykdyti komandą, kuri yra virtualioje atmintyje. Taigi atminties žodis interpretuojamas kaip komanda arba kaip duomenys. VM pradeda darbą, kai registro IP reikšmė yra 0000. CS=0000, SP=0090.

3.1 PROCESORIUS

Procesorius minimalizuotas, yra mažesnė realios mašinos procesoriaus versija. Jį sudaro šie realios mašinos registrai:

CF, ZF, SF, RS, R, P, IP, CS, SS, SP(šie registrai aprašyti 2.1.1 skyrelyje).

3.2 OPERATYVI ATMINTIS

Kiekvienai virtualiai mašinai skiriama 10 vartotojo atminties blokų (1 blokas – 10 žodžių). Užduoties programa turi tilpti į 100 žodžių. Kiekvienas virtualios atminties blokas turi virtualų ir realų adresą. Operatyvi turi kodo ir steko segmentus. CS (0000-0089), SS(0090-0099).

3.3 KOMANDOS

„a“ – adresas.

Aritmetinės darbo su sveikaisiais skaičiais komandos:

- ADD $x_1x_2x_3x_4 - 2$ sveikų aštuonženklių skaičių sudėties komanda ($R := R + [a]$). Atitinkamai nustato CF, ZF ir SF registrų reikšmes. Netilpus sudėties rezultatui į R registrą, likusios reikšmės yra dedamos į P registrą(jaunesnieji, t.y.dešinesni skaičiai), nustatomas CF į 1.
- SUB $x_1x_2x_3x_4 - 2$ sveikų aštuonženklių skaičių atimties komanda ($R := R - [a]$). Atitinkamai nustato ZF ir SF registrų reikšmes.
- MUL $x_1x_2x_3x_4 - 2$ sveikų aštuonženklių skaičių sandaugos komanda ($R := R * [a]$). Nustato CF, ZF registrų reikšmes.
- DIV $x_1x_2x_3x_4 - 2$ sveikų aštuonženklių skaičių dalybos komanda ($R := R / [a]$). Nustato ZF registro reikšmę.
- MOD $x_1x_2x_3x_4 - 2$ sveikų aštuonženklių skaičių liekanos suradimo komanda ($R := R \% [a]$). Keičiasi tik ZF.

Loginės komandos:

- XOR $x_1x_2x_3x_4$ - 2 sveikų aštuonženklių skaičių XOR operacija, išimtinis „arba“ ($R:=R \text{ xor } [a]$).
- AND $x_1x_2x_3x_4$ - 2 sveikų aštuonženklių skaičių AND operacija, loginis „ir“ ($R:=R \text{ and } [a]$).
- OR $x_1x_2x_3x_4$ - 2 sveikų aštuonženklių skaičių OR operacija, loginis „arba“ ($R:=R \text{ or } [a]$).
- NOT - NOT operacija, loginis „neigimas“ ($R:= \text{NOT } R$)

Duomenims apdoroti skirtos komandos:

- $\text{LOAD}_{x_1x_2x_3x_4}$ – žodžio, esančio adresu a, pakrovimas į registrą R($R:=[a]$).
- $\text{STOR}_{x_1x_2x_3x_4}$ – registro R reikšmės įrašymas į atmintį su adresu a($[a]:=R$).
- $\text{STOS}_{x_1x_2x_3x_4}$ – registro RS reikšmės įrašymas į atmintį su adresu a($[a]:=RS$).
- $\text{LODS}_{x_1x_2x_3x_4}$ – žodžio, esančio adresu a, pakrovimas į registrą RS($RS:=[a]$).
- $\text{LODJ}_{x_1x_2x_3x_4}$ – 4 baitų pakrovimas į R registro dešinę reikšmių pusę.
- $\text{LODL}_{x_1x_2x_3x_4}$ – 4 baitų pakrovimas į R registro kairę reikšmių pusę.
- $\text{LOSR}_{x_1x_2x_3x_4}$ – 4 baitų pakrovimas į RS registro dešinę reikšmių pusę.
- $\text{LOSL}_{x_1x_2x_3x_4}$ – 4 baitų pakrovimas į RS registro kairę reikšmių pusę.
- $\text{WRTP}_{x_1x_2x_3x_4}$ – registro P reikšmės įrašymas į atmintį su adresu a($[a]:=P$).
- $\text{WRIP}_{x_1x_2x_3x_4}$ – registro IP reikšmės įrašymas į atmintį su adresu a($[a]:=IP$).
- $\text{WRSP}_{x_1x_2x_3x_4}$ – registro SP reikšmės įrašymas į atmintį su adresu a($[a]:=SP$).
- $\text{WRSS}_{x_1x_2x_3x_4}$ – registro SS reikšmės įrašymas į atmintį su adresu a($[a]:=SS$).

Palyginimo komandos:

- $\text{CPR}_{x_1x_2x_3x_4}$ - veikia panašiai kaip SUB komanda, bet nekeičia nei atminties, nei R registro reikšmių. Palygina registro R ir atminties su adresu a reikšmes ir priskiria reikšmes. Jei reikšmės lygios tai $ZF=1$. Jei R yra mažiau už atmintyje esančią reikšmę, tai $SF=1$.
- $\text{CPS}_{x_1x_2x_3x_4}$ – jeigu RS registro ir atminties su adresu a reikšmės yra lygios($RS=[a]$), tada $ZF=1$.

Įvedimo/išvedimo komandos:

- $\text{PRIN}_{x_1x_2x_3x_4}$ – atminties turinio su adresu a išvedimas į ekraną. Sukelia pertraukimą Nr. 4, žodį iš adreso įdeda į R registrą.
- $\text{WRIT}_{x_1x_2x_3x_4}$ – žodžio iš įvedimo įrenginio rašymas į atmintį su adresu a. Sukelia pertraukimą Nr. 3, žodį iš adreso įdeda į R registrą.

Valdymo perdavimo komandos:

Nesąlyginio:

- $\text{GO}_{x_1x_2x_3x_4}$ – valdymas perduodamas nurodytu adresu. $IP := x_1x_2x_3x_4$.
- HALT – vartotojo programos vykdymo pabaiga. Sukelia pertraukimą nr. 1.

Sąlyginio:

- $\text{JE}_{x_1x_2x_3x_4}$ – pereiti jeigu lygu. Jei $ZF = 1$, tai $IP := x_1x_2x_3x_4$.
- $\text{JN}_{x_1x_2x_3x_4}$ - pereiti, jeigu nelygu. Jei $ZF = 0$, tai $IP := x_1x_2x_3x_4$.
- $\text{JO}_{x_1x_2x_3x_4}$ - pereiti, jeigu jei $CF = 1$, tai $IP := x_1x_2x_3x_4$.
- $\text{JG}_{x_1x_2x_3x_4}$ - pereiti, jeigu daugiau. Jei $SF=0$, tai $IP := x_1x_2x_3x_4$.
- $\text{JL}_{x_1x_2x_3x_4}$ - pereiti, jeigu mažiau. Jei $SF=1$, tai $IP := x_1x_2x_3x_4$.

Steko valdymo komandos:

- PUSH – iš R registro 8 baitai įdedami į steką. $SP := SP + 1$;
- PUSS – iš RS registro 8 baitai įdedami į steką. $SP := SP + 1$;

- POP – iš steko viršūnės išimami 8 baitai ir įdedami į R registrą. SP:=SP-1.
- POPS – iš steko viršūnės išimami 8 baitai ir įdedami į RS registrą. SP:=SP-1
- CLR – ištuština steką. SP:=0000.
- PUSM_{x₁x₂x₃x₄} – į steką įdedami 8 baitai iš atminties a. SP:=SP+1.
- POPM_{x₁x₂x₃x₄} – iš steko viršūnės išimami 8 baitai ir įdedami į atmintį a. SP:=SP-1
- PUSF – į steką įdemami CF, SF, ZF registrų reikšmės, sujungtos į vieną žodį. SP:=SP+1
- POPF – iš steko viršaus gražinamos CF, SF, ZF reikšmės. SP:=SP-1

Skaitymo/rašymo į kietąjį diską komandos:

- REHAX₁x₂x₃x₄ – į R registrą įdedama atminties reikšmė ir nustatomas pertraukimas Nr. 6.
- WRHAX₁x₂x₃x₄ – į R registrą įdedamas adresas ir nustatomas pertraukimas Nr. 5.

3.4 KLAIDŲ KODAI

ERR registre atsirandantys klaidų kodai, vykdam programą. ERR registrui esant 0, skaitome, kad klaidų nėra ir virtuali mašina baigia darbą, tik pasiekusi \$END komandą. ERR registrui pasikeitus, VM baigia darbą, vykdomas pertraukimas nr 1.

1. Netelpa duomenys į registro režius
2. Dalyba iš nulio
3. Neteisingas adresas;
4. Kanalas neegzistuoja
5. Vykdyta POP komanda, tačiau stekas tuščias
6. Programos pradžioje nėra \$STR
7. Neteisinga registro reikšmė
8. Programos pabaigoje nėra \$END
9. Registre yra ne skaitinė reikšmė
10. Neteisinga įvestis iš klaviatūros
11. \$STR panaudotas daugiau nei 1 kartą
12. Neteisingi funkcijos parametrai
13. Bandoma vykdyti komandą be supervizorinių teisių
14. Neatpažinta komanda

4.PERTRAUKIMŲ MECHANIZMAS

Realios mašinos registrui IR gavus vieną iš kodų, persijungiama į MODE:=1.(supervizoriaus režimą). Taigi vienintelis būdas įjungti supervizoriaus režimą yra pertraukimai. Vykdomas int, pagal numerį, iš pertraukimų lentelės, kurioje yra CS:IP adresai, kuriuose laikomos pertraukimo komandos vykdomas kodas. T.Y. valdymas perduodamas adresu esančiu pertraukimo lentelėje. Toks kodas,

pasibaigia IRET funkcija, kad galėtume grįžti į programą, kurioje buvome prieš įvykstant pertraukimui.(MODE := 0)

Pertraukimų kodai:

1. VM baigė darbą.
2. TIME pertraukimas. (TI=0)
3. INPUT – įvedimas iš klaviatūros.
4. OUTPUT – išvedimas į ekraną.
5. Rašymas į HDD.
6. Skaitymas iš HDD.

5 ĮVEDIMO/ IŠVEDIMO ĮRENGINIAI

Išvedimo įrenginių paskirtis – apdoroti vartotojo įvestas komandas ir išvesti rezultatus į grafinę sąsają. Įvedimo įrenginiai leidžia priimti tam tikrą informaciją į kompiuterį, o išvedimo prietaisai leidžia tą apdorotą informaciją pateikti vartotojui. Duomenų srautus – įvedimus ir išvedimus, kontroliuoja 3 kanalai. Kiekvienas kanalas turi savo registrą CHST, kurio reikšmė arba 1 arba 0(kanalas užimtas(1) arba neužimtas(0)). Jeigu kanalas yra laisvas, galima vykdyti įrašymo, skaitymo operacijas taip pat veiksmus su išorine atmintimi. Visus šiuos veiksmus kontroliuos pertraukimai programos vykdymo metu.

6 PUSLAPIAVIMO MECHANIZMAS

Ryšys tarp virtualaus ir realaus adreso nusakomas puslapių lentelėmis. Kiekvienai naujai sukurtai virtualiai mašinai reikia skirti atminties takelių(blokų) iš realios atminties. Tai galima įgyvendinti puslapiavimo mechanizmu. Tarkime pradedame kurti naują virtualią mašiną – jai reikia blokų atminties(nuo 0 iki 9). Galime pasirinkti blokus su tam tikrais numeriais iš realios atminties(nesvarbu koki numeriai – svarbu, kad būtų laisvi blokai ir egzistuoti tokie numeriai). Kaip pavyzdys galėtų būti toks: pasirenkame laisvus realios atminties blokus 2, 6, 8, 15, 17 (iš viso 5). Pati virtuali mašina šiuos blokus matys sunumeruotus nuo 0 iki n-1, kai n yra norimas realios atminties blokų skaičius (iš viso yra nuo 0 iki 29). Kad įgyvendinti tokį sąryšį į pagalbą pasitelkiame puslapių lentelę. Kiekvieno žodžio eilės numeris atitiks virtualios mašinos takelio numerį ir žodyje bus laikomas realus to takelio numeris. Šiuo konkrečiu pavyzdžiu puslapių lentelėje realios atminties blokai atitiks virtualios mašinos eilės numerius tokia tvarka: 2 -> 0; 6 -> 1; 8 -> 2; 15 -> 3; 17 -> 4; Puslapių mechanizmas pereina veiksmų seką kuri prasideda puslapių lentelės bloko numeriu baigiantis virtualiu absoliučiu adresu. Anksčiau minėtas Registras PTR – puslapių lentelės adresas. Kiekviena virtuali mašina, prieš pradėdama darbą, nustatys šį registrą jai reikalinga reikšme. Galima sakyti, kad galutinis puslapiavimo meachanizmo rezultatas yra absoliutus adresas. Bet svarbiausia dalis – registras PTR susiejantis Realias ir Virtualias mašinas, kurio parametrai buvo paminėti anksčiau.

7. PROGRAMOS FAILO FORMATAS

Programos failo formatas turi savo vykdymo parametrus ir duomenis.

Parametrai:

1. Pirmasis laukas – 8 baitai, nurodantys programos pradžią. \$STR.
2. Pabaigos žymė – 8 baitai, nurodantys programos pabaigą. \$END

Duomenims skirta 100 žodžių – 10 blokų. T.y. 800 baitų.

\$STR

/komandos po 8 baitus/

\$END