

Virtualios ir realios mašinos projektas

Įžanga	3
Realios mašinos projektas	4
Techninės įrangos komponentai	4
Centrinis Procesorius	4
Naudotojo atmintis	6
Supervizorinė atmintis	6
Puslapiavimo Mechanizmas	6
Išorinė atmintis	6
Duomenų perdavimo kanalai	6
Įvedimo ir išvedimo įrenginiai	7
Taimerio Mechanizmas	7
Virtualios mašinos projektas	8
Virtualios mašinos aprašymas	8
Virtualios mašinos komponentai	8
Virtualios mašinos komandos	9
Darbo su duomenimis komandos	10
Aritmetinės komandos	10
Palyginimo komanda	10
Valdymo perdavimo komandos	10
Įvedimo ir išvedimo komandos	11
Semaforų komandos	11
Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais mechanizmo aprašymas	12
Virtualios mašinos interpretuojamojo ar kompiliuojamo vykdomojo failo išeities teksto formatas	12
Modeliuojamos virtualios mašinos loginių komponentų sąryšio su realios mašinos techninės įrangos komponentais aprašymas	13
Trumpas aprašas, kaip virtuali mašina įsivaizduojama visos operacinės sistemos kontekste.	13

Įžanga

Šiame projekte yra pateiktas interaktyvios operacinės sistemos aprašas. Jis yra parengtas pagal pirmą užduotį:

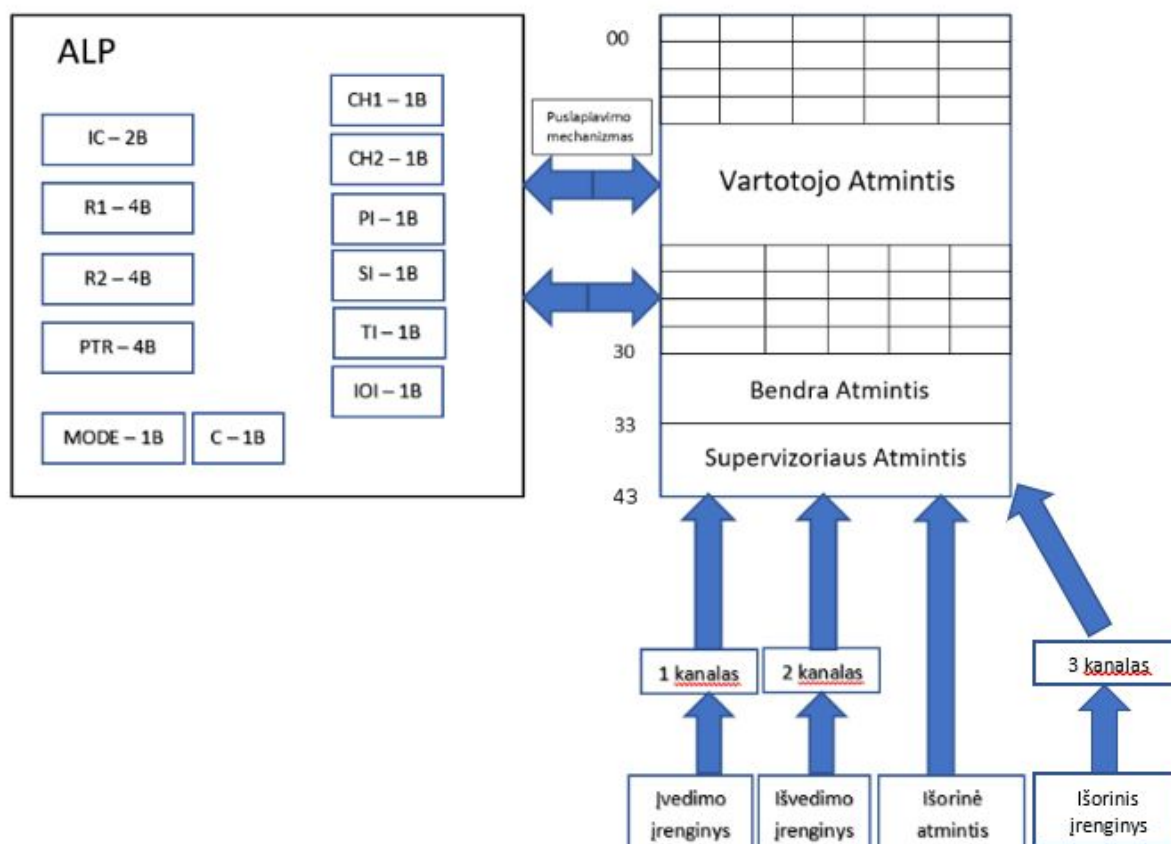
Virtualios mašinos procesoriaus komandos operuoja su duomenimis, esančiais registruose ir ar atmintyje. Yra komandos duomenų persiuntimui iš atminties į registrus ir atvirkščiai, aritmetinės (sudėties, atimties, palyginimo), sąlyginio ir besąlyginio valdymo perdavimo, įvedimo, išvedimo, darbo su bendra 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.

Realios mašinos projektas

Techninės įrangos komponentai



//kanalų įrenginys, 5 kanalai, kintamo dydžio atmintis (puslapiavimas)

Centrinis Procesorius

Paskirtis - skaityti komandą iš atminties ir ją vykdyti (interpretuoti). Procesorius gali dirbti dviem režimais: vartotojo ir supervizoriaus. Registras **MODE** nusako procesoriaus režimą (0 - vartotojo, 1 - supervizoriaus) :

- Vartotojo režimas - jame yra vykdoma paduota programa
- Supervizoriaus režimas - jame komandos iš supervizorinės atminties yra betarpiškai apdorojamos aukšto lygio kalbos procesoriaus (ALP).

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 bendro naudojimo registras.

[R2] - 4 baitų bendro naudojimo registras.

[PTR] - 4 baitų nuorodų registras skirtas puslapiavimui.

[PI] - 1 baito registras programiniams pertraukimams fiksuoti:

- 1 - pažeista atminties apsauga
- 2 - neatpažintas operacijos kodas
- 3 - dalyba iš 0
- 4 - Aritmetinės operacijos klaida

[SI] 1 baito registras supervizoriniams pertraukimams fiksuoti:

- 1 - Nuskaitymo komandos pertraukimas
- 2 - Išvedimo komandos pertraukimas
- 3 - Prietaiso prijungimo pertraukimas
- 4 - Prietaiso atjungimo pertraukimas
- 5 - HALT komandos pertraukimas

[TI] - 1 baito taimerio registras laikrodžio pertraukimams fiksuoti.

[IOI] - 1 baito įvedimo / išvedimo pertraukimo registras – praneša, kad veiksmas baigėsi.

1 - pirmas kanalas, 2 - antras kanalas

[MODE] – 1 baito registras, kurio reikšmė nusako procesoriaus darbo režimą. 0 - dirbama vartotojo režimu, 1 - dirbama supervizoriaus režimu.

[C] - 1 baito požymių registras. Lentelėje pateikta informacija apie logines reikšmes:

Bitas	Trumpinys	Reikšmė	Paaiškinimas
1	ZF (Zero flag)	Nulio požymis	ZF=1, jei reikšmių skirtumas yra 0, ZF=0 kitu atveju.
2	SF (Sign flag)	Ženklo požymis	SF = 1, jeigu atlikus aritmetinę operaciją ženklo bitas yra 1. Kitu atveju SF = 0
3	OF (Overflow flag)	Perpildymo požymis	OF = 0, jeigu dviejų reikšmių ženklo bitai yra vienodi, o jų skirtumo ženklo bitas skiriasi. Priešingu atveju OF = 1

Naudotojo atmintis

Atmintis, skirta virtualių mašinų atmintims bei puslapių lentelėms saugoti. Iš realios mašinos bendros atminties, turinčios 1088 žodžius (44_{16} blokai po 16 žodžių), naudotojo atminčiai bus išskirti 768 žodžiai, t.y. 30_{16} blokų po 16 žodžių. Žodžio ilgis - 4 baitai.

Supervizorinė atmintis

Atmintis, skirta operacinės sistemos poreikiams, laikanti sisteminius procesus, kintamuosius, resursus ir t.t. Realizuojama aukšto lygio programavimo kalba. Ji užims 16 blokų - viso 256 žodžius.

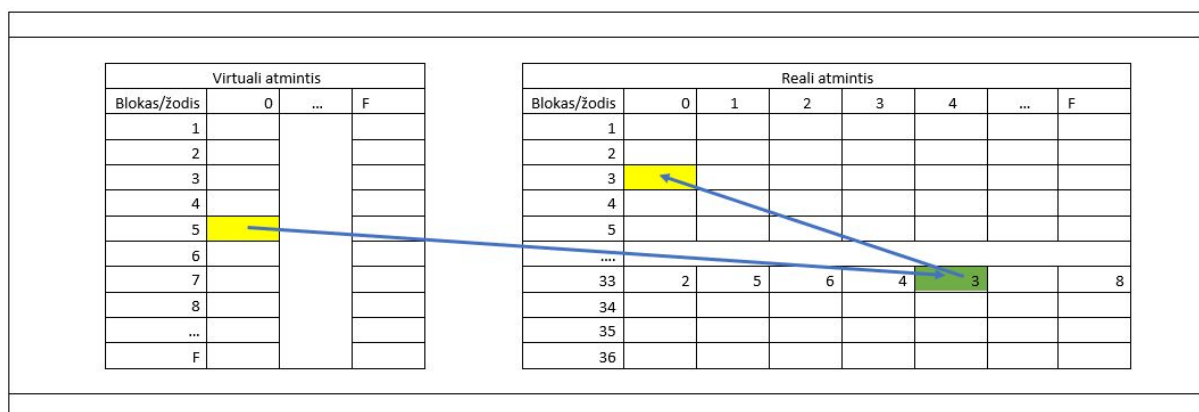
Bendra atmintis

Skirta virtualių mašinų valdymui bei tvarkymui. Jai bus išskirti 3 blokai po 16 žodžių - viso 48 žodžiai.

Puslapiavimo Mechanizmas

Sąryšis tarp virtualios atminties ir realios atminties bus palaikomas per puslapiavimo mechanizmą. Virtualios mašinos puslapi(bloka) sudaro 16 žodžių. Registre [PTR] bus saugomas einamosios puslapių lentelės ilgis bei adresas. Keturi [PTR] baitai **a0, a1, a2, a3** bus paskirstyti taip:

- a0 - bus naudojamas puslapio dydžiui
- a1 - puslapių lentelės ilgis - 1
- a2 - naudojamas kartu su a3 gauti puslapių lentelės adresą pagal: $a2 * 16 + a3$
- Dviženklis adresas [x1, x2] virtualioje adresinėje erdvėje aparatūriškai atvaizduojamas į realų adresą vartotojo atmintyje: $16 * [16 * (16 * a2 + a3) + x1] + x2$



Išorinė atmintis

Tai tiesioginio įėjimo ir išėjimo išorinis įrenginys, kuriame duomenys sugrupuoti į failus (failo pavadinimas yra mašinos žodis). Reali mašina gali atidaryti skaitymui. Iš failo informacija

užkraunama į supervizorinę atmintį. Tam, kad failai išorinės atminties įrenginyje būtų atpažinti realios mašinos, jie bus aprašomi šia struktūra:

```
SOF <failo_pavadinimas>  
<komanda>  
.....  
<komanda>  
EOF
```

SOF (start of file) - raktažodis skirtas atpažinti failo pavadinimą.

EOF (end of file) - raktažodis skirtas atpažinti failo pabaigą.

Duomenų perdavimo kanalai

- **[CH1]** perduodami duomenys iš įvedimo įrenginio (klaviatūros) į supervizorinę atmintį.
- **[CH2]** perduodami duomenys iš supervizoriaus atminties į išvedimo įrenginį (ekraną).
- **[CH3]** perduodami duomenys iš/į supervizoriaus atminties į/iš išorinį įrenginį

Visi kanalai centriniame procesoriuje turi atitinkamus 1 baido registrus **CH1**, **CH2**. 0 - kanalas laisvas, 1 - kanalas užimtas.

Įvedimo ir išvedimo įrenginiai

Įvedimo įrenginys - klaviatūra, išvedimo įrenginys - ekranas. Abu turi 16 žodžių dydžio buferius apsikeisti duomenimis. Užpildžius buferį, jis yra siunčiamas į išvedimo įrenginį arba į kanalą priklausomai nuo situacijos.

Taimerio Mechanizmas

Atsakingas už geresnį užduočių išlygiagretinimą. Laikoma, kad ta pati užduotis negali būti vykdoma ilgiau nei N laiko momentų. Laikome, kad įvedimo/išvedimo instrukcijos atliekamos per 3 taktus, visos kitos per 1 taktą. Pradedant virtualios mašinos užduoties vykdymą, TI registro reikšmė yra nustatoma $M = 10$. Įvykdžius eilinę instrukciją, TI reikšmė mažinama priklausomai nuo to per kiek taktų ta instrukcija yra atliekama. Kai TI reikšmė tampa 0, mikrokomanda aptinka taimerio pertraukimą.

Realios Mašinos Komandos

Tam, kad būtų galima dirbti su išorinės atminties įrenginiu, reali mašina turės šias komandas:

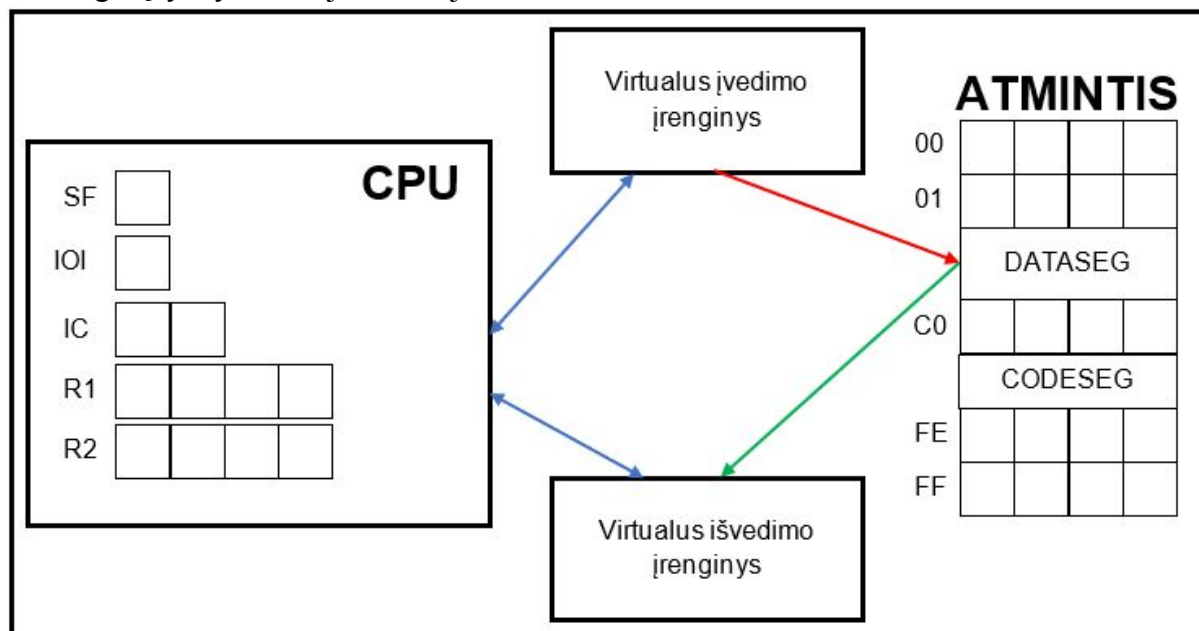
- **START** <failo_pavadinimas> - bus paleidžiamas/vykdomas failas su nurodytu pavadinimu.
- **SHUTDOWN** - bus išjungiamas/baigimas darbas.

Reali mašina bus laukimo būsenoje tol, kol bus gauta komanda dirbti su failu. Pabaigusi darbą ji vėl grīš ī laukimo būsena.

Virtualios mašinos projektas

Virtualios mašinos aprašymas

Virtuali mašina – tai virtuali realios mašinos kopija. Virtualioje mašinoje surenkame mums reikalingas realios mašinos komponentes, tokias kaip procesorius, atmintis, įvedimo/išvedimo įrenginiai, suteikiame joms paprastesnę naudotojo sąsają. Tuo pačiu palengvinamas programavimo procesas - sudėtingos ar naudotojui nepatogios sąsajos virtualioje mašinoje yra supaprastinamos. Virtuali mašina realizuoja realios mašinos komandas paprastesniu, lengviau suprantamu būdu, interpretuojant virtualios mašinos komandas kaip realios mašinos komandas ar jų rinkinį. Taip pat virtuali mašina pateikia supaprastintą atminties adresavimą. Visa tai leidžia pasiekti realią mašiną ir virtualios mašinos mašininiu kodu parašytą programą sėkmingai įvykdyti realioje mašinoje.



CS,DS 1B

Virtualios mašinos komponentai

Atmintis – VM atmintis yra sudaryta iš 16 blokų po 16 žodžių (4 baitai – 32 bit) ir virtualiai mašinai yra paskiriama jos inicijavimo (programos vykdymo) metu. Kitais žodžiais – visa programa turi tilpti į 256 žodžius (1KB). Atminties blokai yra padalinami į dvi dalis – duomenis (dalis nuo DATA iki CODE – 12 blokų) ir vykdomąją programos kodą (dalis nuo CODE iki privalomos komandos EOF – 4 blokai). Virtuali mašina gali keisti tik duomenų dalyje esančią informaciją (programos darbui reikalingi duomenys), o vykdyti tik kodo dalyje esančias komandas. Tiek kodo, tiek duomenų dalių adresai skaičiuojami nuo 0. Ryšiai tarp realaus ir virtualaus adreso bus nusakomi per puslapiavimo mechanizmą.

Žodis Blokas	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

Procesorius – VM procesoriaus pagrindinė paskirtis - vykdyti programą, kuri yra virtualioje atmintyje. Kiekvienas procesas turi savo virtualų centrinį procesorių (kaip ir atmintį).

Virtualius procesorius turės tik procesai – VM. Virtualus procesorius turės 5 registrus:

- IC – 2B virtualios mašinos komandų skaitiklis;
- SF – 1B VM požymių registras. Baitas forma **0100SZC1**. Veikia kaip realioje mašinoje:
 - 1bit = CF – pernešimo pož. – 1, kai ADD arba SUB rezultatas netelpa į žodį;
 - 2bit = ZF – nulio pož. – 1, kai paskutinės operacijos rezultatas yra nulinis;
 - 3bit = SF – ženklo požymis – 1, kai paskutinės operacijos rezultatas yra neigiamas
 - 4bit = OF – perpildymo požymis – 1, kai rezultatas netelpa į skaičių su ženklu diapazoną;
- IOI – 1B įvedimo/išvedimo ir pertraukimų registras – įrašoma, kuris kanalas naudojamas
- R1 ir R2 – 4B bendro naudojimo registrai

Virtualios mašinos komandos

Atmintyje esantis 4 baitų žodis gali būti traktuojamas kaip duomenys arba kaip komanda. Ne visi komandos baitai privalo būti užimti, t.y., jie gali būti tušti (pvz.: komanda ADD).

Komandas skirstysime į kelias grupes:

Darbo su duomenimis komandos:

- LXxy – į registrą R1 nukopijuojamas atminties adresu $16 \cdot x + y$ turinys
 $R1 = [16 \cdot x + y]$
- LW - į registrą R1 nukopijuojamas atminties adresu [IC]+1 turinys (vieną žodį)
- LRx – į registrą Rx nukopijuojamas kito registro turinys
pvz.: LR1 -> R1 := [R2]
- SXxy – į atmintį adresu $16 \cdot x + y$ nukopijuojamas registro R1 turinys
 $[16 \cdot x + y] = [R1]$
- Izxy – veikimas priklauso nuo baito z:
z=0: iš STDIN į atmintį adresu $16 \cdot x + y$ patalpinamas vienas žodis
z=1: iš STDIN į atmintį nuo adreso $16 \cdot x$ patalpinama y žodžių
z=2: į STDOUT iš atminties adresu $16 \cdot x + y$ išvedamas vienas žodis
z=3: į STDOUT iš atminties pradedant adresu $16 \cdot x$ išvedama y žodžių
z=4: į STDOUT iš atminties pradedant adresu $16 \cdot x$ išvedami baitai, kol sutinkamas simbolis '\$' (baitas 24)
zx=R0: iš STDIN į registrą R1 patalpinamas vienas žodis
zx=R1: į STDOUT iš registro R1 išvedamas vienas žodis

Aritmetinės komandos:

- ADD – Sudedamos registrų R1 ir R2 reikšmės ir rezultatas įdedamas į registrą R1.
Atitinkamai formuojami SF požymiai
 $R1 = [R1] + [R2]$
- SUB – Registrų R1 ir R2 reikšmių skirtumo rezultatas įdedamas į registrą R1.
Atitinkamai formuojami SF požymiai
 $R1 = [R1] - [R2]$
- ADxy – Sudedamos registrų R1 ir R2 reikšmės ir rezultatas patalpinamas DS adresu $16 \cdot x + y$. Atitinkamai formuojami SF požymiai
 $[16 \cdot x + y] = [R1] + [R2]$
- SBxy - Registrų R1 ir R2 reikšmių skirtumo rezultatas patalpinamas į DS adresu $16 \cdot x + y$. Atitinkamai formuojami SF požymiai
 $[16 \cdot x + y] = [R1] - [R2]$

Loginės komandos:

- AND - Atliekama registrų bitų loginė daugyba:
 $R1 = R1 \& R2$

- OR - Atliekama registrų bitų loginė sudėtis:
 $R1 = R1 \vee R2$
- NRx - Atliekamas loginis neigimas su registro Rx bitais:
 $Rx = \sim Rx$
- XOR - Atliekama registrų bitų loginė suma moduliu 2:
 $R1 = (R1 \wedge \sim R2) \vee (\sim R1 \wedge R2)$

Loginių komandų (išskyrus neigimą) rezultatas patalpinamas R1 registre.

Palyginimo komanda:

- COMP – palygina registrus R1 ir R2. Formuoja SF požymius:
 $R1 = R2 \Rightarrow ZF = 1$
 $R1 > R2 \Rightarrow ZF = 0 \& SF = 0$
 $R1 < R2 \Rightarrow ZF = 0 \& SF = 1$

Valdymo perdavimo komandos:

- JPxy – besąlyginis valdymo perdavimas kodo segmento žodžiui, kurio adresas atmintyje yra $16 \cdot x + y$
- JExy – Jei $ZF = 1$ (t.y. $R1 = R2$), valdymas perduodamas KS žodžiui adresu $16 \cdot x + y$
- JGxy – Jei $ZF = 0 \& SF = 0$ (t.y. $R1 > R2$), valdymas perduodamas KS žodžiui adresu $16 \cdot x + y$
- JLxy – Jei $ZF = 0 \& SF = 1$ (t.y. $R1 < R2$), valdymas perduodamas KS žodžiui adresu $16 \cdot x + y$
- JNxy – Jei $ZF = 0$ (t.y. $R1 \neq R2$), valdymas perduodamas KS žodžiui adresu $16 \cdot x + y$
- JCxy – Jei $CF = 1$, valdymas perduodamas KS žodžiui adresu $16 \cdot x + y$
- JOxy – Jei $OF = 1$, valdymas perduodamas KS žodžiui adresu $16 \cdot x + y$

Visos JUMP komandos įgyvendina (arba ne) operaciją $IC = [16 \cdot x + y]$

- LPxy - Jei $R2 > 0$, tai $R2 = [R2] - 1$ ir atliekama JPxy, kitu atveju programa tęsiama
- HALT – Užbaigiamas programos darbas

Įvedimo ir išvedimo komandos:

- PDx – SI tampa 1 ir valdymas perduodamas OS, duomenų kopijavimui iš kietojo disko į supervalizorinės atminties vietą x. Išsiunčia išvedimui 16 žodžių srautą iš atminties ląstelių $[16 \cdot x + i]$, kur $i = 0 \dots F$.
- PDxy - SI tampa 1 ir valdymas perduodamas OS, duomenų kopijavimui iš kietojo disko į supervalizorinės atminties vietą x. Išvedimui yra siunčiami y žodžių prasidedantys adresu $16 \cdot x$

- GDx - SI tampa 2 ir valdymas perduodamas OS duomenų kopijavimui į kietąjį diską iš supervizorinės atminties vietos x. Duomenys iš įvedimo srauto įrašomi į ląsteles prasidedančias adresu $16 \cdot x$ ir iš eilės yra įrašomi iki $16 \cdot x + 15$ adreso
- GDxy - SI tampa 2 ir valdymas perduodamas OS duomenų kopijavimui į kietąjį diską iš supervizorinės atminties vietos x. Skaitomi y žodžių ir jie talpinami pradedant adresu $16 \cdot x$
- MNTx - SI tampa 3 ir valdymas perduodamas OS duomenų skaitymui ir siuntimui iš ir į išorinį įrenginį. //x kanalas
- GLxy - SI tampa 2 ir valdymas perduodamas OS duomenų kopijavimui į atminį iš supervizorinės atminties vietos x. Nuskaitomas vienas žodis iš prijungto prietaiso ir patalpinamas adresu $16 \cdot x + y$ //į registrą
- SLxy - SI tampa 1 ir valdymas perduodamas OS duomenų siuntimui iš atminties adreso $16 \cdot x$ vienas žodis į supervizorinę atmintį adresu y. Duomenys siunčiami ir naudojami išorinio įrenginio valdymui.
- DMTx - SI tampa 4 ir valdymas perduodamas OS ir laukiamas patvirtinimas, kad įrenginys atjungtas. //x kanalas

Mūsų naudojamas išorinis įrenginys bus RGB šviestukas (LED). Vienas žodis reguliuos spalvas ir ryškumą (RGBX - RedGreenBlue).

//atskiras kodavimas Paskutinis baitas neatlieka jokios funkcijos, kai prijungtas įrenginys yra RGB LED. Jei kartu prijungiame piezoelektrinį garsiakalbį, paskutinis X baitas tampa atsakingas už skleidžiamą toną.

Semaforų komandos:

- BSM – Užblokuojama benra atmintis
- USM – Atblokuojama bendra atmintis

Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais mechanizmo aprašymas

Įvedimui naudojama klaviatūra, išvedimui - ekranas. Yra išorinės atminties įrenginys - kietasis diskas. Įvedimą/išvedimą kontroliuoja kanalų įrenginys.

Iš pat pradžių duomenys iš įvedimo įrenginio yra įrašomi į kietąjį diską. Tada iš kietojo disko duomenys yra patalpinami į supervizorinę atmintį apdorojimui. Analogiškai apdoroti duomenys iš atminties yra rašomi į kietąjį diską, o po to į išvedimo įrenginį. Darbai su kietuoju disku yra skirtos komandos PD ir GD

Virtualios mašinos interpretuojamojo ar kompiliuojamo vykdomojo failo išeities teksto formatas

Užduoties kodas yra surašomas faile. Paskutinė vykdoma komanda privalo būti HALT. Kitu atveju virtualios mašinos darbo rezultatai nėra apibrėžti.

Bendra programos schema:

DATA

.

.

.

CODE

.

.

.

EOF

Kodo pavyzdys:

```
DATA  
0001  
0005  
0010  
ATS:  
CODE  
LX00  
LY04  
ADD  
LY08  
SUB  
PD2  
HALT  
EOF
```

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

Sistemai aptikus pertraukimą, nutraukiamas vartotojo programos vykdymas, o valdymas perduodamas pertraukimo apdorojimo programai. Pastaroji programa, atlikusi darbą, valdymą grąžina operacinei sistemai pertraukimo vietoje. Ką toliau daryti – sprendžia OS. Kiekvieno pertraukimą sukeliančio įvykio metu bus nustatomi atitinkami pertraukimų registrai: CH1, CH2, SI, PI, TI priskiriamos atitinkamos reikšmės.

Į pertraukimą reaguojama tik vartotojo režime, supervizoriaus režime pertraukimai uždrausti. Vykdomas programos vartotojo režimu virtualioje mašinoje, vykdomas komandos interpretavimo ciklas ir po to duodama supervizorinė komanda Test(), kuri nustato ar neįvyko pertraukimas.

Pertraukimui įvykus, išsaugoma virtualios mašinos būseną, procesorius perjungiamas į supervizoriaus režimą ir valdymas atiduodamas atitinkamo pertraukimo apdorojimo programai.

Pertraukimo programos nustato pertraukimą sukėlusią priežastį ir išvalo pertraukimo registrą, kviesdama komanda Test su atitinkamu parametru. Po to reaguojant į pertraukimą. Bendra taisyklė pertraukimų programų bendravimui su likusia sistemos dalimi turėtų būti tokia: duomenimis keičiamasi ne tiesiogiai, o per sutartus kintamuosius supervizorinėje atmintyje, kurių adresai yra iš anksto žinomi ir nesikeičia. Tai būtina, norint maksimaliai atriboti žemo lygio pertraukimų apdorojimo aparatūra nuo aukšto lygio operacinės sistemos.

Trumpas aprašas, kaip virtuali mašina įsivaizduojama visos operacinės sistemos kontekste.

Virtuali mašina yra realios kopija, kuri veikia kaip tarpininkas. Virtuali mašina paslepia realios mašinos realizaciją po virtualiais komponentais. Ji turi savo atmintį, kur kiekvienas blokas turi

tiek virtualų tiek realų adresą. Ryšiai tarp šių adresų nusakomi puslapių lentelėmis. Virtuali mašina supaprastina vartotojo sąsają ir vykdo programą, kuri yra virtualioje atmintyje. Taigi OS – tai programa, kuri modeliuoja kelių virtualių mašinų darbą vienoje realioje mašinoje.