**Kurso „Operacinės sistemos“ pirmasis darbas**

**Virtualios ir realios mašinų projektai**

Darbą parengė:

Rokas Urlakis

Ernestas Zdanys

Marius Žilinskas

Pratybų dėstytojas:

Liutauras Ričkus

[1. Užduoties sąlyga 3](#_Toc414098660)

[2. Reali mašina 4](#_Toc414098661)

[2.1 Realios mašinos modelis 4](#_Toc414098662)

[2.2 Realios mašinos procesorius 5](#_Toc414098663)

[2.3 Realios mašinos atmintys 5](#_Toc414098664)

[2.4 Taimeris 6](#_Toc414098665)

[2.5 Pertraukimai 6](#_Toc414098666)

[3. Virtuali mašina 7](#_Toc414098667)

[3.1 Virtualios mašinos modelis 7](#_Toc414098668)

[3.2 Virtualios mašinos procesorius 7](#_Toc414098669)

[3.3 Virtualios mašinos atmintis 8](#_Toc414098670)

[3.4 Virtualios mašinos procesoriaus komandos 8](#_Toc414098671)

[4. Puslapiavimo mechanizmas 11](#_Toc414098672)

[5. Užduoties pateikimo forma 11](#_Toc414098673)

# Užduoties sąlyga

Užduotis E.

Projektuojama paketinė operacinė sistema:

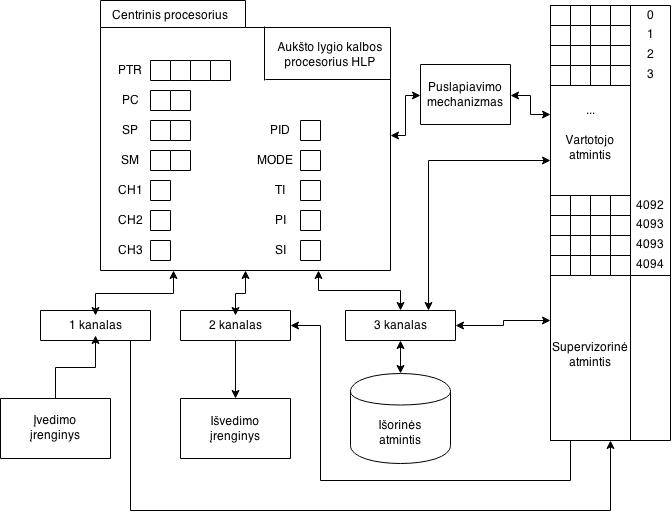
* Virtualios mašinos procesoriaus komandos operuoja su duomenimis, esančiais steko viršūnėje.
  + Yra komandos duomenų persiuntimui iš atminties į steką ir atvirkščiai
  + Aritmetinės (sudėties, atimties, daugybos, dalybos)
  + Sąlyginio ir besąlyginio valdymo perdavimo
  + Įvedimo
  + Išvedimo
  + Šakojimo (sukuria proceso kopiją; pagal jos grąžinamą reikšmę procesas, nustato, ar jis yra tėvinis, ar vaikinis)
  + Programos pabaigos komandos
  + Registrai yra du:
    - Komandų skaitiklio
    - Steko viršūnės
  + Atminties dydis yra 16 blokų po 16 žodžių (žodžio ilgį pasirinkite patys)
* Realios mašinos procesorius gali dirbti dviem režimais:
  + Vartotojo
  + Supervizoriaus
  + Yra taimeris, kas tam tikrą laiko intervalą generuojantis pertraukimus
  + Įvedimui naudojamas virtualių „flash atmintinių“ nuskaitymo įrenginys
  + Išvedimui naudojamas spausdintuvas
  + Yra išorinės atminties įrenginys - kietasis diskas
* Virtualios mašinos atmintis atvaizduojama į vartotojo atmintį naudojant puslapių transliaciją
* Vartotojas užduočių paketą pateikia „prijungęs“ atmintinę.
  + Sistema perkelia visas joje esančias užduotis į diską, patikrindama jų sintaksę, ir, tuo pat metu, jei tik yra reikiamų resursų, pradeda jas vykdyti.

# Reali mašina

Reali mašina – kompiuteris. Apibrėžiame realios mašinos modelį, kurį sudaro:

1. Procesorius
2. Vartotojo atmintis
3. Išorinė atmintis
4. Supervizoriaus atmintis
5. Įvedimo įrenginys („flash“ atmintinių nuskaitymo įrenginys)
6. Išvedimo įrenginys

## Realios mašinos modelis



## Realios mašinos procesorius

Procesorius skaito komandą iš atminties ir ją vykdo. Jis gali dirbti vartotojo ir supervizoriaus režimais.

* Vartotojo režime HLP (aukšto lygio kalbos procesorius) vykdo jai skirtą programą imituojant VM (virtualios mašinos) procesorių ir naudojant puslapiavimo mechanizmą norint naudoti vartotojo atmintį.
* Supervizoriaus režime HLP betarpiškai apdoroja komandas esančias supervizorinėje atmintyje.

Realios mašinos registrai:

* PTR – 4 baitų puslapių lentelės registras
* PC – komandų skaitliukas
* SP – steko viršūnės indeksas
* SM – registras rodantis į bendrą atmintį
* PID – 1 baito proceso numeris
* MODE – registras apibūdinantis darbo režimą
* TI – taimerio registras
* PI – programinių pertraukimų registras
* SI – supervizorinių pertraukimų registras
* CH1, CH2, CH3 – kanalų registrai (0 – atidarytas, 1 – uždarytas)

## Realios mašinos atmintys

RM (reali mašina) turi trijų tipų atmintis: vartotojo, supervizorinę ir išorinę.

Vartotojo atmintyje saugomos puslapių lentelės ir informacija skirta virtualioms mašinoms.

Supervizorinė atmintis skirta operacinei sistemai.

Išorinė atmintis - kietasis diskas. Jame galime saugoti bet kokią informaciją.

## Taimeris

Šis mechanizmas skirtas užduotims suderinti. Viena užduotis negali būti vykdoma ilgiau nei n laiko momentų. Apibrėžiame, kad informacijos įvedimo ir išvedimo operacijos reikalauja 3 taktų, kitos operacijos reikalauja 1 takto.

Tarkime, kai virtuali mašina pradeda darbą į registrą TI įrašoma tam tikra reikšmė. Įsivaizduokime, kad užduotis vykdoma daugiausiai n = 20 laiko momentų. Po kiekvienos įvykdytos operacijos TI registro reikšmė yra sumažinama per 1 ar 3 taktus (priklausomai nuo operacijos rūšies). TI registro reikšmei pasiekus 0 įvyksta pertraukimas.

## Pertraukimai

Pertraukimas – tam tikras signalas apie įvykusį įvykį.

Pertraukimus aptinka komanda Test().

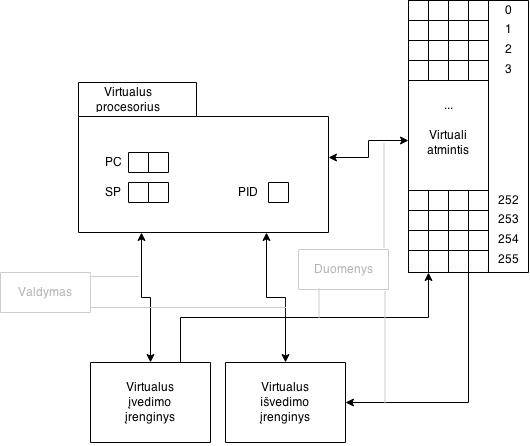
Išskirsime trijų rūšių pertraukimus:

* Taimerio (TI):
  + TI = 0 – taimerio skaitliukas lygus nuliui
* Programiniai (PI):
  + PI = 1 – neteisingas adresas
  + PI = 2 – blogas operacijos kodas
  + PI = 3 – negalimas priskyrimas
  + PI = 4 – perpildymas
* Supervizoriniai (SI):
  + SI = 1 – komanda PRTS
  + SI = 2 – komanda PRTN
  + SI = 3 – komanda P
  + SI = 4 – komanda READ
  + SI = 5 – komanda STOPF
  + SI = 6 – komanda FOxy

# Virtuali mašina

VM (virtuali mašina) – realios mašinos modelis, kuris veikia lyg tarpininkas. Pagrindinė VM paskirtis – vykdyti vartotojo programą

## Virtualios mašinos modelis



## Virtualios mašinos procesorius

Virtualios mašinos procesorius yra kur kas paprastesnis. VM procesoriaus paskirtis – vykdyti programą, kuri yra virtualioje atmintyje. Kiekvienas virtualus procesas turi savo virtualų procesorių. Virtualus procesorius turi tris registrus:

* PC – 2 baitų komandų skaitliukas
* SP – 2 baitų steko viršūnės indeksas
* PID – 1 baito proceso numeris

Visų registrų pradinė reikšmė lygi 0.

## Virtualios mašinos atmintis

Kiekvienai virtualiai mašinai yra skiriama 16 vartotojo atminties blokų. Tuose šešiolikoje blokų (256 žodžių po 4 baitus) turi tilpti programos kodas, duomenys ir stekas. Kiekvienas virtualios atminties blokas turi virtualų ir realų adresą. Virtualiais adresais operuoja virtuali mašina, realiais – reali mašina. Ryšiai tarp virtualaus ir realaus adreso nusakomi puslapių lentelėmis. Tai bus aptarta skyrelyje „puslapiavimo mechanizmas“.

Kaip jau minėta, virtualios mašinos atmintis bus skirstoma į tris dalis, į kurias bus įkeliamos atitinkamos programos dalys:

1. Duomenys – 7 blokai, nuo 0 iki 6 (112 žodžiai)
2. Programa – 7 blokai, nuo 7 iki 13 (112 žodžiai)
3. Stekas – 2 blokai, nuo 14 iki 15 (32 žodžiai)

## Virtualios mašinos procesoriaus komandos

* Aritmetinės komandos
  + ADD – sudeda du viršutinius steko elementus. Rezultatą padeda į steko viršūnę ir steko rodyklę sumažina vienetu.

[SP – 1] = [SP – 1] + [SP]; SP--;

* + SUB – atima steko viršūnėje esantį elementą iš antro nuo viršaus elemento. Rezultatą padeda į steko viršūnę ir steko rodyklę sumažina vienetu.

[SP – 1] = [ SP – 1] - [SP]; SP--;

* + MUL – sudaugina du viršutinius steko elementus. Rezultatą padeda į steko viršūnę ir steko rodyklę sumažina vienetu.

[SP – 1] = [ SP – 1] \* [SP]; SP--;

* + DIV – padalina (gauna sveikąją dalį) antrą nuo viršaus steke esantį elementą iš viršūnėje esančio. Rezultatą padeda į steko viršūnę ir steko rodyklę sumažina vienetu.

[SP – 1] = [ SP – 1] / [SP]; SP--;

* Komandos darbui su bendra atmintimi
  + WRx – į bendrą atmintį nurodytu adresu įrašo steko viršūnėje esantį žodį.
  + RDx – iš bendrosios atminties nurodytu adresu nuskaito žodį ir įdeda į steko viršūnę.
* Palyginimo komandos
  + CMP – lygina steko viršūnėje esančius du žodžius. Ir rezultatą padeda į steko viršūnę.

1 – jei lygūs, 0 – jei viršutinis mažesnis, 2 – jei didesnis.

[SP+1] = 0, jei [SP-1] > [SP]; SP--

[SP+1] = 1, jei [SP-1] == [SP]; SP--

[SP+1] = 2, jei [SP-1] < [SP]; SP—

* + CPID – lygina steko viršūnėje esantį vaiko PID su einamos mašinos registru PID

1 – jei lygūs, 0 – jei nelygūs

[SP] = 0, jei [SP] <> PID

[SP] = 1, jei [SP] == PID

* Darbo su duomenimis/steko komandos
  + LDxy – į steko viršūnę užkrauna reikšmę iš duomenų srities adresu

16 \* x + y, (0 ≤ x < 6), SP--;

* + PTxy – steko viršūnėje esantį žodį deda į duomenų sritį nurodytu adresu

SP++; 16 \* x + y, (0 ≤ x < 6)

* + PUNx – x kaip skaičių patalpina į steko viršūnę.

SP++;[SP] = x

* + PUSx – x kaip simbolį patalpina į steko viršūnę.

[SP] = x

* Valdymo komandos
  + JPxy – nesąlyginio valdymo perdavimo komanda. Valdymas perduodamas kodo sričiai nurodytam adresui.

PC = 16\*x+y

* + JExy – jei steko viršūnėje yra 1 valdymas perduodamas adresu 16\*x + y.

IF([SP] == 1) PC = 16 \* x + y; SP--;

* + JLxy – jei steko viršūnėje yra 0 valdymas perduodamas adresu 16\*x + y.

IF([SP] == 0) PC = 16 \* x + y; SP--;

* + JGxy – jei steko viršūnėje yra 2 valdymas perduodamas adresu 16\*x + y.

IF([SP] == 2) PC = 16 \* x + y; SP--;

* Įvedimo ir išvedimo komandos
  + PRTS – steko viršūnėje esantį žodį traktuoja kaip simbolius ir išveda į išvedimo įrenginį.
  + PRTN – steko viršūnėje esantį žodį traktuoja kaip skaitinę reikšmę ir išveda į išvedimo įrenginį.
  + Pxyz – į išvedimo įrenginį išveda x numeriu nurodyto atminties srities bloko nuo y iki z žodžius (y < z). (0 ≤ x < 6)

Print([16\*x+i, i=y..z])

* + READ – nuskaito vartotojo pateiktą programą ir duomenis ir patalpina į atitinkamus atminties segmentus.
* Šakojimo komandos
  + FOxy – sukuria proceso kopiją ir tolimesnį darbą paskiria vaikiniam procesui ir išsaugo vaiko PID duomenų segmente adresu 16 \* x + y. Vaiko PID++
  + STOPF – besąlyginio programos sustojimo komanda.
  + STxy – iš duomenų segmento adresu 16 \* x + y paima PID reikšmę, IF(PID <> 0) THEN [16 \* x + y] = 0 ELSE STOPF.

# Puslapiavimo mechanizmas

Realios mašinos vartotojo atmintis susidaro iš 256 blokų. Sukuriant naują virtualią mašiną jai yra skiriama 16 iš 256 blokų, kurie yra parenkami atsitiktinai.

Puslapiavimas veikia taip: kuriant naują virtualią mašiną jai reikia išskirti 16 atminties blokų. Atsitiktinai parenkame blokus su numeriais: 1, 3, 7, 12, 13, 25, 42, 69, 70, 72, 100, 101, 180, 181, 203, 249. Virtuali mašina šiuos blokus numeruoja nuo 0 iki 15, t. y. pirmas blokas yra nulinis, trečias – pirmas, 7 – antras, ... , 249 – penkioliktas. Norint išsaugoti ryšius tarp virtualios ir realios atminties reikalinga puslapių lentelė. Puslapių lentelė – tai vienas blokas esantis realioje atmintyje.

PTR – 4 baitų a0a1a2a3 registras, kurį simboliškai žymėsime, o virtualų adresą žymėsime x1x2.

* a0 – nenaudojamas
* a1 – nenaudojamas
* 16\*a2+a3 – puslapių lentelės bloko numeris vartotojo atmintyje
* 16\*(16\*a2+a3) – puslapių lentelės bloko adresas
* 16\*(16\*a2+a3) + x1 – bloko x1 adresas puslapių lentelėje. Jame saugomas bloko numeris į kurį atvaizduotas yra x1 blokas VM
* 16\*[16\*(16\*a2+a3) + x1] – VM bloko x1 realus bloko adresas
* 16\*[16\*(16\*a2+a3) + x1] + x2 – realus adresas atitinkantis virtualų adresą x1x2

# Užduoties pateikimo forma

Užduotį įsivaizduosime kaip atminties žodžių paketą sudarytą iš programos ir duomenų blokų.

Paketai pateikiami tokia tvarka:

* <paketas DATA>.<duomenys>.<paketas CODE>.<programa>.<paketas STOP>
  + <paketas DATA> turi vieną žodį DATA, kuris nurodo duomenų segmento pradžią
  + <duomenys> saugo informaciją, kurios dydis neturi viršyti 109 žodžių
  + <paketas CODE> turi vieną žodį CODE, kuris nurodo kodo segmento pradžią
  + <programa> saugo programos kodą, kuris užima iki 109 žodžių
  + <paketas STOP> turi vieną žodį STOP, kuris nurodo užduoties pabaigą