|  |
| --- |
| VU MIF 3 kursas, 3 grupė |
| OS Projektas |
| Reali ir virtuali mašinos |
|  |
| **Arūnas Bendoraitis** |
| **2/26/2011** |

|  |
| --- |
|  |

# Įvadas

Šiame projekte apžvelgiama ir pateikiama informacija, kuri buvo prašoma reikalavimuose:

1. Reali mašina
   1. Kokie techninės įrangos komponentai sudaro realią mašiną: procesorius, vartotojo atmintis, supervizorinė atmintis, išorinė atmintis, duomenų perdavimo kanalai, įvedimo/išvedimo įrenginiai.
   2. Realios mašinos techninės įrangos komponentų išsidėstymo vienas kito atžvilgiu ir tarpusavio sąveikos schema.
   3. Kiekvienas išskirtas techninis realios mašinos techninės įrangos komponentas turi būti detalizuojamas. Pavyzdžiui, aprašomos procesoriaus charakteristikos: kokios yra registrų grupės, kaip skaičiuojamas procesoriuje laikas, kaip nustatomas procesoriaus darbo režimas ir t.t.
2. Virtuali mašina
   1. Modeliuojamos virtualios mašinos sampratos aprašymas keliais sakiniais.
   2. Modeliuojamos virtualios mašinos loginių komponentų aprašymas. Privalomi komponentai:
      1. Virtualios mašinos atmintis.
      2. Virtualios mašinos procesorius.
      3. Virtualios mašinos komandų sistema. Komandos turi būti suskirstytos į grupes pagal paskirtį (aritmetinės, valdymo perdavimo ir t.t.).
   3. Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais mechanizmo aprašymas.
   4. Virtualios mašinos interpretuojamojo ar kompiliuojamo vykdomojo failo išeities teksto formatas. Pavyzdžiui, kaip išskiriamas duomenų segmentas, kodo segmentas, kaip aprašomi duomenys ir t.t.)
   5. Modeliuojamos virtualios mašinos loginių komponentų sąryšio su realios mašinos techninės įrangos komponentais aprašymas.
3. Trumpas aprašas, kaip virtuali mašina įsivaizduojama visos operacinės sistemos kontekste.

Užduoties sąlyga (G - projektuojama interaktyvi OS):

* 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,
  + darbo su bendra atminties sritimi (prieinama visoms vartotojo programoms; komandos leidžia į ją rašyti ir skaityti; sritis apsaugoma semaforais)
  + 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.
* Virtualios mašinos atmintis atvaizduojama į vartotojo atmintį naudojant puslapių transliaciją.
* Yra taimeris, kas tam tikrą laiko intervalą generuojantis pertraukimus.
* Input, Output
  + 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.

# Reali mašina

* 1. Reali mašina yra kompiuteris. Toliau nagrinėsime realią modelinę mašiną, kuri bus sudaryta tik iš esminių komponentų:
     1. Centrinis procesorius
     2. Atmintis
        + Vartotojo
        + Supervizoriaus
        + Išorinė (kietasis diskas)
     3. Įvedimo įrenginys (klaviatūra)
     4. Išvedimo įrenginys (ekranas)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| ...  ...  ...  ... | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* 1. Realios mašinos schema atrodys taip:

SM

CH3

CH2

CH1

PTR

SP

PC

4092

4093

000

TI

HLP

MODE

PI

SI

Centrinis procesorius

Puslapiavimo mechanizmas

002

003

001

4094

4095

Kanalų įrenginys

Supervizoriaus atmintis

Vartotojo atmintis

Išorinės atminties įrenginys

(kietasis diskas)

Išvedimo įrenginys

(ekranas)

Įvedimo įrenginys

(klaviatūra)

* 1. Plačiau apie komponentus.
     1. **Centrinis procesorius –** skaito komandą iš atminties ir ją vykdo (interpretuoja).

Gali dirbti dviem rėžimais: vartotojo arba supervizoriaus.

* + - * Vartotojo rėžime HLP vykdo tam tikrą užduoties programą. Yra imituojamas virtualios mašinos procesorius ir prieinama prie vartotojo atmintyje esančių programų per puslapiavimo mechanizmą.
      * Supervizoriaus rėžime komandos iš supervizinės atminties yra betarpiškai apdorojamos HLP.
        + HLP – bet kuris aukšto lygio kalbos procesorius (programavimo kalba).

Procesorius turi šiuos registrus:

* MODE – registras nusakantis darbo rėžimą.
* PTR – 4 baitų puslapių lentelės registras.
* SM – registras rodantis į bendrą atmintį.
* PC – komandų skaitliukas.
* SP – registras saugantis steko viršūnės žodžio indeksą.
* TI – timerio registras.
* PI – programinių pertraukimų registras.
* SI – supervizorinių pertraukimų registras.
* CH1, CH2, CH3 – kanalų registrai. Jų reikšmės 0 – kanalas laisvas, arba 1 – kanalas užimtas.
  + 1. **Atmintis –** turime trijų rūšių atmintį:
       - Vartotojo – skirta virtualių mašinų atmintims bei puslapių lentelių laikymui.
       - Supervizorinė – trumpai tariant tai atmintis, kurios reikia pačios OS poreikiams (komandos, sisteminiai kintamieji ir pan.). Visa tai valdys HLP.
       - Išorinė – šiuo atveju tai bus kietasis diskas. Jame gali būti koks failas.
       - Bendra – kiekvienai virtualiai mašinai yra iškirta realios mašinos atmintis prieinama tik konkrečiai virtualiai mašinai. Taip pat išskirsime bendrai naudojamos atminties 1 bloką prieinamų visoms virtualioms mašinoms. Tai galėtų atrodyti taip:

|  |
| --- |
| 2 VM  1 VM  SM  1 |
| 2 |
| ... |
| Bendra atmintis |

* + 1. **Kanalų įrenginys –** skirtas darbui su atmintimi ir įvedimo/išvedimo įrenginiams valdyti. Jie reikalingi tam, kad centrinis procesorius būtų išlaisvintas nuo lėtų išvedimo ir įvedimo įrenginių. Yra trys kanalai, kuriais „bendraujama“:
       - 1 – jungia klaviatūrą su vartotojo atmintimi.
       - 2 – jungia ekraną su vartotojo atmintimi.
       - 3 – jungia kietąjį diską su supervizorine bei vartotojo atmintim.
    2. **Puslapiavimo mechanizmas**Virtualiai mašinai yra išskiriama 16 blokų (arba puslapių, bet toliau blokų). Tuos blokus kiekviena VM susinumeruoja nuo 0 iki 15, tačiau jai taip pat reikia žinoti bei realius šių blokų adresus. Tam yra naudojamas puslapiavimo mechanizmas.  
         
       Taigi realius bloko numerius talpinsime į puslapių lentelę. VM puslapių lentelei bus išskiriamas 1 blokas vartotojo atminty. Lentelėje kiekvieno žodžio eilės numeris atitiks VM bloko numerį ir jame bus laikomas realus to bloko numeris.  
         
       Reikia nepamiršti, kad puslapių lentelė taip pat saugoma atmintyje, ir kad VM žinoti kur yra būtent jos puslapių lentelėje naudosime registrą PTR.  
         
         
         
         
       PTR yra 4 baitų ir simboliškai žymėsime taip a0a1a2a3. O x1x2 virtualius adresas.
       - 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.
    3. **Taimeris** **–** skirtas užduotims suderinti. Yra sakoma, kad užduotis negali trukti ilgiau nei tam tikrą T laiko momentų. Susitarsime, jog išvedimo / įvedimo operacijos reikalauja 3 laiko momentų, o kitos 1 ar 2.  
         
       Taigi kai VM pradeda darbą, speciali supervizorinės atminties ląstelė TI yra nustatoma tam tikrai reikšmei. Tarkime 50, tuomet kai yra įvykdoma instrukcija, TI yra mažinamas priklausomai nuo to kiek laiko momentų reikia instrukcijai. Kai TI tampa lygus nuliui, mikrokomanda Test() aptinka taimerio pertraukimą.  
         
       TI reikšmę galima nustatyti ar pakeisti supervizoriaus rėžime.
    4. **Pertraukimai** – tai tam tikri signalai apie specialius įvykius.   
       Gali būti aptikti tik vartotojo rėžime. Jam įvykus VM registrų reikšmės išsaugomos ir procesorius perjungiamas į supervisoriaus rėžimą, kuriame nustatomas pertaukimo pobūdis bei kviečiama pertraukimą apdorojanti programa.  
       Vėliau valdymas grįžta atgal į VM, vartotojo rėžimą ir atstatomi visi registrai.  
         
       Pertaukimusaptinka komanda Test().  
         
       Išskirsime trijų rūšių pertraukimus:
       - Programiniai, kurių registras yra PI. Galimi atvejai:
         * PI = 1 – atminties apsaugos pažeidimas.
         * PI = 2 – blogas operacijos kodas.
         * PI = 3 – neteisingas priskyrimas.
       - Supervizoriniai, kurių registras SI. Galimi atvejai:
         * SI = 1 – komanda PRTS
         * SI = 2 – komanda PRTN
         * SI = 3 – komanda P
         * SI = 4 – komanda READ
         * SI = 5 – komanda STOP
         * SI = 6 – komanda LC.
         * SI = 7 – komanda UC.
       - Taimerio, kurio registras TI. Galimi atvejai:
         * TI = 0 – taimerio skaitliukas lygus 0.

# Virtuali mašina

1. Virtuali mašina (VM) tai realios mašinos modelis, kuris veikia kaip tam tikras tarpininkas. Ji smarkiai supaprastina tiek ir programų rašymą tiek ir pačią realizaciją. VM pagrindinė paskirtis vykdyti vartotojo programą.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| ...  ...  ...  ... | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Virtualios mašinos schema atrodys taip:

000

Virtuali atmintis

Virtualaus Įvedimo įrenginys

Virtualaus išvedimo įrenginys

Virtualus procesorius

PC

SP

252

003

253

254

002

255

001

Plačiau apie komponentus:

* 1. **VM atmintis** susideda iš 256 žodžių. Kiekvienas žodis - 4 baitai. Bloką sudarys 16 žodžių. Taigi turime 16 blokų po 16 žodžių.  
     Atmintis bus suskirstyta į tris sritis į kurias bus įkeliamos atitankamos programos dalys parašytos VM programoje:
     + Duomenys – 112 žodžiai (7 blokai, nuo 0-6) [tarp DATA ir CODE].
     + Kodas – 112 žodžiai (7 blokai, nuo 7-13) [tarp CODE ir užbaigimo komandos STOP].
     + Stekas – 32 žodžiai (2 blokai, nuo 14-15).

VM atmintį simboliškai galime pavaizduoti taip:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| ... | | | |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| ... | | | |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

0

6

5

1

7

8

12

13

14

15

Duomenys

Kodas

SP - Stekas

**112 žodžiai**

**112 žodžiai**

**32 žodžiai**

* 1. **VM procesorius** yra gerokai supaprastintas. Jo pagrindinė paskirtis vykdyti programą kuri yra virtualioje atmintyje.  
     Procesoriaus registrai:
     + SP - registras saugantis steko viršūnės žodžio indeksą.
     + PC – komandų skaitliukas.

1. **VM komandų sistema.**
   * + Aritmetinės:
       - 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 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--;*
     + Darbo su bendra atmintim
       - WRx – į bendrą atminty nurodytu adresu įrašo steko viršūnėje esantį žodį.
       - RDx – iš bendrosios atminties nurodytu adresu nuskaito žodį ir įdeda į steko viršūnę.
       - LCx – užblokuojama bendros atminties žodis x.
       - UCx – atblokuojama bendra atmintis žodis x.
     + Palyginimo
       - CMP – lygina steko viršūnėje esančius du žodžius. Ir rezultutą 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--*
     + Darbo su duomenimis / steko
       - LDxy – į steko viršūnę užkrauna reikšmę iš duomenų srities adresu 16 \* x + y. (x < 7), SP--;
       - PTxy – steko viršūnėje esantį žodį deda į duomenų sritį nurodytu adresu *SP++;* 16 \* x + y. (x < 7)
       - PUNx – x kaip skaičių patalpina į steko viršūnę.  
         *SP++;[SP] = x*
       - PUSx – x kaip simbolį patalpina į steko viršūnę.  
         *[SP] = x*
     + Valdymo
       - JPxy – nesąlyginio valdymo perdavimo komanda. Valdymas perduodamas kodo sričiai nurodytam adresui.  
         *PC = 16\*x+y*
       - JExy – jei steko viršūnėje yra 1valdymas perduodamas adresu 16\*x + y.  
         *IF( [SP] == 1) PC = 16 \* x + y; SP--;*
       - JLxy – jei steko viršūnėje yra 0valdymas perduodamas adresu 16\*x + y.  
         *IF( [SP] == 0) PC = 16 \* x + y; SP--;*
       - JGxy – jei steko viršūnėje yra 2valdymas perduodamas adresu 16\*x + y.  
         *IF( [SP] == 2) PC = 16 \* x + y; SP--;*
       - STOP – programos sustojimo komanda.
     + Įvedimo bei 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). (x < 7)  
         *Print([16\*x+i, i=y..z])*
       - READ – nuskaito vartotojo įvedimą kaip skaičių ir įrašo į steko viršūnę.
2. **Bendravimas su įvedimo / išvedimo įrenginiais.**Vartotojo užduoties įvedimo / išvedimo operacijos atliekamos supervizoriaus rėžime.  
   Tam yra speciali komanda StartIO(..) , ji kaip vieną iš argumentų ima kanalo numerį.  
   Kanalų užimtumas nustatomas pagal kanalų (CH1, CH2, CH3) registrus. Jei kanalas užimtas procesorius laukia kol jis atsilaisvins.
   * + Įvedimo įrenginys – klaviatūra. Turi 16 žodžių buferį. Užsipildžius buferiui jo turinys perkialiamas į atmintį. Tačiau jei buferis nebuvo pilnai užpildytas, visi neužpildyti baitai užpildomi sutartiniu simboliu ‚$‘ - reiškiančiu srauto pabaigą.  
       Įvedimas vyksta 1 kanalu.
     + Išvedimo įrenginys – ekranas. Turi 16 žodžių buferį. Spausdina jį perduodant informaciją iš atminties 2 kanalu.
3. **VM vykdymo failas**. Programos failas turi būti tam tikros struktūros:

- po DATA raktinio žodžio eina duomenys. Duomenims aprašytnaudojamos šios „komandos“:

* DW X – išskiriams žodis į kurį talpinama skaitinė reikšmė X
* DD dddd – išskiriamas žodis į kurį talpinami nurodyti simboliai
* DN – talpinama simbolinė konstanta \n

DATA

CODE

STOP

...

...

# Apibedrinimas

Virtuali mašina yra realios kopija, kuri veikia kaip tarpininkas. Viruali 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.  
  
Bendrai galėtume pavaizduoti tokia schema:

Reali mašina

Virtuali mašina

Virtuali mašina

...

# Literatūra

1. A. Mitašiūno „Operacinių sistemų“ konspektai.
2. G.Šiaulio magistro darbas „Mokomoji operacinė sistema“.
3. Internetas.
4. Andrew S. Tanenbaum and Albert S. Woodhull „Operating Systems Design and Implementation“.