Operacinės sistemos

Realios ir virtualios mašinos projektas

Darbą atliko: Algirdas Ambrozaitis

Matematikos ir informatikos fakultetas

Matematinė informatika

**Virtualios mašinos modelis**



**Virtualaus procesoriaus registrai**

* R – bendro naudojimo registras
* PTR – 4 baitu puslapių lentelės registras
* C – požymio registras
* IC – 2 baitu komandų skaitliuko registras

**Virtualios mašinos komandos**

* LRxy - atminties ląstelės, kurios adresas x\*10 + y turinio kopijavimas į registrą R. Trumpiau: R:= [x\*10 + y]
* SRxy - registro R reikšmė įrašoma į atminties ląstelę, kurios adresas x\*10 + y. Trumpiau: [x\*10 + y]:= R
* ADxy - prie registro R reikšmės pridedama atminties ląstelės, kurios adresas x\*10 + y, reikšmė. Trumpiau: R:= R + [x\*10 + y]
* CRxy - jei R = [x\*10 + y], tai registrui C priskiriama reikšme „T”, kitu atveju C := „F“
* RLxy – jei R < [x\*10 + y], tai registrui C priskiriama reikšme „T”, kitu atveju C := „F“
* RGxy – jei R > [x\*10 + y], tai registrui C priskiriama reikšme „T”, kitu atveju C := „F“
* BTxy - jei registro C reikšmė lygi „T”, tai registrui IC priskiriama reikšmė [x\*10 + y]
* GDxy - iš įvedimo srauto perskaito 10 žodžių ir įrašo juos į lasteles [x \*10 + y]. Formalus užrašymas:
* For i:=0 to 9 do

[x \*10 + y]:= Buffer[i]; y:=y+1;

Next i

Čia Buffer – 10 žodžių masyvas, kur laikomas įvedimo rezultatas.

* PDxy - išsiuncia išvedimui 10 žodžių srautą iš atminties lastelių pradedant ląstele [x \*10 + y]. Formalus užrašymas:

For i:=0 to 9 do

Buffer[i]:= [x \*10 + y]; y:=y+1;

Next i

Čia Buffer – 10 žodžių masyvas, kur laikomas išvedimo rezultatas.

* JPxy - besąlyginis komandos perdavimas adresu [x \*10 + y];
* CAxy – komanda call, isaugoma procesoriaus būsena ir IC priskiriama xy. IC = xy
* RETR – gryztama į buvusia buseną atstatant buvusius registrus, atstačius IC = IC + 1
* HALT - vartotojo programos vykdymo pabaiga.

**Virtualios mašinos atmintis**

Kiekvienai virtualiai mašinai yra skiriama 10 vartotojo atminties blokų. Tuose dešimtyje blokų (100 žodžių) turi tilpti užduoties programa. Kiekvienas virtualios atminties blokas turi virtualų ir realų adresą.

**Puslapiavimo mechanizmas**

Realios mašinos vartotojo atmintis siekia 100 takelių (arba blokų). Kiekvienai naujai sukurtai virtualiai mašinai reikia skirti 10 takelių iš tų 100. Jie gali būti parinkti bet kokiu būdu. Klausimas: kaip virtuali mašina gali sužinoti kokio nors jai priklausančio takelio realų adresą? Tam naudosime puslapiavimo mechanizmą.

Puslapiavimo mechanizmo esmė: sakykime, kuriama nauja virtuali mašina. Jai reikia dešimt takelių atminties. Mes parinkome takelius su numeriais: 1, 3, 5, 7, 9,11, 13, 15, 17, 19. Šiais takeliais naudosis virtuali mašina. Pati virtuali mašina šiuos takelius mato sunumeruotus nuo 0 iki 9. t.y. 1 takelis jai yra nulinis, 5 takelis jai yra antras, o 19 takelis – devintas. Kaip išlaikyti sąryšius tarp realių ir virtualių takelių adresų? Naudosime puslapių lentelę. Puslapių lentelė – tai vienas takelis (t.y. 10 žodžių). Kiekvieno žodžio eilės numeris atitiks virtualios mašinos takelio numerį, ir jame (žodyje) bus laikomas realus to takelio numeris. Pavyzdžiui, prieš tai pateikto pavyzdžio puslapių lentelė bus tokia:

Taigi, dabar virtuali mašina, norėdama sužinoti realų takelio adresą, kreipiasi į savo puslapių lentelę ir nuskaito reikšmę, esančią žodyje su takelio numeriu. Pvz. penkto virtualios mašinos takelio realus numeris yra 11. Dabar turi išaiškėti procesoriaus registro PTR prasmė. Juk puslapių lentelė taip pat yra atminties takelis. Ir šis takelis, be abejo, taip pat turi savo realų numerį. Todėl registro PTR reikšmė – puslapių lentelės adresas. Kiekviena virtuali mašina, prieš pradėdama darbą, nustatys šį registrą jai reikalinga reikšme.

Dabar galime apžvelgti patį registrą PTR. Tai 4 baitų registras. Simboliškai pažymėsime PTR reikšmę a0a1a2a3, kur ai yra baitai. Baitas a0 yra nenaudojamas, a1 puslapių lentelės ilgis atėmus 1 (arba nenaudojamas. Tai priklauso nuo puslapiavimo mechanizmo realizacijos), o štai a2 \* 10 + a3 žymi puslapių lentelės adresą. Dabar galime pateikti formulę, kuri virtualiam adresui x1x2 gražina realų adresą:

**Realus adresas = 10\*[10\*(10 \* a2 + a3) + x1] + x2**

**Realios mašinos modelis**



Išorinė atmintis

# Realios mašinos registrai

* MOD :- 1 baito procesoriaus buseną nusakantis registras
* R :- 4 baitų bendro naudojimo registras
* PTR :- 4 baitų puslapių lentelės registras
* IC :- 2 baitų komandų skaitliuko registras
* SP :- 3 baitų steko viršūnes registras
* C – požymio registras
* INT :- 3 baitų pertraukimo proceduros registras
* SI :- 1 baito supervisorinių pertraukimų registras
* PI :- 1 baito programinių pertraukimu registras
* TI :- 1 baito taimerio registras

# Realios mašinos procesoriaus komandos

Procesoriaus paskirtis - skaityti komandą iš atminties ir ją vykdyti (interpretuoti). Procesorius gali dirbti dviem režimais – supervizoriaus arba vartotojo.

Komandos:

* CHNG – pakeičiama MOD registro reikšmė i priešingą. If MOD = 1 then MOD = 0, else MOD = 1
* Lreg – Į resgistrą R įrašoma nurodyto registro reiksmė. R = [reg]
* SESx – nustatoma registro SI reikmė nurodytu skaičiumi. SI = x
* SEPx – nustatoma registro PI reikšmė nurodytu skaičiumi. PI = x
* SETx – nustatoma registro TI reikšmė nurodytu skaičiumi. TI = x
* SPxy – nustatoma registro PTR reikšme, ji rodys puslapiu lenteles bloko numerį. PTR = x \* 10 + y
* Sxyz – nustatomas steko viršūnės adresas. SP = x \* 100 + y \* 10 + z
* Ixyz – nustatomas pertraukimų registro adresas. INT = x \* 100 + y \* 10 + z
* Ureg – Į [SP] įšaugomas nurodytas registras, steko viršūnė sumažinama vienetu.
* Oreg – Iš [SP] reikšmė irašoma į nurodytą registrą, steko viršūnė padidinama vienetu
* XCHG - vykdomas apsikeitimas duomenimis pagal nustatytus kanalų įrenginio registrus.

# Procesoriaus darbo algoritmas

1. Dirbama virtualiojoje mašinoje, vartotojo rėžimu, kai MODE = 1. O supervizoriniame, kai MODE = 0
2. Vykdoma instrukcija IC registre esančiu adresu. Vykdant instrukciją gali būti kelios situacijos:
   * Komanda atpažįstama ir įvykdoma
   * Komanda neatpažįstama, įvyksta pertraukimas, identifikuojantis neteisingą komandą (supervizoriaus rėžime, tokios situacijos negali pasitaikyti)
   * TI registrui sumažėjus iki 0, įvyksta taimerio pertraukimas
3. IC registro reikšmė padidėja vienetu.
4. Vartotojo rėžime su funkcija test() tikrinama ar buvo užfiksuotas pertraukimas. Jei taip, tai išsaugojama virtualios mašinos būsena, bus dirbama su realios mašinos atmintimi, vykdomas pertraukimų apdorotojas, kurio adresas nurodytas INT registre, perjungiama į supervizoriaus rėžimą MODE := 0
5. Jei buvo apdorojamas pertraukimas, perjungtas rėžimas MODE = 0, sugrąžinamos virtualios mašinos registrų reikšmės, perjungiamas rėžimas MODE := 1;

**Realios mašinos atmintys**

Atmintis – įrenginys informacijai saugoti. Mūsų reali mašina turi dviejų rūšių atmintis: vartotojo ir išorinę.

Vartotojo atmintis skirta virtualių mašinų atmintims bei puslapių lentelėms laikyti. Mes apibrėšime vartotojo atmintį taip: lentelės dydis – 1000 žodžių po 4 baitus (1 baitas – 8 bitai). 10 žodžių laikysime bloku (takeliu). Taigi vartotojo atmintis lygi 100 blokų, sunumeruotų nuo 0 iki 99, arba 1000 žodžių, sunumeruotų nuo 0 iki 999.

Išorinė atmintis:

Šioje atmintyje saugomi programų tekstai.

Procesorius darbą su atmintimis valdo naudodamas kanalų įrenginį.   
Šios atminties dydis yra pastovus – 400 blokų arba 4 000 žodžių (sunumeruotų nuo 0 - 3999) po 4 baitus (1 baitas – 8 bitai)

**Pertraukimai**

Programinių pertraukimų registras yra PI, supervizorinių pertraukimų registras – SI, įvedimo/išvedimo - IOI, taimerio - TI.

Pertraukimai kils šiais būdais:

* Operacijos GD, PD ir HALT iššauks supervizorinius pertraukimus.
  + SI = 1 - komanda GD,
  + SI = 2 - komanda PD,
  + SI = 3 – komanda HALT.
* Programiniai pertraukimai:
  + PI = 1 – atminties apsaugos pažeidimas,
  + PI = 2 – neleistinas operacijos kodas,
* Esant TI = 0 bus fiksuojamas taimerio pertraukimas.
* Esant situacijai SI = 0 ir PI = 0 ir IOI = 0 ir TI <> 0, pertraukimų sistema neaptiks.

Pertraukimai nustatomi paprasčiausiai virtualaus procesoriaus registrams priskiriant atitinkamas reikšmes (pavyzdžiui, komandų interpretatoriui vykdant komandą GD, jis priskiria SI:= 1) Kiekvieną kartą komandų interpretatoriui įvykdžius programą, kviečiama komanda test(), kuri apklausia registrus, ir, jei kilo pertraukimas, gražina informaciją apie tai.

Komanda test():

If ( PI + SI + TI) > 0 then Test := 1 else Test := 0;

Nustačius įvykusį pertraukimą (t.y. test() <> 0) virtualios mašinos procesoriaus registrų reikšmės yra išsaugomos, procesorius perjungiamas į supervizoriaus režimą, nustatomas pertraukimo pobūdis (pavyzdžiui tiesiogiai apklausiant registrus) ir kviečiama pertraukimą apdorosianti paprogramė. Vėliau, kaip buvo minėta, valdymas grįžta į virtualią mašiną, registrai atstatomi, procesorius perjungiamas į vartotojo režimą, ir operacinė sistema sprendžia, ką daryti toliau.

# Kanalų įrenginys

Kanalų įrenginys leidžia dirbti su atmintimis. Priklausomai nuo nustatytų registrų kanalų

įrenginys gali vykdyti apsikeitimą duomenimis visomis galimomis kryptimis.Veiksmai su

kanalų įrenginiu atliekami tik supervizoriaus režime. Dabar bus pateikta kanalų įrenginio

vartotojo sąsaja:

Kanalų įrenginio registrai:

SB: Takelio, iš kurio kopijuosime numeris.

DB: Takelio, į kurį kopijuosime numeris

ST: Objekto, iš kurio kopijuosime, numeris

1. Vartotojo atmintis;

2. Išorinė atmintis;

3. Įvedimo srautas;

DT: Objekto, į kurį kopijuosime, numeris

1. Vartotojo atmintis;

2. Išorinė atmintis;

3. Išvedimo srautas;

Kartu kanalų įrenginys turi komandą XCHG, tačiau neturi procesoriaus, kuris galėtų ją

įvykdyti. Šią komandą vykdo centrinis procesorius, taigi, šis kanalų įrenginys nėra

lygiagrečiai su centriniu procesoriumi veikianti aparatūra.

Procesas, norėdamas pasinaudoti kanalų įrenginiu, turi nustatyti kanalų įrenginio

registrus ir tada įvykdyti komandą XCHG.

**Taimerio mechanizmas**

Laikysim kad įvedimo/išvedimo instrukcijos atliekamos per 3 laiko momentus, visos kitos per 1 laiko momentą. Dabar apie veikimo principą.

Pradedant virtualios mašinos užduoties vykdymą TI registro reikšmė nustatoma tam tikrai reikšmei. Tarkim TI = 10. Įvykdžius eilinę instrukciją TI reikšmė mažinama priklausomai nuo to per kiek laiko momentų ši instrukcija yra atliekama. Kuomet TI reikšmė yra lygi nuliui, mikrokomanda Test () aptinka taimerio pertraukimą.