Kurso „Operacinės sistemos” I-asis darbas

Realios ir virtualios mašinų projektai

Darbą parengė:
Algis Povilauskas, Informatika 3k. 2gr.
Deividas Zaleskis, Informatika 3k. 2gr.
Deividas Žemeckas, Informatika 3k. 2gr.

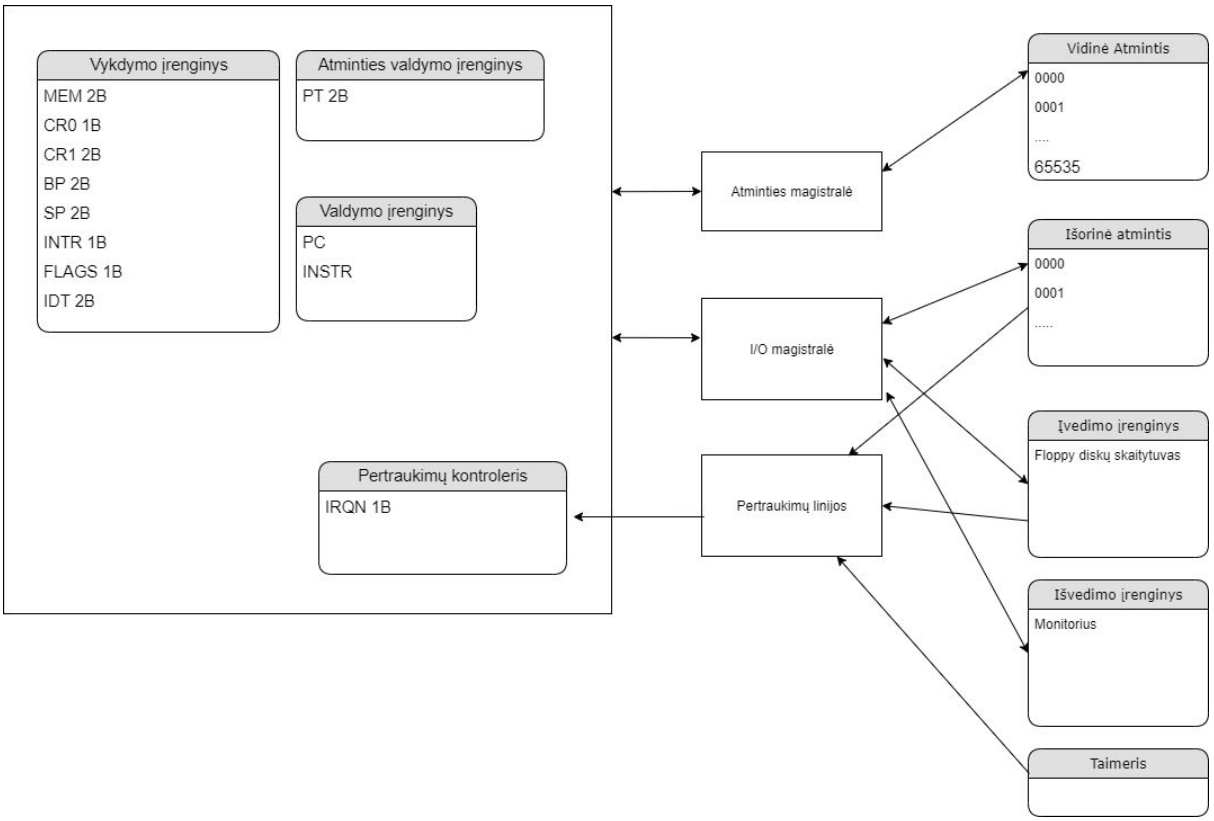
Pratybų dėstytojas:
Rokas Masiulis

Turinys

1. Reali mašina

Mašiną sudarys šios komponentės:

1. Centrinis procesorius (CPU)
2. Atmintis
    a. Vidinė (RAM)
    b. Išorinė (kietasis diskas)
3. Įvedimo įrenginys (floppy diskų nuskaitymo įrenginys)
4. Išvedimo įrenginys (monitorius)
5. Taimeris



Realios mašinos schema

Realios mašinos centrinis procesorius

Centrinis procesorius skaito instrukcijas iš atminties ir jas vykdo. Procesorius operuoja 16 bitų žodžiais. Taip pat yra vidinis laikrodis, kuris valdo ir sinchronizuoja procesoriaus ciklus (1 laikrodžio taktas - 1 procesoriaus ciklas).

Procesoriaus komponentai:

- Vykdymo įrenginys - jame yra aritmetinis loginis įrenginys bei procesoriaus registrai.
- Valdymo įrenginys - jame yra registras, saugantis esamą instrukciją, bei komandų skaitiklis.
- Atminties valdymo įrenginys - įgyvendina virtualių adresų transliaciją į realius. Registre saugo esamą puslapių lentelę.
- Pertraukimų kontroleris - leidžia išoriniams įrenginiams pateikti informaciją procesoriui naudojant pertraukimus. Turi registrą, kuriame saugomas pertraukimą sukėlusio įrenginio linijos numeris.

Procesoriaus registrai:

- CR0 - 1 baito registras, nusakantis procesoriaus veikimą.
- CR1 - 2 baitų registras, naudojamas kai įjungtas puslapiavimas, saugo bandytą pasiekti virtualų adresą, kuris sukėlė puslapio klaidą.
- PC - 2 baitų komandų skaitiklis.
- BP - 2 baitų base pointer.
- SP - 2 baitų steko viršūnės registras.
- PT - 2 baitų puslapių lentelės rodyklė.
- INTR - 1 baito pertraukimų registras.
- IRQN - 1 baito pertraukimų kontrolerio linijos numerio registras.
- FLAGS - 1 baito registras, saugo procesoriaus būseną.
- IDT - 2 baitų registras, saugo adresą į pertraukimų vektorių lentelę.

Apriboti registrai (nėra prieinami instrukcijomis):

- INSTR - 4 baitų registras, saugo nuskaitytą instrukciją.
- MEM - 2 baitų registras, saugo nuskaitytą iš atminties reikšmę.

Realios mašinos procesorius gali dirbti dviem rėžimais, priklausomai nuo registro CR0 reikšmės:

- Apsaugotame režime procesorius imituoja virtualios mašinos procesorių ir naudoja puslapiavimą bei virtualią atmintį.
- Realiame rėžime instrukcijos iš realios atminties yra betarpiškai apdorojamos.

Vieno realaus procesoriaus ciklo aprašymas:

- Procesorius nuskaito sekančią instrukciją iš atminties pagal PC registro reikšmę ir patalpina ją į registrą INSTR.
- PC registras yra padidinamas taip, kad rodytų į sekančią instrukciją.
- Instrukcija dekoduojama.
- Jei dekodavus aptinkama, kad instrukcijai reikia reikšmės iš atminties, ji yra nuskaitoma į registrą MEM.
- Procesorius įvykdo instrukciją. Operacijai naudojant ALU, nustatomas registras FLAGS. Įvykus klaidai, atitinkamai nustatomas registras INTR.

- Jei registro INTR reikšmė nelygi nuliui, iššaukiamas programinio pertraukimo apdorojimas, remiantis pertraukimų vektorių lentele. INTR nustatomas atgal į 0.
- Jei pertraukimų kontrolerio registras IRQN nelygus 0, iššaukiamas aparatinio pertraukimo apdorojimas, remiantis pertraukimų vektorių lentele. IRQN nustatomas atgal į 0.

Realios mašinos atmintis

Procesorius turės tiesioginį priėjimą prie atminties per sistemos magistralę. Magistralė yra 16 bitų ilgio, taigi adresų erdvė bus 65536 baitų (64KB).

Pertraukimai

Tai tam tikri signalai apie specialius įvykius.
Pertraukimai gali būti programiniai arba aparatiniai. Programinius pertraukimus sukelia komanda INT, aparatinius - išoriniai įrenginiai arba pats procesorius.

Procesorius pertraukimus aptinka ir apdoroja vykdymo ciklo pabaigoje. Apdorojimui naudojama pertraukimų vektorių lentelė, kuri saugoma IDT registre. Aptikus pertraukimą, procesoriaus būsena yra išsaugoma steke, pagal pertraukimo kodą iššaukiamas atitinkamas apdorojimo vektorius remiantis vektorių lentele, pvz pertraukimas 05h bus apdorojamas šokant į IDT[5]. Apdorojus pertraukimą, procesoriaus būsena yra atstatoma.

Išorinių įrenginių sukeltiems pertraukimams aptikti reali mašina turi pertraukimų kontrolerį. Kontroleris turi 8 IRQ linijas, į kurias siunčiant signalus atitinkamai į registrą IRQN įrašomos reikšmės nuo 1 iki 8. Per vieną procesoriaus ciklą įvykus keletai signalų, bus apdorojamas tik tas, kurį procesorius nuskaitys aptikdamas pertraukimus.

Įvedimas ir išvedimas

Įvedimui ir išvedimui naudojamos komandos IN ir OUT. Jos į atitinkamą prievadą išsiunčia/nuskaito duomenis. Išorinių įrenginių prievadai egzistuoja atskiroje atminties erdvėje, kuri ir adresuojama šiomis instrukcijomis.

Išoriniai įrenginiai, norėdami signalizuoti, jog atsitiko kažkoks įvykis, pvz. Vartotojas paspaudė klaviatūros klavišą, iššaukia aparatinį pertraukimą per atitinkamą IRQ liniją. Tokiu būdu procesoriui pranešama, jog tam tikras įrenginys turi būti aptarnautas. Laikome, jog IRQ 1 bus prijungtas taimeris. Į kokias IRQ linijas sujungti kitus įrenginius ir kaip apdoroti jų pertraukimus sprendžia atitinkamai vartotojas ir OS.

Būsena

Registro FLAGS sandara:

ZF 0x01	SF 0x02	OF 0x04	CF 0x08	R 0x10	R 0x20	R 0x40	R 0x80

ZF - Zero Flag.
SF - Sign Flag.
OF - Overflow Flag.
CF - Carry Flag.
R - Reserved

Realios mašinos procesoriaus instrukcijos

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 – (be ženklo) sudaugina du viršutinius steko elementus. Rezultatą padeda į steko viršūnę ir steko rodyklę sumažina vienetu.
[SP – 1] = [SP – 1] * [SP]; SP--;

IMUL – (su ženklu) sudaugina du viršutinius steko elementus. Rezultatą padeda į steko viršūnę ir steko rodyklę sumažina vienetu.
[SP – 1] = [SP – 1] * [SP]; SP--;

DIV – (be ženklo) 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--;

IDIV – (su ženklu) 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--;

Loginės

AND – loginė konjunkcija pagal kiekvieną bitą.
[SP – 1] = [SP – 1] & [SP]; SP--;

OR – loginė disjunkcija pagal kiekvieną bitą.
[SP – 1] = [SP – 1] | [SP]; SP--;

XOR - loginis XOR pagal kiekvieną bitą
[SP – 1] = [SP – 1] ^ [SP]; SP--;

NOT - loginis neigimas pagal kiekvieną bitą
[SP] = ![SP];

Palyginimo

CMP – apskaičiuojamas pirmų dviejų nuo viršaus steko elementų skirtumas. Rezultatas niekur neįrašomas, nustatomas registras FLAGS.

Darbo su duomenimis / steko

LOD *address* – į steko viršūnę užkrauna žodį iš atminties nurodytu adresu
[SP+1] = [address]; SP++;

LODE – į steko viršūnę užkrauna žodį iš atminties steko viršūnėje esančiu adresu
[SP+1] = [[SP]]; SP++;

STO *address* – steko viršūnėje esantį žodį įrašo į atmintį nurodytu adresu
[address] = [SP];

STOE – steko viršūnėje esantį žodį įrašo į atmintį adresu, esančiu antrame nuo viršaus steko elemente
[[SP-1]] = [SP];

PUSH *immediate* - į steko viršūnę padeda nurodytą žodį.
[SP+1] = immediate; SP++;

POP - pastumia steko viršūnę atgal vienu žodžiu.
SP--;

Valdymo

JMP *address* – valdymas besąlygiškai perduodamas nurodytu adresu.
PC = address;

JE *address* – jei palyginimo operandai lygūs, valdymas perduodamas nurodytu adresu.
IF(ZF == 1) PC = address;

JL *address* – jei pirmas palyginimo operandas mažesnis už antrą, valdymas perduodamas nurodytu adresu.

IF(SF != OF) PC = address;

JLE *address* – jei pirmas palyginimo operandas mažesnis arba lygus antram, valdymas perduodamas nurodytu adresu.
IF(SF != OF OR ZF == 1) PC = address;

JG *address* – jei pirmas palyginimo operandas didesnis už antrą, valdymas perduodamas nurodytu adresu.
IF(ZF == 0 AND SF == OF) PC = address;

JGE *address* – jei pirmas palyginimo operandas didesnis ar lygus antram, valdymas perduodamas nurodytu adresu.
IF(SF == OF) PC = address;

LOOP address - sumažina steko viršūnės reikšmę, jei viršūnėje ne 0, šoka nurodytu adresu.
[SP] -= 1; IF([SP] != 0) PC = address;

CALL address - valdymas perduodamas procedūrai nurodytu adresu.
[SP+1] = PC; SP++;

RET - grįžtama iš procedūros atgal į kvietėją.
PC = [SP]; SP--;

INT immediate - iškviečiamas programinis pertraukimas.
INTR = immediate.

Būsenos

POPR register - nustato registrą į reikšmę iš steko viršūnės.
register = [SP]; SP--;

PUSHR register - registro reikšmę patalpina į steko viršūnę.
[SP+1] = register; SP++;

Įvedimo ir išvedimo

OUTB port - į atitinkamą prievadą išsiunčia žemesnį baitą iš steko viršūnės.
[SP] => port;

OUTW port - į atitinkamą prievadą išsiunčia žodį iš steko viršūnės.
[SP] => port;

INB port - iš atitinkamo prievado nuskaito baitą į steko viršūnę kaip žodį.
port => [SP+1]; SP++;

INW port - iš atitinkamo prievado nuskaito reikšmę į steko viršūnę.
port => [SP+1]; SP++;

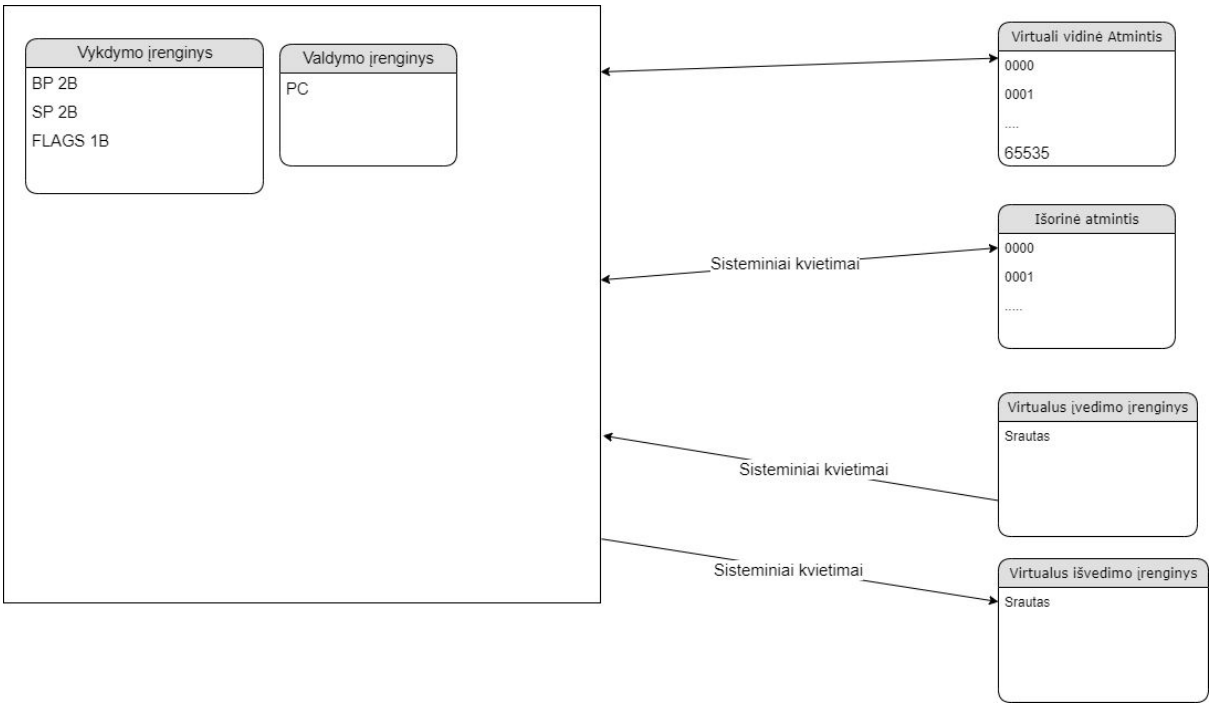
Laikysime, jog kiekviena operacija yra 4 baitų ilgio. Operacijos kodui bus skiriami 2 baitai, operandui 2 baitai. Kai kurios instrukcijos nenaudoja operandų, tad jų vieta turės būti užpildyta nuliais.

Perėjimas į apsaugotą rėžimą

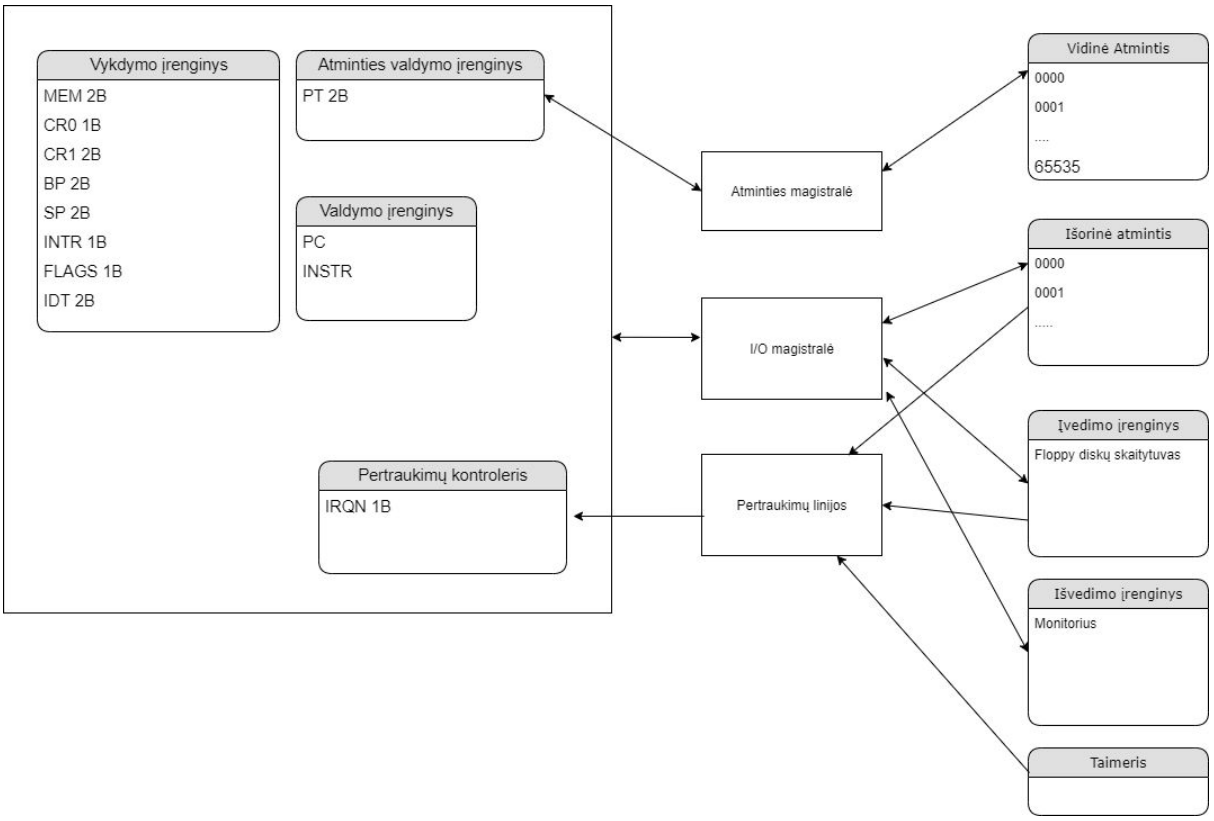
Norint pereiti į apsaugotą rėžimą, reikia sukurti puslapių lentelę, įrašyti jos adresą į registrą PT ir nustatyti CR0 registrą į 1.

2. Virtuali mašina

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, sukuriant iliuziją, jog programa yra vienintelė veikianti tame kompiuteryje ir naudojanti jo resursus.



Virtualios mašinos schema (tokia, kokią mato vartotojas)



Virtualios mašinos schema (“po kapotu”)

Virtualios mašinos centrinis procesorius

Virtualus procesorius skirtas vykdyti vartotojo programas, todėl jo veikimas vietomis apribotas ar supaprastintas.

Virtualus procesorius turi tokius registrus:

- PC - 2 baitų komandų skaitiklis.
- BP - 2 baitų base pointer.
- SP - 2 baitų steko viršūnės registras.
- FLAGS - 1 baito registras, saugo procesoriaus būseną.

Apriboti registrai (nėra prieinami instrukcijomis):

- CR0 - 1 baito registras, nusakantis procesoriaus veikimą. Apribotas
- CR1 - 2 baitų registras, naudojamas kai įjungtas puslapiavimas, saugo bandytą pasiekti virtualų adresą, kuris sukėlė puslapio klaidą. Apribotas
- PT - 2 baitų puslapių lentelės rodyklė. Apribotas
- INTR - 1 baito pertraukimų registras. Apribotas.
- IRQN - 1 baito pertraukimų kontrolerio linijos numerio registras. Apribotas.
- IRQN - 1 baito pertraukimų kontrolerio linijos numerio registras. Apribotas.

- IDT - 2 baitų registras, saugo adresą į pertraukimų vektorių lentelę. Apribotas

Vieno virtualaus procesoriaus ciklo aprašymas:

- Procesorius nuskaito sekančią instrukciją iš virtualios atminties pagal PC registro reikšmę ir patalpina ją į registrą INSTR.
- PC registras yra padidinamas taip, kad rodytų į sekančią instrukciją.
- Instrukcija dekoduojama.
- Jei dekodavus aptinkama, kad instrukcijai reikia reikšmės iš virtualios atminties, ji yra nuskaitoma į registrą MEM naudojant atminties valdymo įrenginį.
- Jei nėra puslapio klaidos (INTR == 0), procesorius įvykdo instrukciją. Operacijai naudojant ALU, nustatomas registras FLAGS. Vykdant aptikus klaidą, atitinkamai nustatomas registras INTR.
- Jei registro INTR reikšmė nelygi nuliui, iššaukiamas programinio pertraukimo apdorojimas, remiantis pertraukimų vektorių lentele. INTR nustatomas atgal į 0.
- Jei pertraukimų kontrolerio registras IRQN nelygus 0, iššaukiamas aparatinio pertraukimo apdorojimas, remiantis pertraukimų vektorių lentele. IRQN nustatomas atgal į 0.

Virtualios mašinos instrukcijos

Instrukcijos tokios pat, kaip ir realios mašinos, tik apribotas priėjimas prie kai kurių registrų. Laikome, jog visi adresai dirba su virtualia atmintimi. Įvedimo ir išvedimo instrukcijos nėra prieinamos, tam naudojami pertraukimai.

Virtualios mašinos atmintis

Virtuali mašina su atmintimi dirba kitaip nei reali. Kiekviena virtuali mašina turi savo adresų erdvę, kuri įgyvendinama naudojant virtualių adresų transliaciją į realius per atminties valdymo įrenginį. Tai leidžia VM procesams dirbti laikant, jog yra prieinama visa 64 KB dydžio mašinos atmintis. Iš tikrųjų, atmintis yra išskiriama puslapiais po 512 baitų, o jų atitikmenys saugomi puslapių lentelėje, kurios adresas yra registre PT.

Puslapių lentelė yra pusės puslapio dydžio struktūra, kurioje saugomi 128 realių puslapių adresai (128 žodžiai, 256 baitai).

Matome, jog 128 puslapiai po 512 baitų leidžia mums adresuoti visą mašinos atmintį (64 KB)

Pateikiame puslapių lentelės segmentą kaip pavyzdį:

i = 0	i = 1	i = 2	i = 3	i = 4	i = 5
48568	27684	1024	10236	24512	5188

Norint rasti virtualų adresą atitinkantį realų adresą, pirmiausia turime rasti virtualų puslapį atitinkantį realaus puslapio adresą.

Rasime realaus puslapio indeksą puslapių lentelėje:
realaus_puslapio_indeksas = virtualus_adresas / 512.

Tada realaus_puslapio_adresas = PT[realaus_puslapio_indeksas].

Turint realaus puslapio adresą, belieka prie jo pridėti virtualaus adreso poslinkį nuo virtualaus puslapio pradžios, t.y.
realus_adresas = realaus_puslapio_adresas + virtualus_adresas % 512.

Jei ieškant virtualaus adreso atitikmens, puslapių lentelėje nerandamas tinkamas įrašas, į registrą INTR įrašomas 10, o į CR1 registrą įrašomas bandytas pasiekti virtualus adresas.

Pertraukimai

Apsaugotame rėžime aptikus pertraukimą, CR0 atstatomas į 0, t.y. grįžtama į realų rėžimą. Tada apdorojimas vyksta taip pat, kaip aprašyta prie realios mašinos. Po apdorojimo, OS nusprendžia ką daryti toliau.

Virtualūs įvedimo/išvedimo įrenginiai

Virtualus išvedimo įrenginys - srautas. Į jį galima nuosekliai rašyti baitus, nesirūpinant kiek jų ten yra, ar jie tilps ir kokiu formatu juos išvesti.

Virtualus įvedimo įrenginys - taip pat srautas. Iš jo galima nuosekliai skaityti baitus kol bus pasiekta pabaiga.