Vilniaus Universitetas
Matematikos ir Informatikos fakultetas
Informatikos katedra

Realios ir virtualios mašinų projektas

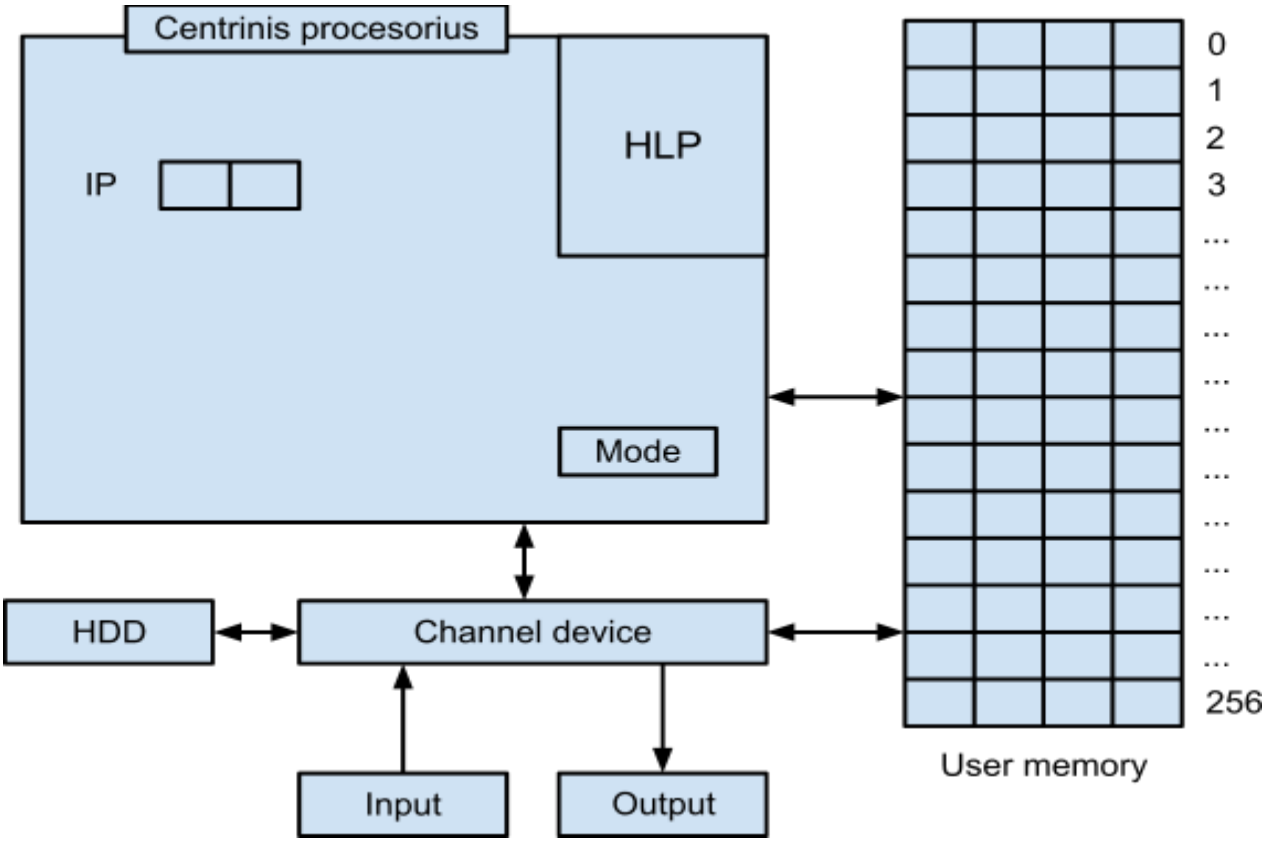
Darbą atliko:
Valerij Bielskij
Tadeuš Varnas

Vilnius
2012

Reali mašina - tai kompiuteris, kuris mūsų atvejų bus sudarytas iš šių komponentų:

- Centrinis procesorius - procesoriaus paskirtis skaityti komanda iš atminties ir ją vykdyti. Procesorius dirbs dviem režimais.

    - Supervizoriaus režimas - komandos bus betarpiškai apdorojamos aukšto lygio kalbos procesoriaus HLP.
    - Vartotojo režimas - HLP imituos virtualios mašinos procesorių kuris vykdys užduotį.
      HLP – bet kuris aukšto lygio kalbos procesorius (programavimo kalba).
- Atmintis - realios mašinos atmintis susidės iš 256 žodžių, vienas žodis 4B. Visa atmintis bus padalyta i blokus, 1 blokas = 16 žodžių, iš viso 16 blokai po 16 žodžių.
    - Vartotojo - skirta laikyti virtualių mašinų atmintims.
    - Išorinė - mūsų atveju kietasis diskas(HDD).
    - Supervizoriaus - kadangi sisteminius procesus, kintamuosius ir resursus valdys aukšto lygio kalbos procesorius HLP, supervizoriaus atmintys tampa tik savoką ir realiai realizuota nebus.
    - Bendra – kiekvienai virtualiai mašinai yra išskirta realios mašinos atmintis prieinama tik konkrečiai virtualiai mašinai.
- Įvedimo įrenginys - klaviatūra.
- Išvedimo įrenginys - ekranas.

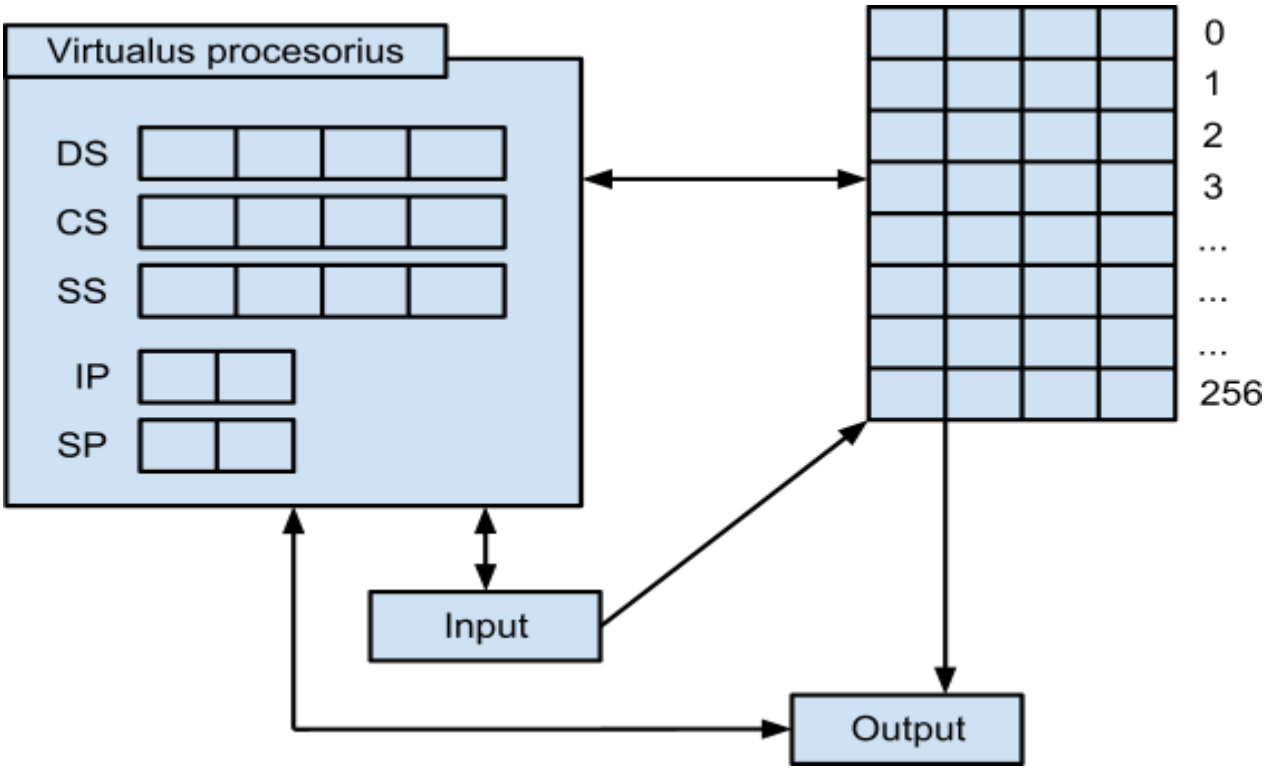


Procesoriaus registrai:

- IP - Instruction Pointer - 2B registras skirtas saugoti vykdomos komandos adresą..

- Mode - registras nusakantis darbo režimą.

Virtuali mašina - tai supaprastintas realios mašinos modelis. Virtualios mašinos paskirtis vykdyti programą kurį yra virtualioje atmintyje.



Virtualios mašinos procesoriaus registrai:

- DS - Data Segment - 4B registras skirtas saugoti data segmento adresą.
- CS - Code Segment - 4B registras skirtas saugoti kodo segmento adresą.
- SS - Stack Segment - 4B registras skirtas saugoti steko segmento adresą.
- IP - Instruction Pointer - 2B registras skirtas saugoti vykdomos komandos adresą.
- SP - Stack Pointer. - 2B registras skirtas saugoti steko viršūnės indekso adresą.

Virtualios mašinos atmintis - atmintis susidės iš 256 žodžių, kiekvienas žodis po 4B. Visa atmintys bus suskirstyta į blokus, 1 blokas = 16 žodžių, iš viso 16 blokų po 16 žodžių. Atmintis bus padalinta į trys dalys, į kurias atitinkamai bus rašomas programos kodas.

- DS - Data Segment turės 4 blokus.
- CS - Code Segment turės 6 blokus.
- SS - Stack Segment turės 6 blokus.

Virtualios mašinos komandos:

- Aritmetinės

    - ADD - sudeda du steko viršūnes elementus, sumažina SP vienetu ir padeda rezultatą į steko viršūnę. SS[SP-1] = SS[SP-1] + SS[SP]; SP -= 1;
    - SUB - atima du steko viršūnes elementus, sumažina SP vienetu ir padeda rezultatą į steko viršūnę. SS[SP-1] = SS[SP-1] - SS[SP]; SP -= 1;
    - MUL - sudaugina du steko viršūnes elementus, sumažina SP vienetu ir padeda rezultatą į steko viršūnę. SS[SP-1] = SS[SP-1] * SS[SP]; SP -= 1;
    - DIV - padalina du steko viršūnes elementus, sumažina SP vienetu ir padeda rezultatą į steko viršūnę. SS[SP-1] = SS[SP-1] / SS[SP]; SP -= 1;
    - NEG - steko viršūnėje esanti elementą pakeičia priešingu. SS[SP] = 0 - SS[SP];
- Loginės
    - AND - atlieka dviejų viršutinių steko elementų konjunkciją, SP sumažina vienetu ir rezultatą priskiria steko viršūnei.
      SS[SP-1] = SS[SP-1] & SS [SP]; SP -= 1;
    - OR – atlieka dviejų viršutinių steko elementų disjunkciją, SP sumažina vienetu ir rezultatą priskiria steko viršūnei.
      SS[SP-1] = SS [SP-1] | SS [SP]; SP -= 1;
    - NOT – atlieka steko viršunėje esančio žodžio loginį neigimą.
      SS[SP] = !SS[SP];
- 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 -= 1
- Duomenų/Steko
    - DSxy - į steko viršūnę SP +=1 užkraunama reikšmė iš data segmento adresu 16 * x + y.
      SP +=1; SS[SP] = 16 * x + y
    - SDxy - steko viršūnėje esančia reikšme užkrauname į data segmentą adresu 16 * x + y, SP -=1.
- Valdymo perdavimo
    - JPxy - nesąlyginio valdymo perdavimo komanda. Valdymas perduodamas kodo sričiai nurodytam adresui.
      IP = 16 * x + y
    - JExy - jei steko viršūnėje yra 1 valdymas perduodamas adresu 16 * x + y.
      IF(SS[SP] == 1) IP = 16 * x + y; SP -= 1;
    - JLxy - jei steko viršūnėje yra 0 valdymas perduodamas adresu 16 * x + y.
      IF(SS[SP] == 0) IP = 16 * x + y; SP -= 1;
    - JGxy – jei steko viršūnėje yra 2 valdymas perduodamas adresu 16 * x + y.
      IF(SS [SP] == 2) IP = 16 * x + y; SP -= 1;
    - HALT - programos užbaigimo komanda
- Įvedimo/Išvedimo
    - PUTS – steko viršūnėje esantį žodį traktuoja kaip simbolius ir nusiunčia į išvedimo įrenginį. SP -= 1;

    - PUTI – steko viršūnėje esantį žodį traktuoja kaip skaitinę reikšmę ir nusiunčia į išvedimo įrenginį. SP -= -1;
    - READ - nuskaito vartotojo įvedimą kaip simbolius ir įrašo į steko viršūnę.

Programos formatas:
Programos bus laikomos failuose. Vienoje eilutėje viena komanda. Kokia seka duomenys pateikiami duomenų segmento apraše, tokia tvarka jie talpinami ir į atmintį, skirtą duomenų segmentui. Duomenų segmente:

- DW x - išskiriamas vienas žodis su nurodyta skaitine reikšme.
- DB xxxx - išskiriamas vienas žodis su nurodytais keturiais simboliais.

#DATA
….
#CODE
….
HALT

Programos kodas:
išraiška 50 + 100 - 24 * 5

DATA
DW 50
DW 100

```
DW 24
DW 5
DB rezu
DB ltat
DB as:
CODE
DS00
DS01
ADD
DS02
DS03
MUL
SUB
DS04
PUTS
DS05
PUTS
DS06
PUTS
PUTI
HALT
```