|  |
| --- |
|  |
| **Operacinės sistemos**  Pirmasis praktinis darbas  „Virtualios mašinos bei realios mašinos apibrėžimas ir realizacija“ |
| Darbą atliko:  Inf 4 kurso studentas Lukas Chodosevičius |

Turinys

[I. Realios mašinos aprašas 3](#_Toc475271610)

[ Realios mašinos komponentu schema ir jų tarpusavio sąryšis: 3](#_Toc475271611)

[ Realią mašiną sudaro: 3](#_Toc475271612)

[ Detalesnė kievkieno komponento aprašas: 3](#_Toc475271613)

[II. Virtualios mašinos aprašas 4](#_Toc475271614)

[ Procesorius 4](#_Toc475271615)

[ Atmintis 4](#_Toc475271616)

[ Virtaulios mašinos instrukcijos 4](#_Toc475271617)

[ Virtualios mašinos interpretuojamo failo formatas 5](#_Toc475271618)

[ Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais aprašymas 5](#_Toc475271619)

[III. Virtuali mašina visos operacinės sistemos kontekste 5](#_Toc475271620)

# Realios mašinos aprašas

## Realios mašinos komponentu schema ir jų tarpusavio sąryšis:

**RAM**

**Kernel memory**

**User memory**

**Page Table**

C

**CPU**

**MMU**

MODE

IC

SC

PR

**Keyboard**

**Monitor**

**External memory**

**Timer**

**Channel 3**

CHAN

**Channel 4**

**Channel 1**

**Channel 2**

## Realią mašiną sudaro:

CPU (centrinis procesorius), RAM („Kernel“ ir „User“ atmintis), channels (kanalai), keyboard (įvesties įrenginys), monitor (išvesties įrenginys) ir external memory (išorinė atmintis).

## Detalesnė kiekvieno komponento aprašas:

**CPU** (Centrinis procesorius) – pagrindinis įrenginys leidžiantis interpretuoti paduotas instrukcijas.

Turi du veikimo režimus, kuris nurodomas MODE registre:

1. „Kernel“ – dar kaip žinomas supervizorinis, t.y. pilnas CPU režimas, kuriam nėra taikomi apribojimai. Ši režimas prieinamas tik OS programai.
2. „User – apribotas režimas, kuriame negalima pasiekti „kernel memory“, taip pat ribojamos kai kurios instrukcijos.

Procesorių sudaro šie registrai:

1. C – loginis trigeris, naudojamas loginio valdymo perdavimo operacijose. Toliau bus surašomi šio registro flag‘ai (1B)
   1. EF – rodys paskutinio paliginimo rezultatą ar operandai buvo lygus.
   2. LF – rodys paskutinio paliginimo rezultatą ar pirmasis operandas buvo mažesnis už antrąjį.
   3. GF – rodys paskutinio paliginimo rezultatą ar pirmasis operandas buvo didesnis už antrąjį.
   4. OF – rodys paskutinio operacijos rezultatą ar įvyko „overflow“.
2. MODE – nurodo CPU veikimo režimą. (1B)
3. CHAN – nurodys kurie kanalai ir užimti flag‘ų principų. 0x1 reikš pirmas kanalas užimtas, 0x2 – antras kanalas, 0x4 – trečias kanalas. (1B)
4. IC (Instruction counter) – instrukcijų skaitliukas, nurodo sekančios instrukcijos adresą atmintyje. (4B)
5. SC (Stack counter) – steko skaitliukas, nurodo esamo steko adresą atmintyje. (4B)
6. PR (Page register) – nurodo MMU (Page Table) pradžia atmintyje. (4B)

MMU (Memory management unit) – sudarytas iš „Page Table“, kuriame laikomi „Page table entry“ (virtualios atminties adresai į fizinę). Leidžia versti virtualios atminties adresus į fizinės atminties adresus.

Timer – įrenginys, kuris fiksuotais laiko intervalais generuos patraukimas.

**RAM** (Random access memory) – fizinė atmintis sudaro:

1. „Kernel memory“ – atmintis skirta „Kernel“ MODE naudojimui. Šiame projekte toliau ji nebus datalizuojama, nes naudojam tik OS programoje.
2. „User memory“ – atmintis skirta „User“ MODE, tačiau pasiekiama ir „kernel“ režime. Dažniausiai naudojama procesuose.

**Channels** – kontroliuoja išorinių ir vidinių įtaisų kanalų būsena, kas užtikrina determinuota resursų pasiekimą. Vėliau bus naudojama operacinių sistemų kontekste, kai reiks užtikrinti, kad kelios programos vienu metu nebandytu imti to paties resurso.

**Keyboard** – įvesties įrenginys, leidžiantis nustatyti klavišų būsena.

**Monitor** – išvesties įrenginys, leidžiantis paduoti simbolių eilutę ir vaizduoti ją ASCII formatų ekrane.

# Virtualios mašinos aprašas

## Procesorius

Virtualios mašinos procesorius bus beveik toks pat kaip realios tik paprastesnis, išimant nereikalingus komponentus. Šio procesoriaus pirminis tikslas vykdyti instrukcijas kuriuos yra virtualiojo atmintyje. Registrai kuriuos naudos virtuali mašina:

1. IC (Instruction counter) – instrukcijų skaitliukas, nurodo sekančios instrukcijos adresą atmintyje. (4B)
2. SC (Stack counter) – steko skaitliukas, nurodo esamo steko adresą atmintyje. (4B)

## Atmintis

Virtualios mašinos atmintis bus iš 16 page‘ų po 16 WORD, o WORD sudarys 4 baitus. Segmentai turės išlaikyti eiliškumą pirma eis DATA segmentas (kuriame bus saugomi statiniai kintamieji), toliau STACK segmentas ir pabaigoje (CODE segmentas (kur bus saugomas proceso kodas). Kaip anksčiau minėta ryši tarp virtualios ir fizines atminties bus nusakoma per CPU įrenginį MMU.

## Virtualios mašinos instrukcijos

1. Aritmetinės darbo su sveikaisiais skaičiais instrukcijos: (4B)
   1. ADD – ištraukia iš steko du skaičius bei suma įstumia atgal į steką. ([SC] = [SC] + [SC – 1])
   2. SUB – ištraukia iš steko du skaičius bei atimti įstumia atgal į steką. ([SC] = [SC] - [SC – 1])
   3. MUL – ištraukia iš steko du skaičius bei sandauga įstumia atgal į steką. ([SC] = [SC] \* [SC – 1])
   4. MUL – ištraukia iš steko du skaičius bei dalybos rezultatą įstumia atgal į steką. ([SC] = [SC] / [SC – 1])
   5. AND – ištraukia iš steko du skaičius bei atlieka logine AND operacija su jais, rezultatą įstumia atgal į steką. ([SC] = [SC] & [SC – 1])
   6. OR – ištraukia iš steko du skaičius bei atlieka logine OR operacija su jais, rezultatą įstumia atgal į steką. ([SC] = [SC] | [SC – 1])
   7. CMP – ištraukia iš steko du skaičius bei atimti įstumia atgal į steką. ([SC] = [SC] - [SC – 1])
2. Darbo su duomenimis komandos:
   1. LDI X – įstumia į steką reikšmę atminties adrese X.
   2. STI X – ištraukia iš steko reikšmę ir patalpina atminties adrese X.
   3. LDC X – įstumia į steką konstanta X.
3. Pertraukimai:
   1. INT 0 – ištraukia iš steko skaičių ir patikrina ar klaviatūroje nuspaustas mygtukas su iš steko ištrauktu skaičiaus indeksu.
   2. INT 1 – ištraukia iš steko skaičių ir panaudoja ji kaip adresą išvedant visus simbolius, nutraukiama sutikus 0.
   3. INT 2 – atidaro filą. Stack pop = accessFlag, stack pop = filepath, stack push = fileHandle.
   4. INT 3 – uždaro filą. Stack pop = fileHandle.
   5. INT 4 – skaito iš filo. Stack pop = size, stack pop = address to write, stack pop = fileHandle.
   6. INT 5 – rašo į filą. Stack pop = size, stack pop = address to write, stack pop = fileHandle.
   7. Sąlyginio/nesąlyginio valdymo perdavimo komanda. Ištraukiamas skaičius iš steko ir jos reikšme panaudojama valdymo perdavimui.
      1. JMP – nesąlyginis.
      2. JME – sąlyginio, jai steko viršus == 0.
      3. JML – sąlyginio, jai steko viršus neigiamas.
      4. JMLE – sąlyginio, jai steko vir6us neigiamas arba == 0.
   8. HALT – programos pabaiga.

## Virtualios mašinos interpretuojamo failo formatas

Virtualios mašinos išeities kodas bus interpretuojamas. Kiek išskiriama data segmentui bus nustatoma iš išeities kodo, nurodyti kintamąjį data segmentui reikia rašyti .FLD <LABEL> <DATA> (Jų gali būti daugybe). Toliau steko segmento didis nurodomas su .STACK <SIZE IN PAGES> ir pabaigoje kodo segmentas taip pat nustatomas iš išeities kodo.

Štai mažas pavyzdys išvedantis į ekraną „hello world“:

.FLD message x68, x65, x6c, x6c, x6f, x20, x77, x6f ,x72, x6c, x64, x0

.STACK 5

LDC message // Pushes message address

INT 0

HALT

// Output:

// hello world

## Virtualios mašinos bendravimo su įvedimo/išvedimo įrenginiais aprašymas

Virtuali mašina su išvedimo ir įvedimo įrenginiais bendraus per pertraukimų sistema.

* Bendravimas su įvedimo įrenginiu – reikia naudoti pertraukimo instrukciją INT 1, kuri vėliau realioj mašinoje bus interpretuojama su C biblioteka.
* Bendravimas su išvedimo įrenginiu – reikia naudoti pertraukimo instrukciją INT 0, kuri vėliau realioj mašinoje bus interpretuojama su C biblioteka.

# Virtuali mašina visos operacinės sistemos kontekste

Virtuali mašina bus plėtojama su operacine sistema, t.y. procesas kuris simuliuos operacines sistemos darbą, „kernel“ MODE režime. Operacinė sistema užtikrins šių virtualių mašinų sukūrimą, bei resursų padalinimą. Visos virtualios mašinos dirbs po truputi, taip sukurdama lygiagretaus programų veikimo iliuzija.