

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS  
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS  
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

# **Asemblerinis programavimas**

KURSINIS DARBAS

## **BILIETŲ PIRKIMAS**

PARENGĖ: II-06/03 GR. STUD.  
KĘSTUTIS STANKEVIČIUS,  
TIKRINO: LEKT. JANINA GALKAUSKAITĖ

VILNIUS 2009

# Turinys

<u>Turinys.....</u>	<u>2</u>
<u>Užduotis.....</u>	<u>3</u>
<u>Priemonės užduočiai įvykdyti.....</u>	<u>4</u>
<u>Programos sąsaja.....</u>	<u>5</u>
<u>Programos funkcionalumo aprašymas.....</u>	<u>6</u>
<u>Programos pavyzdžiai.....</u>	<u>7</u>
<u>Išvados.....</u>	<u>8</u>
<u>ASM kodas.....</u>	<u>9</u>

## Užduotis

Sukurti Jūsų pačių nustatyto funkcionalumo programą, tenkinančią sekančius reikalavimus:

1. Pradiniai duomenys turi būti saugomi byloje arba įvedami iš klaviatūros. Atitinkamai rezultatai turi būti išsaugomi byloje arba išvedami į ekraną.
2. Apdorokite galimas klaidas: išveskite į ekraną/duomenų failą atitinkamus pranešimus apie įvykusias klaidas.
3. Skaičiavimai turi būti atliekami ne tik su teigiamais, bet ir su neigiamais skaičiais.
4. Laikinam duomenų išsaugojimui naudokite dėklinę atmintį (stack).
5. Programoje naudokite sąlyginius/besąlyginius ciklus (loop).
6. Atlikite skaičiavimus su duomenimis daugiamačiuose duomenų masyvuose (multidimensional data arrays).
7. Naudokite paprogrames/mikroprogramas pasikartojančių komandų grupėms programoje aprašyti.

## Priemonės užduočiai įvykdyti

Kursiniam darbui parašyti panaudojau Notepad2, kuris turi galimybę atpažinti .ASM failus ir juos spalvinti, panaudojau savo parašytą .BAT failą kompiliuoti kursinio darbo kodą į .EXE failą, kuris iškviečia TASM.EXE (Turbo assembler) ir TLINK.EXE Windows DOS aplinkoje ir sutransliavus paleidžia programą.

TD.EXE (Turbo debugger) naudoju klaidoms assemblerio kode ieškoti, nes su TD.EXE yra galimybė vykdyti kodą po žingsnį ir matyti registrus, vėliavėles ir operatyvią atmintį. ALAB.EXE nenaudoju kaip per laboratorinius darbus, nes jis neveikia Windows Vista aplinkoje.

### **.BAT failas** Kompiliuoti.bat KST\_KURS

```
@echo off

title MANO ALAB

if "%1" == "" goto PABAIGA

:KOMPILIUOTI

TASM.EXE /zi/l %1.ASM

TLINK.EXE %1.OBJ

%1.EXE

:PABAIGA

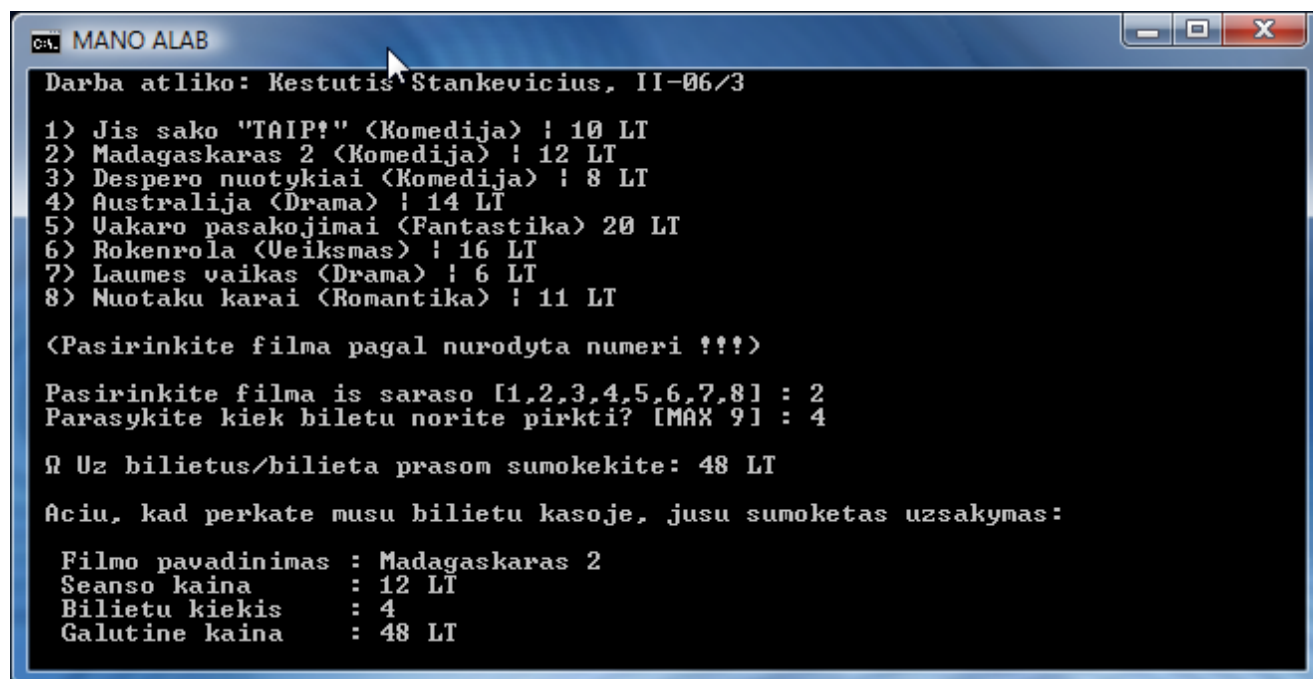
echo.
echo.
echo.

echo Spauskite, bet koki klavisa..

pause > NUL

cls
```

## Programos sąsaja



## **Programos funkcionalumo aprašymas**

Programa skirta nusipirkti bilietą arba daugiau bilietų į filmą kinoteatre, pasirenkant filmą ir bilietų skaičių bus paskaičiuota galutinė suma priklausomai nuo to filmo bilieta kainos ir bilietų kiekio. Leidžiama pasirinkti iš 8 dabar kinoteatruose rodomų filmų, nurodyti jų pavadinimai, žanras ir kaina. Pasirinkti galima įvedus į konsolę vieną skaičių nuo 1 iki 8, bet kokie kiti simboliai yra neleidžiami, jeigu bandysite įvesti raidę, 0 arba 9 gausite klaidą ir galimybę pakartoti savo pasirinkimą.

Teisingai pasirinkus filmą, filmo numeris yra išsaugojamas atmintyje ir tada jūs turite pasirinkti bilietų kiekį, įvesti galima tik skaičius, nuo 0 iki 9, jokie kiti simboliai neleidžiami, uždėtas apribojimas, kuris neleidžia per vieną kartą nupirkti daugiau, nei 9 bilietus į filmo seansą.. Padarius klaidą, t.y. įvedus tarkim raidę, bus gautas klaidos pranešimas ir jūs galėsite iš naujo pasirinkti bilietų kiekį tam pačiam savo pasirinktam filmui.

Teisingai nurodžius bilietų kiekį, bilietų kiekis išsaugojamas atmintyje, tada surandama to filmo konkreti kaina į tą seansą ir paskaičiuojama jūsų galutinė suma ir duodamas pranešimas kiek jums viskas kainuos, paspaudus bet koki mygtuką jūs patvirtinate mokėjimą ir gaunate savo bilieta informaciją su duomenimis: Filmų pavadinimas, seanso kaina. bilietų kiekiu ir galutine suma.

Padarytas nereikalingų nulių pašalinimas, tai yra: 008 LT bus 8 LT, 080 LT bus 80 LT .. o triženklis skaičius 129 LT taip ir atrodys, be to nuliai nebus šalinami iš tokių skaičių kaip 100 LT

Filmų kainos yra saugomos masyvo principu, pagal filmo numerį gaunama kainos reikšmė iš kintamojo/masyvo pavadinimu „kainos“.

Programa naudoja procedūras vienodos instrukcijoms vykdyti.

## Programos pavyzdžiai

```
film1      DB 'Jis sako "TAIP! "', '$'
film2      DB 'Madagaskaras 2', '$'
film3      DB 'Despero nuotykių', '$'
film4      DB 'Australija', '$'
film5      DB 'Vakaro pasakojimai', '$'
film6      DB 'Rokenrola', '$'
film7      DB 'Laumes vaikas', '$'
film8      DB 'Nuotaku karai', '$'

kainos     DB 10,12,8,14
           DB 20,16,6,11

;##### FILMO PASIRINKIMAS IR FILMO KAINOS NUSTATYMAS #####

; ivedimas
LEA DX, iverstis1
MOV AH, 0Ah
INT 21h

MOV CX, 0
; reikalingas ciklu skaicius
MOV CL, iverstis1+2

; galima iversti tik skaiciu intervale [1;8]
CMP CL, 31h
JL Klaidal_2
CMP CL, 38h
JG Klaidal_2

SUB CL, 30h

LEA SI, iverstis1+2

MOV BX, 0
LEA BX, kainos
; ieskom filmo kainos
CIKLAS1:
    INC BX
    LOOP CIKLAS1
    DEC BX
    MOV DX, 0
    MOV DL, [BX]
    ; irasom filmo kaina
    MOV kaina, DL
    ; irasom filmo numeri
    MOV DL, [SI]
    MOV filmo_nr, DL

;##### FILMO PASIRINKIMO NUSTATYMAS #####

CMP filmo_nr, 31h
JE FILMAS1
CMP filmo_nr, 32h
JE FILMAS2
CMP filmo_nr, 33h
JE FILMAS3
CMP filmo_nr, 34h
JE FILMAS4
CMP filmo_nr, 35h
JE FILMAS5
CMP filmo_nr, 36h
JE FILMAS6
CMP filmo_nr, 37h
JE FILMAS7
CMP filmo_nr, 38h
JE FILMAS8
```

## **Išvados**

Asembleris yra mašininio lygio programavimo kalba, kurią sunku rašyti, sunku skaityti ir kitam programuotojui sunku suprasti. Rašydamas šį kursinį darbą supratau kodėl taip yra ir kad tokia paprastą programą kaip ši užtrunka daug ilgiau rašyti, negu su kokia kita programavimo kalba kaip C, Java, Pascal ir t.t. Bet tokios programos daug mažiau užima atminties, vietos kietam diske ir veikia daug greičiau, jeigu pasirinkti geri algoritmai. Aš supratau, kad gerai mokant programuoti assembleriu, galima mokėti daug geriau programuoti kitom programavimo kalbom, kurios nėra mašininio lygio, tai yra dėl kodo optimizacijos ir algoritmų pasirinkimo.



## ASM kodas

```
;
; Kestutis Stankevicius, II-06/3
;
; Asemblerinis programavimas, Kursinis darbas.
;
; 2009
;
; Kasos aparatas, filmu bilietams pirkti.
;

; Steko segmentas
stekas SEGMENT STACK
    DB 256 DUP(0) ; isskiriame atmintyje 256 baitus
stekas ENDS

; konstantos
CR EQU 13
LF EQU 10

; Duomenų segmentas
duomenys SEGMENT
; aprasome kintamuosius
investis1    DB 2, ?, 2 DUP(?)
investis2    DB 3, ?, 3 DUP(?)

titulinis    DB CR, LF, ' ASM Kursinis darbas - Filmu bilietu pirkimas ', CR, LF
              DB CR, LF, ' Darba atliko: Kestutis Stankevicius, II-06/3', CR, LF
              DB CR, LF, ' 1) Jis sako "TAIP!" (Komedijs) | 10 LT'
              DB CR, LF, ' 2) Madagaskaras 2 (Komedijs) | 12 LT'
              DB CR, LF, ' 3) Despero nuotykijs (Komedijs) | 8 LT'
              DB CR, LF, ' 4) Australijs (Drama) | 14 LT'
              DB CR, LF, ' 5) Vakaro pasakojimai (Fantastika) 20 LT'
              DB CR, LF, ' 6) Rokenrola (Veiksma) | 16 LT'
              DB CR, LF, ' 7) Laumes vaikas (Drama) | 6 LT'
              DB CR, LF, ' 8) Nuotaku karai (Romantika) | 11 LT', CR, LF
              DB CR, LF, ' (Pasirinkite filma pagal nurodyta numeri !!!)', CR, LF, '$'

pra1         DB CR, LF, ' Pasirinkite filma is saraso [1,2,3,4,5,6,7,8] : ', '$'
pra2         DB CR, LF, ' Parasykite kiek bilietu norite pirkti? [MAX 9] : ', '$'
pra3         DB CR, LF, CR, LF, ' ê Uz bilietus/bilieta prasom sumokekite: ', '$'
pra4         DB          ' LT', CR, LF, '$'

kld1         DB CR,LF,CR,LF, ' ( KLAIDA: Pasirinkite filma nurodydami numeri is saraso! [1;8] )', '$'
kld2         DB CR,LF,CR,LF, ' ( KLAIDA: Bilietu kiekis turi buti sudarytas tik is skaitmenu! )', '$'
kld3         DB CR,LF,CR,LF, ' ( KLAIDA: Negalima vienu kartu uzsakyti daugiau nei 9 bilietus! )', '$'

pra5         DB CR, LF, ' Aciu, kad perkate musu bilietu kasoje, jusu sumoketas uzsakymas: ', CR, LF
              DB CR, LF, ' Filmo pavadinimas : ', '$'
pra6         DB CR, LF, ' Seanso kaina : ', '$'
pra7         DB          ' Bilietu kiekis : ', '$'
pra8         DB CR, LF, ' Galutine kaina : ', '$'

film1        DB 'Jis sako "TAIP!"', '$'
film2        DB 'Madagaskaras 2', '$'
film3        DB 'Despero nuotykijs', '$'
film4        DB 'Australijs', '$'
film5        DB 'Vakaro pasakojimai', '$'
film6        DB 'Rokenrola', '$'
film7        DB 'Laumes vaikas', '$'
film8        DB 'Nuotaku karai', '$'

kainos       DB 10,12,8,14
              DB 20,16,6,11

kaina        DB 2 DUP(?), '$'

bilietu_sk   DB 0, '$'
filmo_nr     DB 0, '$'
```

```

suma          DB 3 DUP(' '), '$'

desimt        DB 10

duomenys ENDS

; Programos segmentas
programa SEGMENT
    ASSUME CS:programa, DS:duomenys, SS:stekas

START:

; pradzia
MOV AX, duomenys
MOV DS, AX

; titulinis pranesimas
CALL Pagrindinis

PRAN1:
; pranesimas1
MOV AH, 09h
LEA DX, pral
INT 21h

; ivedimas
LEA DX, iverstis1
MOV AH, 0Ah
INT 21h

MOV CX, 0

; reikalingas ciklu skaicius
MOV CL, iverstis1+2

; galima iversti tik skaiciu intervale [1;8]
CMP CL, 31h
JL Klaidal_2
CMP CL, 38h
JG Klaidal_2

SUB CL, 30h

LEA SI, iverstis1+2

MOV BX, 0
LEA BX, kainos
; ieskom filmo kainos
CIKLAS1:
    INC BX
    LOOP CIKLAS1

DEC BX
MOV DX, 0
MOV DL, [BX]
; irasom filmo kaina
MOV kaina, DL
; irasom filmo numeri
MOV DL, [SI]
MOV filmo_nr, DL

PRAN2:
; pranesimas2
MOV AH, 09h
LEA DX, pra2
INT 21h

; ivedimas
LEA DX, iverstis2

```

```

MOV AH, 0Ah
INT 21h
; ciklu skaicius
MOV CL, ivaldymas+1
MOV CH, 0
MOV AH, 0
LEA SI, ivaldymas+2
MOV BH, [SI]
; tikrinam ar skaicius ivaldymas
CALL Ar_Skaicius

MOV AL, BH
DEC CL
CMP CL, 0
JE TOLIAU

CIKLAS2: INC SI
        MOV BH, [SI]
        ; tikrinam ar skaicius ivaldymas
        CALL Ar_Skaicius

        MUL desimt
        ADD AL, BH
        JO Klaida2_2
        LOOP CIKLAS2

JMP TOLIAU

KLAIDA1_2: JMP KLAIDA1_1
KLAIDA2_2: JMP KLAIDA2_1

TOLIAU:
; tikrinam ar neivesta per 9 bilietu vienetus
CMP AL, 09h
JG KLAIDA3_1
; skaiciavimai: kaina*bilietu_sk=galutine_kaina
MOV bilietu_sk, AL
MUL kaina
MOV suma, AL

; konvertuojam
DESIMTAINE:
MOV AH, 0
; sveika dalis
DIV desimt
MOV suma+2,AH
ADD suma+2,30h
MOV AH, 0

DIV desimt
MOV suma+1,AH
ADD suma+1,30h
MOV AH, 0

DIV desimt
MOV suma,AH
ADD suma,30h

MOV AL, kaina
MOV AH, 0

DIV desimt
MOV kaina+1,AH
ADD kaina+1,30h
MOV AH, 0

DIV desimt
MOV kaina,AH
ADD kaina,30h

ADD bilietu_sk,30h

```

```

; pasalinti nereikalinga 0
CMP kaina, 30h
JE NULIS1
JMP NULIAI
NULIS1:
    MOV kaina, 07h
    JMP NULIAI

KLAIDA3_1: JMP KLAIDA3
KLAIDA2_1: JMP KLAIDA2
KLAIDA1_1: JMP KLAIDA1

NULIAI:
; pasalinti nereikalingus 0
CMP suma, 30h
JE NULIS2
JMP REZULTATAS
NULIS2:
    MOV suma, 07h

    CMP suma+1, 30h
    JE NULIS3
    JMP REZULTATAS
NULIS3:
    MOV suma+1, 07h

JMP REZULTATAS

; atvaizduojam viska
REZULTATAS:
; sumokekite
MOV AH, 09h
LEA DX, pra3
INT 21h

    MOV AH, 09h
    LEA DX, suma
    INT 21h

    MOV AH, 09h
    LEA DX, pra4
    INT 21h

; paspausti bet koki klavisa
MOV AH, 07h
INT 21h

; filmo pavadinimas
MOV AH, 09h
LEA DX, pra5
INT 21h
; nustatyti pavadinima
CMP filmo_nr, 31h
JE FILMAS1
CMP filmo_nr, 32h
JE FILMAS2
CMP filmo_nr, 33h
JE FILMAS3
CMP filmo_nr, 34h
JE FILMAS4
CMP filmo_nr, 35h
JE FILMAS5
CMP filmo_nr, 36h
JE FILMAS6
CMP filmo_nr, 37h
JE FILMAS7
CMP filmo_nr, 38h
JE FILMAS8

FILMAS1:
    MOV AH, 09h

```

```

    LEA DX, film1
    INT 21h
JMP SEANSAS
FILMAS2:
    MOV AH, 09h
    LEA DX, film2
    INT 21h
JMP SEANSAS
FILMAS3:
    MOV AH, 09h
    LEA DX, film3
    INT 21h
JMP SEANSAS
FILMAS4:
    MOV AH, 09h
    LEA DX, film4
    INT 21h
JMP SEANSAS
FILMAS5:
    MOV AH, 09h
    LEA DX, film5
    INT 21h
JMP SEANSAS
FILMAS6:
    MOV AH, 09h
    LEA DX, film6
    INT 21h
JMP SEANSAS
FILMAS7:
    MOV AH, 09h
    LEA DX, film7
    INT 21h
JMP SEANSAS
FILMAS8:
    MOV AH, 09h
    LEA DX, film8
    INT 21h
JMP SEANSAS

SEANSAS:
    ; seanso kaina
    MOV AH, 09h
    LEA DX, pra6
    INT 21h

    MOV AH, 09h
    LEA DX, kaina
    INT 21h

    MOV AH, 09h
    LEA DX, pra4
    INT 21h

    ; bilietu kiekis
    MOV AH, 09h
    LEA DX, pra7
    INT 21h

    MOV AH, 09h
    LEA DX, bilietu_sk
    INT 21h

    ; galutine kaina
    MOV AH, 09h
    LEA DX, pra8
    INT 21h

    MOV AH, 09h
    LEA DX, suma
    INT 21h

```

```

MOV AH, 09h
LEA DX, pra4
INT 21h

; Baigti
JMP PABAIGA

PRAN1_1: JMP PRAN1

KLAIDA1:
; klaidos pranesimas
MOV AH, 09h
LEA DX, kld1
INT 21h

JMP KLAIDA

KLAIDA2:
; klaidos pranesimas
MOV AH, 09h
LEA DX, kld2
INT 21h

JMP KLAIDA

KLAIDA3:
; klaidos pranesimas
MOV AH, 09h
LEA DX, kld3
INT 21h

JMP KLAIDA

KLAIDA:
; paspausti bet koki klavisa
MOV AH, 07h
INT 21h

CALL PAGRINDINIS
; tikrinam ar jau iverstas pirmas pasirinkimas
CMP filmo_nr, 0h
JE PRAN1_1

; pranesimas1
MOV AH, 09h
LEA DX, pral
INT 21h

MOV AH, 09h
LEA DX, filmo_nr
INT 21h

JMP PRAN2

PABAIGA:
; paspausti bet koki klavisa
MOV AH, 07h
INT 21h

; iseiti is programos
MOV AH, 4ch
INT 21h

; Procedura AR_SKAICIUS
Ar_Skaicius PROC

    CMP BH, 30h
    JL Klaida2
    CMP BH, 39h
    JG Klaida2

```

```
    SUB BH, 30h
    RET
Ar_Skaicius ENDP

; Procedura PAGRINDINIS
Pagrindinis PROC
    ; isvalyti ekrana
    MOV AX, 0002h
    INT 10h

    ; pagrindinis pranesimas
    MOV AH, 09h
    LEA DX, titulinis
    INT 21h
    RET
Pagrindinis ENDP

programa ENDS

END START
```