

FACULTATEA CALCULATOARE, INFORMATICA SI
MICROELECTRONICA

UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A
PRODUSELOR SOFT

LUCRAREA DE LABORATOR#5

Lucru in echipa. Aplicatie complexa

Autor:
Ernest BITCA

lector asistent:
Irina COJANU
lector superior:
Radu MELNIC

Lucrarea de laborator #5

1 Obiectivele lucrării

Crearea unei aplicatii complexe in echipa.
Divizarea sarcinilor pe membrii echipei.

2 Scopul lucrării de laborator

Lucrul in echipa.
Aplicarea tehnicilor, limbajelor de programare si IDE-uri studiate in laboratoarele precedente.

3 Mersul lucrării de laborator

In lucrarea de laborator s-a creat jocul ”**Corewar**”. Insiusi jocul consta in crearea unor **jucatori** care reprezinta niste **proces**e (programe). Procesele date sunt incarcate pe un sector de memorie (arena unde vor lupta jucatorii.) Scopul principal a jucatorilor este se lupte intre ei si sa ramina in viata. Fiecare jucator este scris intr-un **pseudo limbaj de programare**, prin urmare ei necesita sa fie compilati (setul de instructiuni este prezent in fisierul extern).

Pentru a compila jucatorii s-a creat un ”**compiler**” care va transla instructiunile (Fig 1) intr-un cod care va fi perceput de ”**masina virtuala**” (Fig 2).

```
1  .name "Neo"
2  .comment "Matrixa"
3
4  ld %4, r3
5  ld %20, r2
6  ld %-1, r4
7  sti r1, %:mm, %2
8  ernest: add r3, r2, r2
9  mm: live %1
10 st r4, r2
11 zjmp %:ernest
12
```

Fig 1.

```

1  00ea 83f3 4e65 6f00 0000 0000 0000 0000
2  0000 0000 0000 0000 0000 0000 0000 0000
3  0000 0000 0000 0000 0000 0000 0000 0000
4  0000 0000 0000 0000 0000 0000 0000 0000
5  0000 0000 0000 0000 0000 0000 0000 0000
6  0000 0000 0000 0000 0000 0000 0000 0000
7  0000 0000 0000 0000 0000 0000 0000 0000
8  0000 0000 0000 0000 0000 0000 0000 0000
9  0000 0000 4d61 7472 6978 6100 0000 0000

```

Fig 2.

Pentru a compila jucatorul trebuie sa avem un fisier cu extensia **.s**. In caz contrar vom primi eroare la compilare. Totodata la compilare se indica rindul in care este prezenta eroarea. Fisierul compilat se pastreaza sub format **.cor** (Fig3).



Fig 3.

Fiecare jucator are numele sau si totodata un comentariu care este indicat la inceputul programului. Exemplu de pornire a compilatorului (Fig 4).

```

C:\6\compiler\bin\Debug>compiler.exe
usage: asm input_file

C:\6\compiler\bin\Debug>compiler.exe champions/neo.s
C:\6\compiler\bin\Debug>compiler.exe champions/neoasda
Fatal: Invalid input
C:\6\compiler\bin\Debug>

```

Fig 4.

Secvente de cod(Fig 5).

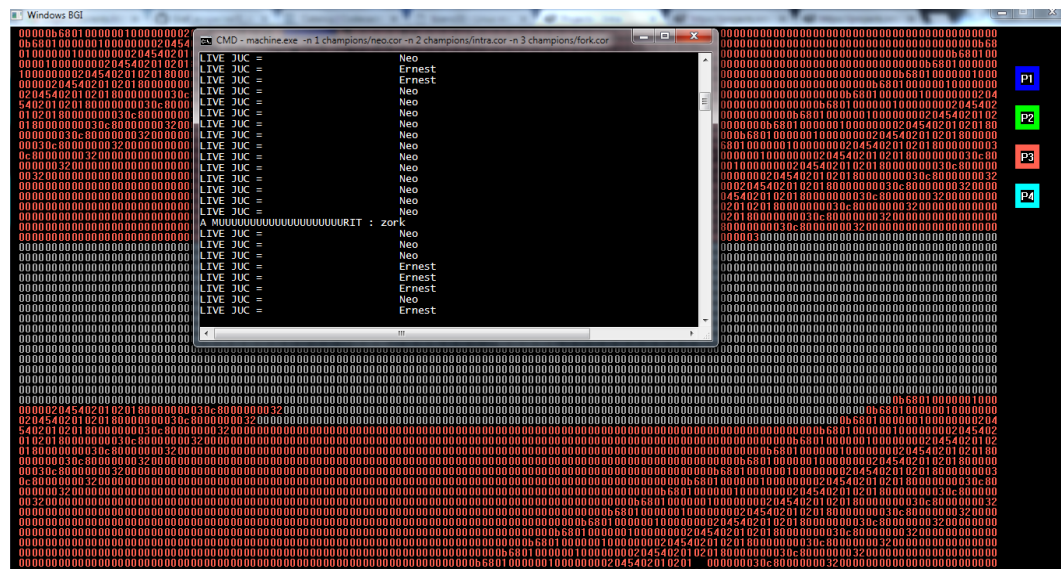
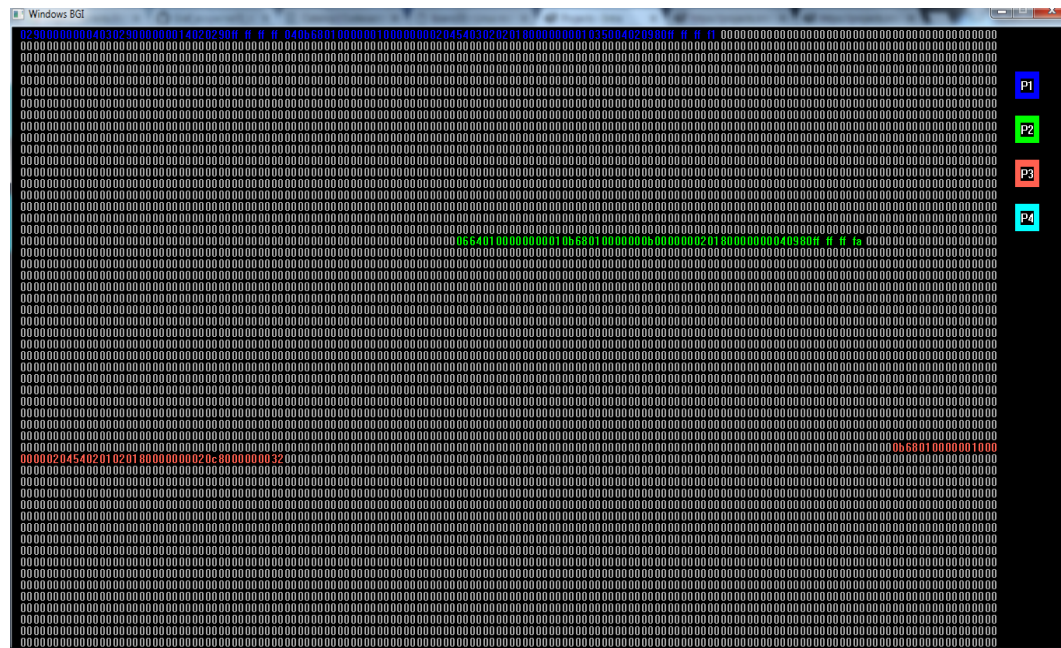
```
40 int    main(int argc, char **argv)
41 {
42     VAR;
43     ft_init();^M
44
45     if (argc != 2)
46         ft_usage();
47     g_mat = (t_mat*)malloc(sizeof(t_mat) * 1000);
48     fd = open(argv[1], O_RDONLY);
49     if (fd < 0)
50         ft_invalid_input();
51     ft_check_ext(argv[1]);
52     ft_first_check(&fd);^M
53
54     while (good_line(&fd, &line))
55     {
56         words = ft_split_whitespaces(line);
57         g_offset += ft_strlen(line) + 1;
58         ft_check_words_1(words, 1);
59         ft_skip_spaces(&fd);
60         free(words);^M
61         free(line);^M
62         printf("ok\n");
63     }^M
```

Fig 5

4 Regulile jocului

Fiecare jucator este scris in limbajul asm modificat(RedCode) care prin Compilator se transforma in bytecode. Codul la fiecare jucator este incarcat in memoria masinei virtuale, care este apoi executat consecutiv. Scopul jocului este de a ramine in viata, pentru asta fiecare jucator trebuie sa execute instructiunea live cu nr lui. Pe langa aceasta instructiune sunt prezente si multe altele: operatiile aritmetice, de incarcare datelor in registri, de accesare si modificare a memoriei, de creare a proceselor noi. Datorita faptului ca memoria este partajata de toti jucatori, ele pot influenta portiunea de cod a altor jucatori si astfel sa le dauneze, sau chiar sa le "omoare".

5 Screenshoturi



Repartizarea lucrului

Compiler - Bitca Ernest

Masina virtuala - Crivenco Vladislav, Lascu Mihai

Vizualizare - Bitca Ernest

Concluzie

În lucrarea dată s-a creat o joacă pentru programatori. A fost împartit proiectul în mai multe părți și fiecare membru al echipei era responsabil de partea lui. Pentru dezvoltare s-a folosit ca IDE **Code Blocks**. Pentru afișarea memoriei s-a folosit biblioteca grafică din C++. Pe parcursul lucrării s-a studiat conceptul de mașină virtuală, s-a lucrat mult cu parsingul și s-au îmbunătățit calitățile de lucru în grup. În echipă se pot efectua proiecte complexe, crește viteza de lucru și totodată scade numărul de erori, deoarece toți se ajută reciproc. În concluzie putem spune că este foarte important de lucrat în echipe și de împartit taskurile rațional și efectiv între membri.