**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 5

la cursul ***„Arhitectura calculatoarelor”***

***Varianta 9***

**A efectuat:**  **St. gr. CR-221FR Serba Cristina**

**A verificat: lect. univ. Postovan Dumitru**

**Chișinău 2024**

# Scopul lucrării:

Se prezinta probleme legate de prelucrarea in limbaj de asamblare a șirurilor. Pentru aceasta se recomanda utilizarea instrucțiunilor speciale pentru tratarea șirurilor. Se prezinta de asemenea tehnici specifice limbajului de asamblare pentru realizarea conversiilor de coduri, bazate pe utilizarea tabelelor de conversie.

# Sarcina lucrării:

Se considera o tabela ale carei intrari au lungimea de un cuvint. Fiecare intrare este formata din cinci cimpuri dispuse astfel:



Sa se scrie un program care afiseaza continutul tabelei sub forma unui sir de linii, pe ficare linie fiind afisate in hexazecimal continuturile cimpurilor corespunzatoare unei intrari. Spre exemplu, daca tabela contine doar doua intrari cu structura:





se vor afisa liniile urmatoare:

3 1D 5 2 0

5 09 B 3 1

# Codul sursă:

Programul de referință:

|  |
| --- |
| .DATA SEGMENT  asc\_tbl DB '0123456789ABCDEF'  DATA ENDS  CODE SEGMENT  ASSUME cs:CODE, ds:DATA  start: mov ax,DATA  mov ds,ax  mov cx,10h  xor al,al  mov bx,OFFSET asc\_tbl  bucl: mov dh,al  xlatb  mov dl,al  mov ah,06h  int 21h  mov al,dh  inc al  loop bucl  mov ax,4c00h  int 21h  CODE ENDS  END start |

Programul de execuție:

|  |
| --- |
| INCLUDE Irvine32.inc  .data  asc\_tbl db '0123456789ABCDEF' ;Tabelul ASCII  numar dw 7C3Fh ;Numarul de referinta  masti dw 0E000h, 1F00h, 00F0h, 000Eh, 1h ;Mastile pentru fiecare grup  deplasari db 13, 8, 4, 1, 0 ;pozitia de shiftat  num\_grupuri db 5  newline db 13, 10, '$'  .code  start:  movzx cx, num\_grupuri ;Incarcam toate valorile definite in registri  mov esi, offset masti  mov edi, offset deplasari  procesare\_grupuri:  mov ax, numar ;Numarul original  mov dx, ax  mov ax, [esi] ;Masca curenta  and dx, ax ;Se aplica masca  mov cl, [edi] ;Se incarca numarul cu care se va face shift  shr dx, cl ;Se executa shift pentru a selecta doar bitii necesari  mov al, dl ;Salveaza valoarea  call ConversieSiAfisare  mov dl, ' ' ;Afisam spatiu  call WriteChar  add esi, 2 ;Trecem la urmatoarea masca  add edi, 1 ;Trecem la urmatoarea pozitie de shiftat  loop procesare\_grupuri ;Repetam pana se termina grupurile  mov edx, offset newline  call WriteString  exit  ConversieSiAfisare PROC  lea si, asc\_tbl ;Adresa efectiva a sirului ascii  movzx ax, dl ;mutam valoarea obtinuta in ax pentru a o putea folosi  xlatb ;Convertim valoarea din ax in ascii  mov dl, al ;WriteChar foloseste registrul dl  call WriteChar  ret  ConversieSiAfisare ENDP  end start |

# Fișierele listing:

|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.00.24210.0 11/27/24 22:18:03  program8.asm Page 1 - 1  INCLUDE Irvine16.inc  C ; Irvine16.inc - Include file for programs using  C ; the Irvine16.lib (Real-address mode library).  C  C ; Last update: 7/29/05  C  C .NOLIST  C .LIST  C  0000 .data  0000 30 31 32 33 34 35 asc\_tbl db '0123456789ABCDEF'  36 37 38 39 41 42  43 44 45 46  0010 7C3F numar dw 7C3Fh  0012 E000 1F00 00F0 000E masti dw 0E000h, 1F00h, 00F0h, 000Eh, 1h  0001  001C 0D 08 04 01 00 deplasari db 13, 8, 4, 1, 0  0021 05 num\_grupuri db 5  0022 0D 0A 24 newline db 13, 10, '$'  0000 .code  0000 start:  0000 0F B6 0E 0021 R movzx cx, num\_grupuri  0005 66| BE 00000012 R mov esi, offset masti  000B 66| BF 0000001C R mov edi, offset deplasari  0011 A1 0010 R mov ax, numar  0014 procesare\_grupuri:  0014 8B D0 mov dx, ax  0016 67& 8B 06 mov ax, [esi]  0019 23 D0 and dx, ax  001B 67& 8A 0F mov cl, [edi]  001E D3 EA shr dx, cl  0020 8A C2 mov al, dl  0022 E8 001C call ConversieSiAfisare  0025 B2 20 mov dl, ' '  0027 E8 0000 E call WriteChar  002A 66| 83 C6 02 add esi, 2  002E 66| 83 C7 01 add edi, 1  0032 E2 E0 loop procesare\_grupuri  0034 66| BA 00000022 R mov edx, offset newline  003A E8 0000 E call WriteString  exit  0041 ConversieSiAfisare PROC  0041 8D 36 0000 R lea si, asc\_tbl  0045 0F B6 C2 movzx ax, dl  0048 D7 xlatb  0049 8A D0 mov dl, al  004B E8 0000 E call WriteChar  004E C3 ret  004F ConversieSiAfisare ENDP  end start  Microsoft (R) Macro Assembler Version 14.00.24210.0 11/27/24 22:18:03  program8.asm Symbols 2 - 1 |

# Concluzii:

În urma realizării lucrării de laborator, am implementat un program în limbajul assembly care procesează un număr de 16 biți, împărțit în grupuri de dimensiuni variabile, și le convertește în format hexazecimal, utilizând instrucțiuni avansate de manipulare a datelor și proceduri din biblioteca Irvine32. Programul demonstrează utilizarea eficientă a operațiunilor logice (AND, SHR) și a funcțiilor de intrare/ieșire pentru a extrage și afișa valorile fiecărui grup de biți, fără a utiliza operațiuni de tip DOS. Conversia fiecărui grup de biți într-o valoare hexazecimală și afișarea acesteia în format ASCII a fost realizată cu ajutorul instrucțiunilor de procesare și a tabelului de conversie în format ASCII (folosind funcția xlatb). Programul permite manipularea dinamică a tabelei de măști și a deplasărilor, ceea ce îl face ușor de adaptat pentru alte seturi de date.