**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМ**

**ЛАБОРАТОРНА РОБОТА №1.2.**

**Тема: Програмування розгалужених процесів на Асемблері**

Виконав:

Студент групи 3СТ Чума Д. І.

Перевірив:

Асист. Ваш Ю. В.

Ужгород 2022

**Мета роботи:** Опрацювати і вивчити команди умовного та безумовного переходів, набути навички програмувати на мові Асемблер процеси з розгалуженням.

1. **Теоретичні відомості**

Всі команди умовного переходу можна поділити на три групи. У першу групу входять команди, що ставляться після команди порівняння. У їх мнемокодах за допомогою певних символів описується той чи інший результат порівняння, за яким відбувається перехід. Це такі символи: **E** – equal (рівно); **N** – not (не); **G** – greater (більше) - для чисел зі знаком; **L** – less (менше) - для чисел зі знаком; **A** – above (вище) - для чисел без знака; **B** – below (нижче) - для чисел без знака.

Далі наведені назви усіх команд умовного переходу, що використовуються після команди порівняння (через слеш вказані назвисиноніми), табл.. 2.1.

Табл. 2.1.

Команди умовного переходу, що використовуються після команди порівняння

|  |  |  |
| --- | --- | --- |
| Мнемокод | Змістовна умова для переходу | Стан прапорців для переходу |
| для будь-яких чисел | | |
| je  jne | op1=op2  op1<>op2 | zf=1  zf=0 |
| для чисел зі знаком | | |
| jl/jnge  jle/jng  jg/jnle  jge/jnl | op1<op2 op1<=op2  op1>op2  op1>=op2 | sf<>of  sf<>of або zf=1  sf=of і zf=0  sf=of |
| для чисел без знаку | | |
| jb/jnae  jbe/jna  ja/jnbe  jae/jnb | op1<op2  op1<=op2  op1>op2  op1>=op2 | cf=1  cf=1 або zf=1  cf=0 і zf=0  cf=0 |

До другої групи команд входять ті, що ставляться після команд, відмінних від команд порівняння, і ті, які реагують значення деякого прапорця, табл. 2.2.

Таблиця 2.2.

Команди умовного переходу, які реагують на те чи інше значення прапорця

|  |  |  |  |
| --- | --- | --- | --- |
| Мнемокод | Умова переходу | Мнемокод | Умова переходу |
| jz  js  jc  jo  jp | zf=1  sf=1  cf=1  of=1  pf=1 | jnz  jns  jnc  jno  jnp | zf=0  sf=0  cf=0  of=0  pf=0 |

1. **Текст програми**

Програма розроблена на мові програмування C++.

Варіант 17:

Лістинг:

#include <iostream>

using namespace std ;

int main()

{

std::cout<<("Input x, y, z")<<endl;

int x, y, z, f;

std::cin >> x >> y >> z;

// 17.

\_asm {

cmp z, 2

jl m2 // z < 2

cmp x, 3

jl m2 // x < 3

cmp x, 5

jge m1 // x >= 5

jmp m3

m1 :

cmp y, 0

jne m11 // y != 0

jmp m3

m11 :

mov eax, 5

mov ebx, y

xor edx, edx // очищення edx регістру

idiv ebx

add eax, x

jmp stop

m2 :

mov eax, 4

imul x

mov ebx, 3

xchg eax, ebx

imul z

xchg eax, ebx

sub eax, ebx

jmp stop

m3 :

mov eax, x

add eax, z

jmp stop

stop:

mov f, eax

}

std::cout << "f = " << f << endl;

}

1. **Результати роботи програми**



Рисунок 1. Виконання першої умови

Перевірка:

f = 5 / y + x;

f = 5 / 3 + 7 = 8,6.



Рисунок 2. Виконання другої умови

Перевірка:

f = 4x - 3z;

f = 4 \* 2 - 3 \* 1 = 5.



Рисунок 3. Виконання третьої умови

Перевірка:

f = x + z;

f = 4 + 4 = 8.

1. **Висновки**

На даній лабораторній роботі було ознайомлено з командами умовного та безумовного переходів, набуто навички програмувати на мові Асемблер процеси з розгалуженням. Та покращено знання використання асемблерної вставки на мові програмування C++.

1. **Відповіді на контрольні питання**
2. Опишіть команди умовного переходу.

Всі команди умовного переходу можна поділити на три групи. У першу групу входять команди, що ставляться після команди порівняння. У їх мнемокодах за допомогою певних символів описується той чи інший результат порівняння, за яким відбувається перехід. Це такі символи: **E** – equal (рівно); **N** – not (не); **G** – greater (більше) - для чисел зі знаком; **L** – less (менше) - для чисел зі знаком; **A** – above (вище) - для чисел без знака; **B** – below (нижче) - для чисел без знака.

1. Опишіть команди безумовного переходу.

Команди безумовного переходу мають одну і ту ж мнемоніку ***JMP*** і додаткову уточнюючу інформацію, визначальну дальність переходу. У командах безумовного переходу адресу нової команди може формуватися двома способами.

У першому способі в самій команді вказується зсув щодо поточної команди, яке може бути як позитивним, так і негативним. Адреса, що завантажується в програмний лічильник, утворюється шляхом складання з його поточним вмістом цього зміщення, або його віднімання. Зсув може бути задане як однобайтним числом, так і двохбайтним.

У другому способі адресу нової виконуваної команди безпосередньо завантажується в програмний лічильник. При цьому він може бути взятий з команди у вигляді двохбайтного числа, або з комірки пам'яті, адреса якої вказана в команді всіма можливими режимами адресації. Другий спосіб забезпечує перехід не тільки всередині поточного сегмента програми, але і в будь-який інший.