

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина: Архитектура компьютера

Студент: Бешкуров Михаил

Группа: НКНбд-01-18

МОСКВА

2018 г.

Цель работы

Приобретение навыков написания программ с использованием логических операций и операций побитового сдвига.

Ход работы

1. Написал программу со следующим алгоритмом:

- Вывести приглашение;
- Ввести с клавиатуры строку (предполагается, что она содержит десятичные цифры и любые буквы);
- Найти во введенной строке все цифры и для каждой найденной цифры установить в «1» в регистре ах бит, номер которого равен этой цифре;
- Вывести на экран содержимое регистра ах в виде нулей и единиц;

2. Объяснение полученного результата

Программа при вводе любой строки будет выводить последовательность из битов, по которой можно будет определить, какие числа встречаются в данной строке.

```

SECTION .data
text: DB "Input", 10
textLen: EQU $-text

SECTION .bss
buf1: RESB 80
buf2: RESB 80

SECTION .text
GLOBAL _start

Bit:
    mov eax, 0
lp1:
<----->mov al, [esi]
<----->cmp al, 10
<----->je end
<----->sub al, '0'
<----->cmp al, 0
<----->JB lp2
<----->cmp al, 9
<----->JA lp2
<----->mov cl, al
<----->mov edx, 1
<----->shl edx, cl
<----->or [buf2], edx
    lp2:
<----->inc esi
<----->jmp lp1
    end:
<----->ret

convert:
    xor ecx, ecx
    xor ebx, ebx
    mov bl, 2
    .divide:
    xor edx, edx
    div ebx
    add dl, '0'
    push dx
    inc ecx
    cmp eax, 0
    jnz .divide
    .reverse:
    pop ax
    stosb
    loop .reverse
    ret

_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, text
    mov edx, textLen
    int 80h
    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 80h
    mov esi, buf1
    call Bit
    mov eax, [buf2]
    mov edi, buf2
    call convert
    mov eax, 4
    mov ebx, 1
    mov ecx, buf2
    mov edx, 80
    int 80h
    mov eax, 1
    mov ebx, 0
    int 80h

```

Код программы

```

mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab/lab7 $ nasm -f elf32 -o lab.o lab7.asm
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab/lab7 $ ld -o lab lab.o
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab/lab7 $ ./lab
Input
1350
101011
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab/lab7 $ ./lab
Input
MISHA
0
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab/lab7 $

```

Программа работает

Вывод

Приобрёл навыки написания программ с использованием логических операций и операций побитового сдвига.

Контрольные вопросы

1. Команды сдвига и их краткая характеристика:

- shr – операнд, счетчик_сдвигов (логический (беззнаковый) сдвиг вправо);
- shl – операнд, счетчик_сдвигов (логический (беззнаковый) сдвиг влево);
- ror – циклический сдвиг вправо;
- rol – циклический сдвиг влево;
- rcr – циклический сдвиг вправо с переносом;
- rcl – циклический сдвиг влево с переносом;

2. При логическом сдвиге «освобождающиеся» биты заполняются нулями. При циклическом сдвиге «освобождающиеся» биты заполняются ушедшими битами. После нескольких циклических сдвигов регистру и флагу возвращаются их первоначальные значения, в отличие от логических сдвигов.

3. Умножение и деление можно заменить командами сдвига. Сдвиг влево на один двоичный разряд часто используется для удваивания чисел, а сдвиг на один разряд вправо — для деления на 2. Эти операции осуществляются быстрее, чем команды умножения или деления.

4. Логические команды и их краткие характеристики:

or – если хотя бы один из сравниваемых битов равен 1, то результат равен 1; если сравниваемые биты равны 0, то результат — 0.

and – если оба из сравниваемых битов равны 1, то результат равен 1; во всех остальных случаях результат — 0.

xor – если один из сравниваемых битов равен 0, а другой 1, то результат равен 1; если сравниваемые биты одинаковы (оба — 0 или оба — 1), то результат — 0.

5. Команду mov ax,0 можно заменить командой xog ax,ax.