**Упакованные данные**

Дана дата:

y db 99

m db 12

d db 31

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| year | | | | | | | month | | | | day | | | | |
| 7 | | | | | | | 4 | | | | 5 | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

15 9 8 5 4 0

Для выделения, например, месяца, накладываем маску

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**and**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

В результате, в битах, в которых записан месяц останется его значение, а в остальных битах – нули. Остается, только сдвинуть полученный результат на 5 битов вправо.

**МНОЖЕСТВА**

Еще одним из примеров представления битовых строк в программировании является представление подмножества из конечного числа исходных элементов. Имеется конечное множество объектов (например, студенты) и программа должна дать возможность представлять *подмножество* этого множества: кто из студентов в настоящее время находится в институте.

Самое естественное представление для подмножества множества из N элементов – это область памяти, содержащая N двоичных разрядов.

Например, если во множество могут входить 512 студентов с идентификационными номерами от 0 до 511, то нам потребуется 512 разрядов, т.е . 64 однобайтовые ячейки, где каждому из N возможных элементов прописывается один разряд, который будет равен 1, если соответствующий элемент входит в подмножество (студент находится в институте) и 0 в противном случае (студент отсутствует).

Например: N=48

Чтобы хранить такое множество, мы опишем массив из 3 “слов”.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | |  | | | | | | | | | | | | | | | |  | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

0 47

Пусть нам нужно обратиться к студенту с идентификационным номером X=27

Для этого разделим X на 16 (количество бит в каждом элементе) и частное это – номер элемента (), а остаток – номер бита в элементе.

Сделаем это с помощью логических команд и команд сдвига, поскольку 16 является степенью двойки ().

Кроме того, мы понимаем, что номер элемента меньше 3, а номер бита меньше 16 ( и то и другое помешается в 1 байт).

Итак: пусть X находится в eax.

Запишем в ah номер элемента, а в al номер бита в этом элементе.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov eax, 27

mov ebx, eax ; сохранили число в ebx

and eax, 1111b ; выделили 4 младшие разряда, наложив маску (остаток от

; деления на 16

shr ebx, 4 ; делим число, сохраненное в ebx на 16 получаем частное

mov ah, bl ; сохраняем частное в ah

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Таким образом, в al у нас номер бита, а в ah номер элемента.

Либо воспользуемся тем, что X помещается в слово.

Первые (правые) 4 бита это остаток от деления, а с 5 начинается частное.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

shl eax ,4 ; переместили частное в ah

shr al, 4 ; вернули остаток на место

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Теперь, давайте, внесем студента с идентификационным номером 27 во множество.

section .bss

s resw 3

section .text

\_start:

;;;;;;;; обнуляем все биты множества ;;;;;;;;;;;;;;;;

xor ax,ax

xor ebx, ebx

move ecx, 3

l: mov [s+ebx\*2], ax

inc ebx

loop l

move eax, 27 ; идентификационный номер студента

mov cl, al ; получаем номер бита

and cl, 1111b ; в регистре cl

mov ebx, 1 ; создаем маску

shl ebx, cl ; в регистре ebx

mov edx, eax ; вычисляем номер элемента

shr edx,4 ; в регистре dx

or [s+edx\*2], ebx ; применяя маску, записываем 1 в бит, соответствующий

; студенту с идентификационным номером 27, не изменяя

; остальные биты