DidadoY

Previo

1. Explicad que significan y para qué sirven los parámetros “argc “argv”del main.

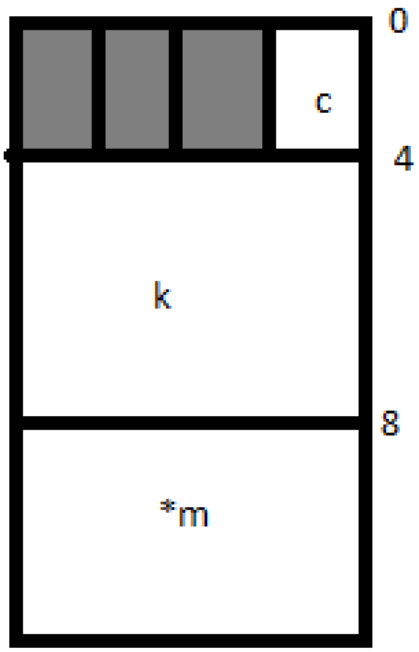
“argv” es un array que contiene los argumentos que se han pasado desde el sistema operativo al invocar el programa, mientras que “argc” es un entero que contiene el número de argumentos. Éste, como mínimo, valdrá 1 pues el nombre del programa ya se toma como primer argumento.

2. Explicad para qué sirve la función atoi

“atoi” sirve para convertir un string o char a un entero

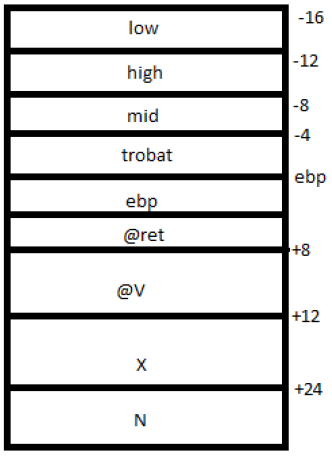
3. Dibujad el struct S1:

```
typedef struct {
  char c;
  int k;
  int *m;
}S1;
```

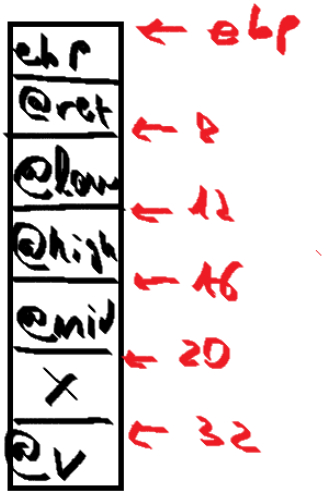


4. Dibujad el bloque de activación de las rutinas Buscar y BuscarElemento.

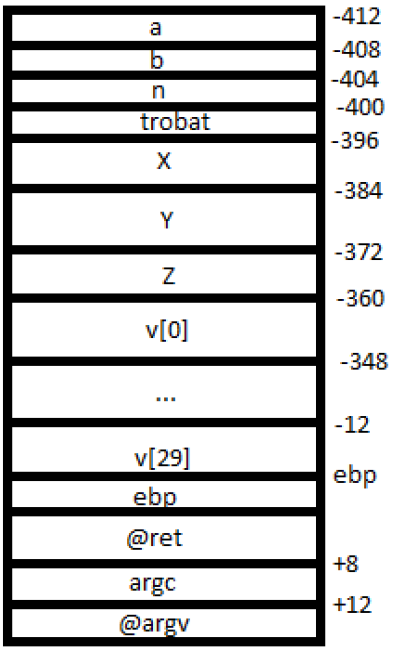
Buscar



BuscarElemento



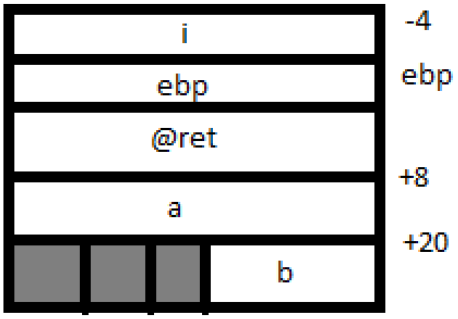
5. Dibujad el bloque de activación de la rutina main.



6. Dibujad el bloque de activación y traducid el siguiente código en ensamblador del x86:

```
int SencillaSub(Sl a, char b) {
  int i;

  if (a.c==b)
    i=0;
  else
    i=*(a.m);
  return i+a.k;
}
```

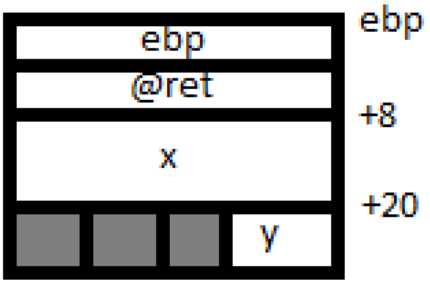


Traducción:

```
pushl %ebp
movl %esp, %ebp
subl $4, %esp
movb 8(%ebp), %al
cmpb %al, 20(%ebp)          #a.c == b
jne else
movl $0, %eax               #i=0
jmp fin
else: movl 16(%ebp), %ecx   #ecx = a.m
      movl (%ecx), %eax     #eax =*(a.m)
fin: addl 12(%ebp), %eax    #eax += a.k
     movl %ebp, %esp
     popl %esp
     ret
```

7. Dibujad el bloque de activación y traducid a ensamblador del x86 la siguiente subrutina:

```
int LlamaSencilla(Sl x, char y) {
  *(x.m)=SencillaSub(x,x.c);
  return x.k;
}
```



Traducción:

```
pushl %ebp
movl %esp, %ebp
movl 8(%ebp), %ecx   #x.c
pushl %ecx
leal 8(%ebp), %edx   #x
pushl %edx
```

```
call SencillaSub
addl $8, %esp
movl 16(%ebp), %ecx # ecx <- x.m
movl %eax, (%ecx)
movl 12(%ebp), %eax
movl %ebp,%esp
popl %ebp
ret
```