



Práctica 12

Objetivo

Seleccionar las instrucciones de control de flujo y llamadas al sistema para desarrollar aplicaciones de sistemas basados en microprocesador, mediante el análisis de su funcionalidad, de forma responsable y eficiente.

Desarrollo

1. Complete la Tabla 1 sobre los parámetros necesarios en las llamadas al sistema operativo para el manejo de archivos en Linux por medio de la interrupción 80h.

Acción sobre un archivo	Parámetros del servicio
Abrir	EAX = servicio 5 (apertura) EBX = direccion de una cadena de caracteres terminada en nulo (nombre de archivo) ECX = modo de acceso EDX = permisos
Leer	EAX = servicio 3 (lectura) EBX = direccion de una cadena de caracteres terminada en nulo ECX = buffer de entrada EDX = numero de caracteres a leer
Escribir	EAX = servicio 4 (escritura) EBX = direccion de una cadena de caracteres terminada en nulo ECX = buffer de salida EDX = numero de caracteres a escribir
Cerrar	EAX = servicio 6 (clausura) EBX = direccion de una cadena de caracteres terminada en nulo (nombre de archivo)

--	--

Tabla 1. Manejo de archivos con la int 80h.

2. Cree una carpeta llamada OAC en su computadora.
3. Dentro de la carpeta OAC, cree un programa llamado **P12.asm** que contenga el código necesario para realizar lo siguiente:
 - a) Implementar la subrutina pirámide la cual recibe en **CX** el tamaño de esta, imprime con putchar desde el 1 hasta la Z mayúscula, además recibe en **ESI** la dirección de una cadena para guardar la pirámide de salida
 - b) Imprimir la cadena de salida, la impresión debe respetar la forma de la pirámide
 - c) Crear un archivo con nombre OAC_P12.txt utilizando la int80h en modo escritura solamente
 - d) Escribir en el documento abierto la salida de la pirámide guardada en la cadena
 - e) Abrir el archivo creado anteriormente utilizando la int80h en modo lectura solamente
 - f) Leer el contenido del archivo utilizando la int80h en modo escritura solamente
 - g) Cerrar el archivo utilizando la int80h
 - h) Desplegar en pantalla la lectura del archivo obtenida en el inciso f)

```
1
22
333
4444
55555
666666
7777777
88888888
999999999
AAAAAAAAA
BBBBBBBBB
CCCCCCCCC
DDDDDDDDDD
EEEEEEEEEE
FFFFFFFFFFFF
GGGGGGGGGGGGGGGG
HHHHHHHHHHHHHHHHH
IIIIIIIIIIIIIIIIII
JJJJJJJJJJJJJJJJJJ
KKKKKKKKKKKKKKKKKKK
LLLLLLLLLLLLLLLLLLLLL
MMMMMMMMMMMMMMMMMMMMM
NNNNNNNNNNNNNNNNNNNNN
OOOOOOOOOOOOOOOOOOOOO
PPPPPPPPPPPPPPPPPPPPPP
QQQQQQQQQQQQQQQQQQQQQ
RRRRRRRRRRRRRRRRRRRRR
SSSSSSSSSSSSSSSSSSSSSS
TTTTTTTTTTTTTTTTTTTTT
UUUUUUUUUUUUUUUUUUUUU
VVVVVVVVVVVVVVVVVVVVV
WWWWWWWWWWWWWWWWWWWWW
XXXXXXXXXXXXXXXXXXXXXXXXX
YYYYYYYYYYYYYYYYYYYYYYY
ZZZZZZZZZZZZZZZZZZZZZZZ
```

```
section .data
    buffer: db "1"
    newLine: db 10,13
    newLine_L: equ $-newLine
    fileName: db "OAC_P12.txt",0
section .bss
    str_piramid: resb 64
    fdOut resb 1
    fdIn resb 1
    fileBuffer resb 26
section .text

global _start:

_start:
    mov ebx,str_piramid
```

```

mov ecx,35
call piramid

;Imprimir cadena
mov eax, 4 ;servicio
mov ebx, 1 ;Entrada
mov ecx, str_piramid+4;Cadena
mov edx, [str_piramid] ;Caracteres
int 80h

; Crear archivo
mov eax,8
mov ebx,fileName
mov ecx,0777
int 80h
mov [fdOut],eax

; Escribir a archivo
mov eax,4
mov ebx,[fdOut]
mov ecx,str_piramid+4
mov edx,[str_piramid]
int 80h

; Cerrar archivo
mov eax,6
mov ebx, [fdOut]
int 80h

; Abrir archivo
mov eax,5
mov ebx,fileName
mov ecx,0
mov edx, 0777
int 80h
mov [fdIn],eax

; Leer de archivo
mov eax,3
mov ebx, [fdIn]
mov ecx, fileBuffer
mov edx, [str_piramid]
int 80h

; cerrar archivo
mov eax,6
mov ebx, [fdIn]
int 80h

; imprimir fileBuffer

```

```
mov eax, 4
mov ebx, 1
mov ecx, fileBuffer
mov edx, [str_piramid]
int 0x80

.end:
;End program
mov eax,1
mov ebx,0
int 80h

;Funcion piramid, obtiene en ecx el tamaño de la piramide
; obtiene en ebx el espacio reservado en memoria para guardar la piramide
; guarda el numero de caracteres en los primeros 32b de memoria reservada
; para obtener el numero de caracteres se utiliza la instruccion mov ecx,[ebx]
piramid:
add ebx,4
mov esi,ebx
mov al,'9'
mov edx,101h
.cicle:
.while:
pushad
mov al,[buffer]
mov [ebx],al
;Imprimir cadena
mov eax, 4 ;servicio
mov ebx, 1 ;Entrada
mov ecx, buffer;Cadena
mov edx, 1 ;Caracteres
int 80h
popad
inc ebx
dec dl
cmp dl,0
jne .while
call newLn
inc dh
push eax
mov al,10
mov [ebx],al
mov al,13
inc ebx
mov [ebx],al
pop eax
inc ebx
mov dl,dh
```

```

    cmp al,[buffer]
    je .num2letters
    inc byte[buffer]

    loop .cicle
    jmp .end
.num2letters:
    add byte[buffer],8
    loop .cicle
.end:
    sub ebx,esi
    mov [str_piramid],ebx
    ret
newLn:
    pushad
;Imprimir cadena
    mov eax, 4 ;servicio
    mov ebx, 1 ;Entrada
    mov ecx, newline;Cadena
    mov edx, newline_L ;Caracteres
    int 80h
    popad
    ret

```

```

section .data
    buffer: db "1"
    newline: db 10,13
    newline_L: equ $-newline
    fileName: db "OAC_P12.txt",0
section .bss
    str_piramid: resb 64
    fdOut resb 1
    fdIn resb 1
    fileBuffer resb 26
section .text

global _start:

_start:
    mov ebx,str_piramid
    mov ecx,35
    call piramid

;Imprimir cadena
    mov eax, 4 ;servicio
    mov ebx, 1 ;Entrada
    mov ecx, str_piramid+4;Cadena
    mov edx, [str_piramid] ;Caracteres

```

```
    int 80h

; Crear archivo
mov  eax,8
mov  ebx,fileName
mov  ecx,0777
    int 80h
mov  [fdOut],eax

; Escribir a archivo
mov  eax,4
mov  ebx,[fdOut]
mov  ecx,str_piramid+4
mov  edx,[str_piramid]
    int 80h

; Cerrar archivo
mov  eax,6
mov  ebx, [fdOut]
    int 80h

; Abrir archivo
mov  eax,5
mov  ebx,fileName
mov  ecx,0
mov  edx, 0777
    int 80h
mov  [fdIn],eax

; Leer de archivo
mov  eax,3
mov  ebx, [fdIn]
mov  ecx, fileBuffer
mov  edx, [str_piramid]
    int 80h

; cerrar archivo
mov  eax,6
mov  ebx, [fdIn]
    int 80h

; imprimir fileBuffer
mov  eax, 4
mov  ebx, 1
mov  ecx, fileBuffer
mov  edx, [str_piramid]
    int 0x80

.end:
```

```

;End program
mov eax,1
mov ebx,0
int 80h

;Funcion piramid, obtiene en ecx el tamaño de la piramide
; obtiene en ebx el espacio reservado en memoria para guardar la piramide
; guarda el numero de caracteres en los primeros 32b de memoria reservada
; para obtener el numero de caracteres se utiliza la instruccion mov ecx,[ebx]
piramid:
    add ebx,4
    mov esi,ebx
    mov al,'9'
    mov edx,101h
.cicle:
.while:
    pushad
    mov al,[buffer]
    mov [ebx],al
;Imprimir cadena
    mov eax, 4 ;servicio
    mov ebx, 1 ;Entrada
    mov ecx, buffer;Cadena
    mov edx, 1 ;Caracteres
    int 80h
    popad
    inc ebx
    dec dl
    cmp dl,0
    jne .while
    call newLn
    inc dh
    push eax
    mov al,10
    mov [ebx],al
    mov al,13
    inc ebx
    mov [ebx],al
    pop eax
inc ebx
    mov dl,dh
    cmp al,[buffer]
    je .num2letters
    inc byte[buffer]

    loop .cicle
    jmp .end
.num2letters:
    add byte[buffer],8

```



```
    loop .cicle
.end:
    sub ebx,esi
    mov [str_piramid],ebx
    ret
newLn:
    pushad
;Imprimir cadena
    mov eax, 4 ;servicio
    mov ebx, 1 ;Entrada
    mov ecx, newLine;Cadena
    mov edx, newLine_L ;Caracteres
    int 80h
    popad
    ret
```