

Interface Hardware/Software

Centro de Informática - UFPE

Paulo Maciel

prmm@cin.ufpe.br

- INT 1Ah
 - Ler o Relógio;
 - Atualizar o Relógio;
 - Ler a Data;
 - Atualizar a Data.

- Ler o contador do relógio.
 - Chama-se com:
 $AH = 00h$
 - Recebe-se em:
 $AL = 0$, se não se passaram 24h da última leitura.
 $CX =$ Parte de alta-ordem do contador.
 $DX =$ Parte de baixa-ordem do contador.

- Atualizar o contador do relógio.
 - Chama-se com:
 - AH = 01h.
 - CX = Parte de alta-ordem do contador.
 - DX = Parte de baixa-ordem do contador.

- Ler o relógio.

- Chama-se com:

- AH = 02h

- Recebe-se em:

- CH = Horas em BCD.

- CL = Minutos em BCD.

- DH = Segundos em BCD.

- DL = Milesegundos em BCD.

- Ler a data.

- Chama-se com:

- AH = 04h

- Recebe-se em:

- CH = Século (19, 20,21) em BCD.

- CL = Ano em BCD.

- DH = Mês em BCD.

- DL = Dia em BCD.

- Atualizar a data.

- Chama-se com:

- AH = 05h.

- CH = Século (19,20,21) em BCD.

- CL = Ano em BCD.

- DH = Mês em BCD.

- DL = Dia em BCD.

- INT 21h
 - Chamar uma diversidade de funções do DOS.

- Leitura do teclado com eco.
 - Chama-se com:
AH = 01h
Se um caractere ASCII foi digitado.
 - Recebe-se em:
AL = código ASCII.
Se um caracter não foi digitado, aguarda.
 - Exemplo.:
char db ?
.....
mov ah,1
int 21h
mov char,al

- Mostra um caracter em tela.
 - Chama-se com:
 - AH = 02h
 - DL = código ASCII do caracter.
 - Exemplo.:

```
mov ah,02
mov dl,'*'
int 21h
```

- Leitura de uma entrada auxiliar (serial 1).
 - Chama-se com:
AH = 03h.
 - Recebe-se em:
AL = código ASCII.
Se um caracter não foi digitado, aguarda.
 - Exemplo.:
char db ?
....
mov ah,3
int 21h
mov char,al

- Transmite caracter via saída auxiliar.
 - Chama-se com:
 - AH = 04h
 - DL = código ASCII do caracter.
 - Exemplo.:

```
mov ah,04
mov dl,'*'
int 21h
```

- Envia caracter para a impressora.

- Chama-se com:

- AH = 05h

- DL = código ASCII do caracter.

- Exemplo.:

- mov ah,05

- mov dl,'*'

- int 21h

- **Leitura do teclado sem eco (sem filtro).**
 - Chama-se com:
AH = 07h.
Se um caractere ASCII foi digitado .
 - Recebe-se em:
AL = código ASCII.
Se um caracter não foi digitado, aguarda.
Não é sensível a control-c e control-break
 - Exemplo.:
char db ?
.....
mov ah,7
int 21h
mov char,al

- **Leitura do teclado sem eco.**
 - Chama-se com:
AH = 08h.
Se um caractere ASCII foi digitado .
 - Recebe-se em:
AL = código ASCII.
Se um caracter não foi digitado, aguarda.
Sensível a control-c e control-break
 - Exemplo.:
char db ?
....
mov ah,8
int 21h
mov char,al

- Leitura do teclado com armazenamento no buffer.

- Chama-se com:

AH = 0Ah

DS:DX = apontam para o endereço do início do buffer.

- Formato do buffer:

| Byte | Conteúdo |
|------|---|
| 0 | Máximo número de caracteres a serem lidos. |
| 1 | Número de caracteres que de fato foram lidos. |
| 2+ | Cadeia de caracteres lida. |

- Ex.:

```
buff    db    81
        db    0
        db    81 dup(0)
```

```
mov ah,0ah
mov dx,seg buff
mov ds,dx
lea dx,buff
int 21h
```


- Mostra cadeia de caracter na tela.
 - Chama-se com:
 - AH = 09h.
 - DS:DX = apontam para o endereço do início da cadeia de caracter.
 - A cadeia dever ser finalizada com '\$'.
 - Exemplo.:

```
mens db 'Bom Dia','$'  
....  
mov dx, seg mens  
mov ds,dx  
lea dx, mens  
mov ah,9  
int 21h
```

- Lê a data do sistema.
 - Chama-se com:
AH = 2Ah.
 - Recebe-se em:
CX = Ano.
DH = Mês (1-12).
DL = Dia (1-31).
AL = Dia da semana (0= dom, ...7=sab).

- Atualiza a data do sistema.
 - Chama-se com:
 - AH = 2Bh.
 - CX = ano.
 - DH = Mês (1-12).
 - DL = Dia (1-31).
 - Recebe-se em:
 - AL = 0, se definido com sucesso.
 - AL = FFh, caso contrário.

- Lê o tempo do sistema.
 - Chama-se com:
AH = 2Ch.
 - Recebe-se em:
 - CH = Hora (0 – 23).
 - CL = Minutos (0-59).
 - DH = Segundos (0-59).
 - DL = Milisegundos (0 – 99).

- Atualiza o tempo do sistema.
 - Chama-se com:
 - AH = 2Dh.
 - CH = Hora (0 - 23).
 - CL = Minutos (0-59).
 - DH = Segundos (0-59).
 - DL = Milisegundos (0 - 99).
 - Recebe-se em:
 - AL = 0, se definido com sucesso
 - AL = FFh, caso contrário.

- Finalização do programa.
 - Chama-se com:
AH = 4Ch.
 - Exemplo.:
mov ax,4c00h
int 21h ; Finalização do programa.

- Cria um arquivo.

- Chama-se com:

AH = 3Ch

CX = atributo.

| Bit | Significado |
|-----|-----------------|
| 0 | somente leitura |
| 1 | invisível |
| 2 | sistema |

...

DS:DX apontam para o endereço que contem o caminho (string terminada por 0) e o nome do arquivo a ser criado.

- Retorno:

CF = 0 e AX = Manipulador, se sucesso.

CF = 1 e AX = Código de erro, caso contrário.

- EX.:

```
fname db 'C:\MYDIR\MYFILE.DAT',0
fhandle      dw      ?
```

```
mov ah,3ch
xor cx,cx; atributo normal
mov dx,seg fname
mov ds,dx
lea dx,fname
int 21h
jc erro
mov fhandle,ax
```

- Ler arquivo

- Chama-se com:

AH = 3Dh

AL = modo de acesso

0 - leitura

1 - escrita

2 - leitura e escrita

...

DS:DX apontam para o endereço que contem o caminho (string terminada por 0) e o nome do arquivo a ser criado.

- Retorno:

CF = 0 e AX = manipulador, se sucesso.

CF = 1 e AX = código de erro, caso contrário.

- EX.:

```
fname db 'C:\MYDIR\MYFILE.DAT',0
fhandle      dw      ?
```

```
mov ah,3dh
mov al,2; leitura e escrita
mov dx,seg fname
mov ds,dx
lea dx,fname
int 21h
jc erro
mov fhandle,ax
```


- Fechar arquivo
 - Chama-se com:
AH = 3Eh.
BX = manipulador.
 - Retorno
CF = 0, se sucesso.
CF = 1, caso contrário.
- EX.:
fhandle dw ?

mov ah,3Eh
mov bx,fhandle
int 21h
jc erro

- Ler arquivo

- Chama-se com:

- AH = 3Fh.

- BX = manipulador.

- CX = número de bytes a ser lido.

- DS:DX = endereço inicial do buffer.

- Retorno:

- CF = 0, se sucesso. AX = bytes lidos.

- CF = 1, caso contrário. AX = código de erro.

- Exemplo.:

- fhandle dw ?
 - buff db 1024 dup(?)

- mov ah,3Fh
 - mov dx, seg buff
 - mov ds,dx
 - lea dx,buff
 - mov bx,fhandle
 - mov cx,1024
 - int 21h
 - jc erro
 - cmp ax,cx
 - jl done

- Escrever em arquivo
 - Chama-se com:
 - AH = 40h.
 - BX = manipulador.
 - CX = número de bytes a ser escrito.
 - DS:DX = endereço inicial do buffer.
 - Retorno:
 - CF = 0, se sucesso. AX = bytes escritos.
 - CF = 1, caso contrário. AX = código de erro.

- EX.:

| | | |
|---------|----|-------------|
| fhandle | dw | ? |
| buff | db | 1024 dup(?) |


```
mov ah,40h
mov dx, seg buff
mov ds,dx
lea dx,buff
mov bx,fhandle
mov cx,1024
int 21h
jc erro
cmp ax,cx
jl done
```

- Apaga arquivo
 - Chama-se com:
 - AH = 41h
 - DS:DX apontam para o endereço que contem o caminho (string terminada por 0) e o nome do arquivo a ser apagado.
 - Retorno:
 - CF = 0 , se sucesso.
 - CF = 1 e AX = código de erro, caso contrário.

- EX.:

```
fname db 'C:\MYDIR\MYFILE.DAT',0

mov ah,41h
mov dx,seg fname
mov ds,dx
lea dx,fname
int 21h
jc erro
```

- Posicionar Ponteiro de Arquivo
 - Chama-se com:
 - AH = 42h
 - BX = manipulador.
 - AL = método de posicionamento.
 - CX:DX = deslocamento.
 - Retorno:
 - CF = 0, se sucesso. DX:AX = nova posição.
 - CF = 1, caso contrário. AX = código de erro.
 - Método
 - 0 - (absoluto) Começo do arquivo.
 - 1 - Relativo a posição atual.
 - 2 - (absoluto) Fim do arquivo.

- EX.:
fhandle dw ?

```
mov ah,42h
mov al,2h
mov bx,fhandle
mov cx,0
Mov dx,1024
int 21h
jc erro
```