

# PCS 3446 – Sistemas Operacionais

Prof. João José Neto

Estudos de Casos

Aula 16 – MS-DOS

**MS  
DOS**

# O Sistema Operacional MS-DOS

- Específico para **Intel 8088** e seus sucessores  
286, 386, 486
- Foi o sistema operacional **mais usado de todos os tempos**  
(mais de **50 milhões** de unidades instaladas)

# HISTÓRICO

# Histórico

- Os primeiros computadores pessoais baseados em **Intel 8088** executavam um interpretador autônomo da linguagem **Basic**.
- A seguir, fragmentos da listagem de um interpretador Tiny Basic para 8080, de 1976, que emulava um diálogo com um terapeuta, e outra, do famoso aplicativo conversacional Elisa, produzido entre os anos 1964 e 1966, em linguagem Basic.





- Versões posteriores adotaram um sistema operacional muito simples, chamado **CP/M**.
- O **DOS** foi resultado do atendimento de uma *encomenda da IBM à Microsoft* para que esta produzisse um sistema operacional similar ao CP/M para as suas máquinas.

# Características

- As máquinas da IBM eram **sistemas abertos** com ciclo de máquina de 4.77MHz.
- Eram **compatíveis com os aparelhos comuns** de TV colorida, e também controlavam fitas cassete de áudio para gravar e ler dados e programas.
- Sendo sistemas abertos, rapidamente apareceram fábricas e lojas de **placas e equipamentos** para serem incorporados aos PCs, e surgiram também **clones**.





# Limitações

- Foi decidido **limitar em 640KB** a memória dos PCs para programas, **apesar de** as vias de endereçamento do 8088 **comportarem até 1MB**.
- Outra característica dos PCs da IBM foi que não dispunham de **nenhum esquema de proteção** em hardware, o que causou muitos problemas aos seus usuários.

# IBM DOS 1.0

Enter today's date (m-d-y): 08-04-81

The IBM Personal Computer DOS  
Version 1.00 (C)Copyright IBM Corp 1981

```
A>dir *.com
IBMBIO      COM           1920   07-23-81
IBMDOS      COM          6400   08-13-81
COMMAND     COM          3231   08-04-81
FORMAT      COM          2560   08-04-81
CHKDSK      COM          1395   08-04-81
SYS          COM           896   08-04-81
DISKCOPY    COM          1216   08-04-81
DISKCOMP    COM          1124   08-04-81
COMP        COM          1620   08-04-81
DATE        COM           252   08-04-81
TIME        COM           250   08-04-81
MODE        COM           860   08-04-81
EDLIN       COM          2392   08-04-81
DEBUG       COM          6049   08-04-81
BASIC       COM         10880   08-04-81
BASICA      COM         16256   08-04-81
```

A>\_

Diretório típico de  
um disquete  
utilizado no IBM DOS

# O MS-DOS v1.0

## (1981, 4000 linhas em assembly)

- Ocupava **12K** dos 64K disponíveis na memória e era praticamente compatível com o CP/M
- Usava **disquetes de 5¼" de 160K**, gravados em face simples e com densidade simples (avanço em relação aos então usuais disquetes de 8")
- Usava uma **BIOS residente em ROM** para controlar operações de entrada e saída
- **Diretório simples**, sem sub-diretórios, como no CP/M
- Dava suporte a ***batch files***
- Na versão **1.1** aceitava discos com capacidade de **320K** (face dupla e densidade simples)

# MS-DOS 1.0

MS-DOS Command release 1.00, version 1.19

C:\dir

CHKDSK	COM	1754	8-16-83	12:56p
COMMAND	COM	4986	1-18-84	2:01p
CONFIGUR	COM	19724	5-03-84	10:33a
DEBUG	COM	6003	8-16-83	1:02p
DISKCOMP	COM	5344	11-11-83	1:37p
DISKCOPY	COM	5728	12-13-83	1:37p
EDLIN	COM	2313	8-16-83	12:56p
EXE2BIN	EXE	1280	8-16-83	1:01p
FILCOM	COM	8320	8-16-83	12:56p
FORMAT	COM	3856	4-24-84	3:50p
IO	SYS	1713	12-29-83	1:54p
LIB	EXE	32128	8-16-83	1:00p
LINK	EXE	41856	8-16-83	1:00p
MSDOS	SYS	6138	8-16-83	12:49p
PRINT	COM	1740	8-16-83	1:43p
RDCPM	COM	1548	12-13-83	1:28p
SYS	COM	914	1-18-84	2:50p

17 File(s)

C: \_

# Estrutura básica: 3 programas

- **ibmbio.com**  
sistema de **entrada/saída** em caracteres e disco
- **ibmdos.com**  
administrador de **disco** e de **arquivos**
- **command.com**  
processador de comandos (**shell** primitivo, interpretador de comandos de console)
- **BIOS** – pastilha de memória *read-only* pré-gravada com **drivers** dos dispositivos de E/S do sistema (mapeada nos endereços altos da memória, para não ocupar a área de RAM dos programas)



# O MS-DOS v2.0 - (1983, 20.000 linhas)

- Acompanhava o novo **PC-XT com disco rígido**
- Ainda aceitava as chamadas de sistema do CP/M
- Disquetes de **360K**
- Parecido com **UNIX** e não mais com CP/M
- Permitia executar **aplicativos grandes** e começou a ser usado em empresas
- v2.1 acompanhou o fracassado hardware PC Jr.
- A famosa **v2.11**, com suporte internacional, vendeu **milhões** de exemplares

# O MS-DOS v3.0 - (1984, 40.000 linhas)

- Acompanhou o **PC/AT** baseado em **Intel 286**
- PC/AT tinha memória de até **16MB**, modos supervisor/usuário, multiprogramação, mas o MS-DOS v3.0 **não** dava suporte a essas novas características
- Tratava disquetes de **1.2MB**, **relógio** interno com **bateria**, informações de configuração em memória **CMOS**, discos rígidos maiores que **10MB** e **RAM disks**.

# MS-DOS 3.0

```
Microsoft(R) MS-DOS(R) Version 3.30
(C)Copyright Microsoft Corp 1981-1987
```

```
A>c:
```

```
C>a:
```

```
A>dir /w
```

```
Volume in drive A is MS330APP01
Directory of A:\
```

4201	CPI	5202	CPI	ANSI	SYS	APPEND	EXE	ASSIGN	COM
ATTRIB	EXE	CHKDSK	COM	COMMAND	COM	COMP	COM	COUNTRY	SYS
DISKCOMP	COM	DISKCOPY	COM	DISPLAY	SYS	DRIVER	SYS	EDLIN	COM
EXE2BIN	EXE	FASTOPEN	EXE	FDISK	COM	FIND	EXE	FORMAT	COM
GRAFTABL	COM	GRAPHICS	COM	JOIN	EXE	KEYB	COM	LABEL	COM
MODE	COM	MORE	COM	NLSFUNC	EXE	PRINT	COM	RECOVER	COM
SELECT	COM	SORT	EXE	SUBST	EXE	SYS	COM	TREE	COM

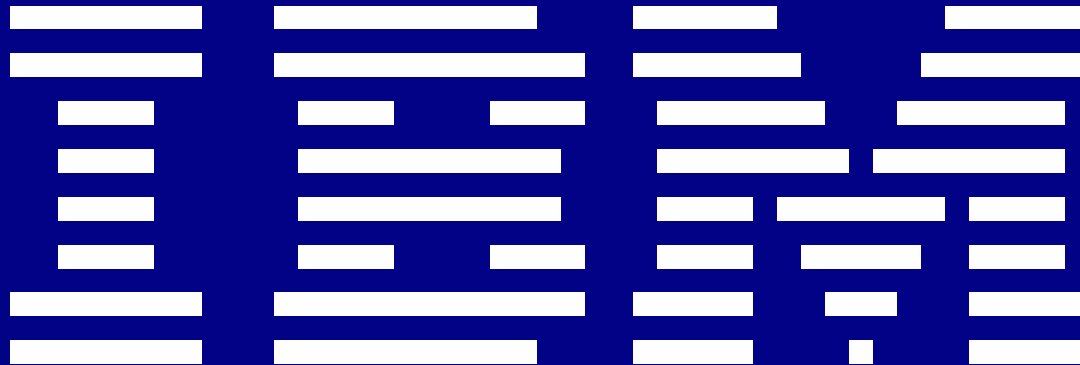
35 File(s)      1024 bytes free

```
A>_
```

- v3.1 fornecia o primeiro **suporte a redes** de computadores (só funcionava em rede para programas bem comportados)
- v3.2 suportava discos de **3½"**, mas não teve sucesso devido aos numerosos erros remanescentes
- Em 1987 a IBM lançou o **PS/2 com** a nova versão **MS-DOS 3.3**

- Lançou também o **OS/2**, sistema operacional novo para o PS/2
- Em 1991 a Microsoft abandona o OS/2 e, em represália, a IBM rompe a parceria, aliando-se à Apple Computers para o desenvolvimento do seu software

# IBM OS-2



Personal Computer

Introducing the IBM Operating System/2  
Version 1.00

(C) Copyright IBM Corp. 1981, 1987. All rights reserved.



# O MS-DOS v4.0 - (1988)

- Foi lançado pela IBM, e posteriormente relançado em versão de engenharia reversa pela Microsoft para atender os fabricantes de clones.
- Suportava tamanhos de discos **maiores que 32MB**, aceitando **até 2GB**
- **DOS-shell** (menu-driven)
- A exemplo da v3.2, a v4.0 também não teve sucesso

# MS-DOS 4.0

```
A>ver
```

```
MS-DOS Version 4.01
```

```
A>dir /w
```

```
Volume in drive A is INST V401
```

```
Volume Serial Number is 10E8-0C34
```

```
Directory of A:\
```

COMMAND	COM	4201	CPI	4208	CPI	5202	CPI	ANSI	SYS
AUTOEXEC	BAT	CONFIG	SYS	COUNTRY	SYS	DISKCOPY	COM	DISPLAY	SYS
DRIVER	SYS	EGA	CPI	FASTOPEN	EXE	FDISK	EXE	FORMAT	COM
GRAFTABL	COM	GRAPHICS	COM	GRAPHICS	PRO	HIMEM	SYS	KEYB	COM
KEYBOARD	SYS	LCD	CPI	NLSFUNC	EXE	PRINT	COM	PRINTER	SYS
RANDRIVE	SYS	REPLACE	EXE	SELECT	DAT	SELECT	EXE	SELECT	HLP
SELECT	PRT	SHARE	EXE	SMARTDRV	SYS	SYS	COM	XMA2EMS	SYS
35 File(s)		19456 bytes free							

```
A>_
```

# O MS-DOS v5.0 - (1991)

- Usava a **memória estendida**, liberando para os programas do usuário **600K** dos 640K de memória principal existentes nos endereços baixos, anteriores aos ocupados pelo sistema
- Pela primeira vez foi **comercializado em lojas**, para o consumidor, e não somente para fabricantes de computadores
- Dava suporte a disquetes de até **2.88MB**
- Foi uma versão **cuidadosamente testada**

# DETALHAMENTO TÉCNICO

# Detalhamento do MS-DOS



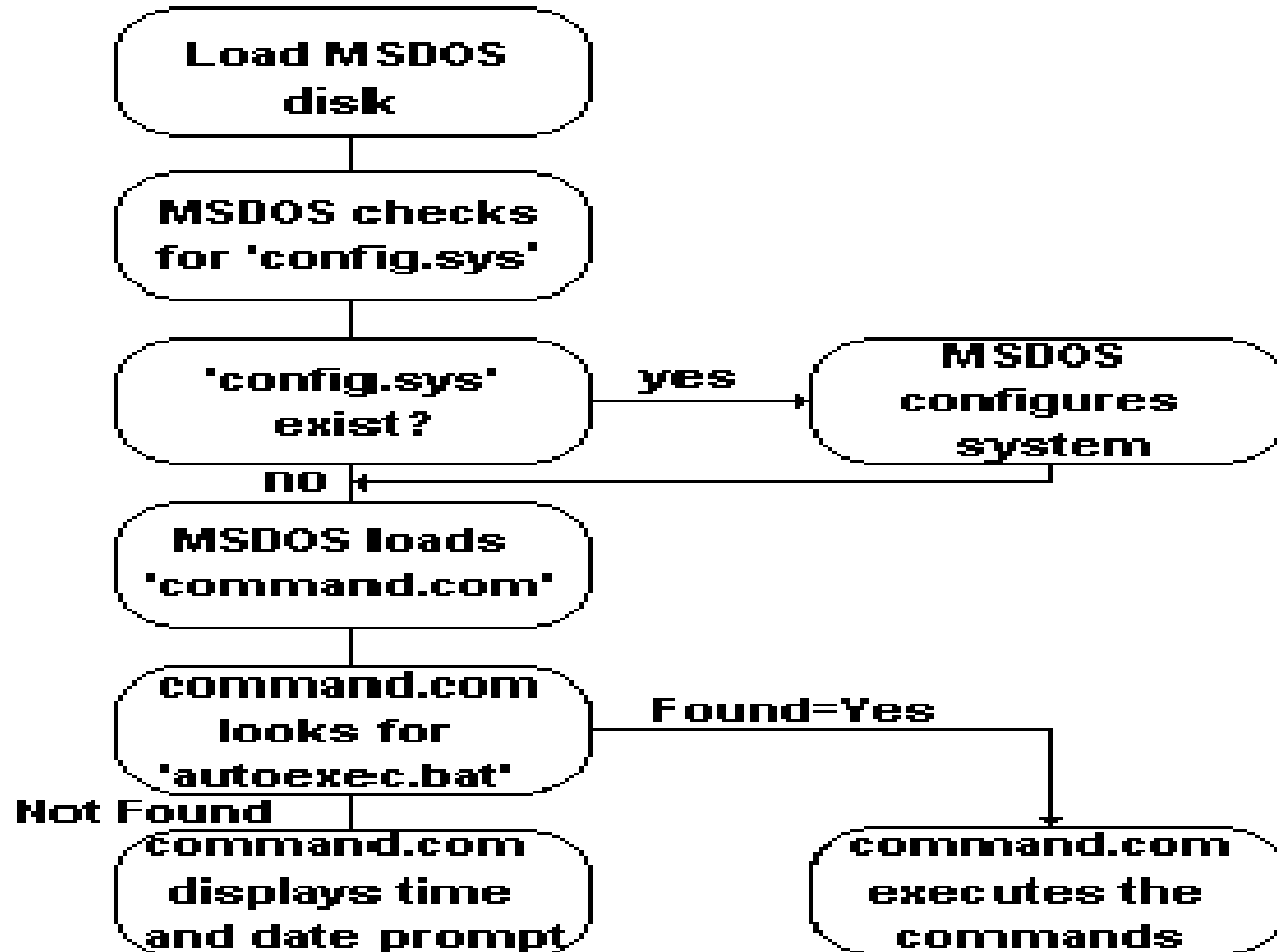
# Processos

- **MS-DOS** não é multi-, nem uni-programado, mas algo **intermediário**
- **command.com** é disparado na ocasião do boot
- Processos **pai e filho não podem ser executados concorrentemente**
- O disparo de um **processo filho bloqueia o processo pai** até que aquele seja concluído
- Programas executáveis com sufixo **.com** são **absolutos**, e com sufixo **.exe** são **relocáveis**



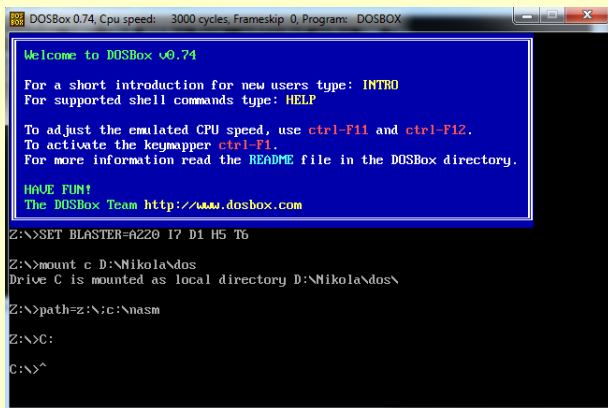
- Os **primeiros 256 bytes** do código do processo constituem o seu **PSP (*Program Segment Prefix*)**
- Por isso todos os programas **.com** começam sempre no endereço relativo 256 e não 0.
- Programas com sufixo **.exe** são relocados por sobre seu PSP, economizando esses 256 bytes.

# Processo de boot no MS-DOS

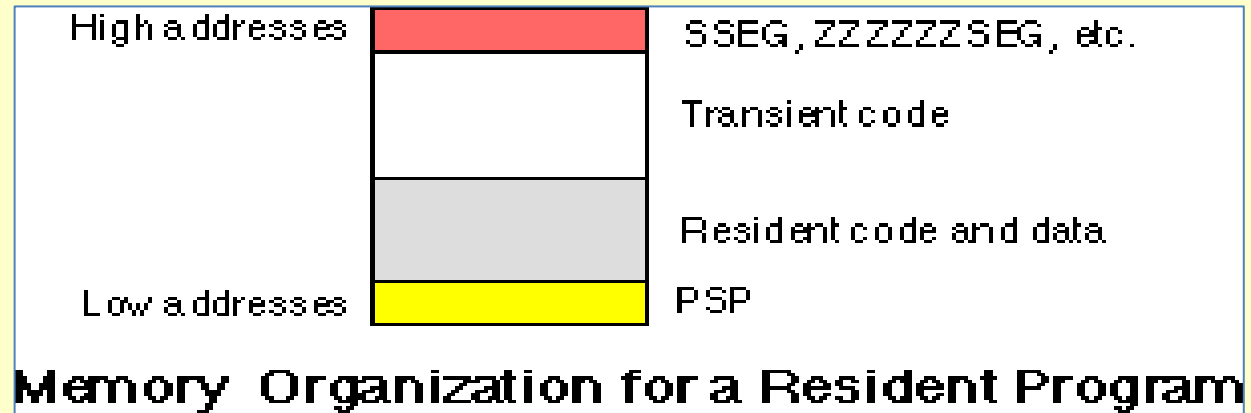


# ***Program segment prefix***

- O **PSP** contém:
  - o ***tamanho*** do programa,
  - um *ponteiro* para o bloco de ambiente,
  - o endereço do tratamento de ***control-C***,
  - a cadeia de ***comando*** passado ao programa ao ser este posto em execução,
  - um ponteiro para o ***PSP do seu processo pai***,
  - a tabela de descritores de ***arquivos*** do programa
  - outras informações.



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip: 0, Program: DOSBOX
Welcome to DOSBox v0.74
For a short introduction for new users type: INTRO
For supported shell commands type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount c D:\Nikola\dos
Drive C is mounted as local directory D:\Nikola\dos\
Z:\>path=z:\n;c:\nasm
Z:\>C:
C:\>
```



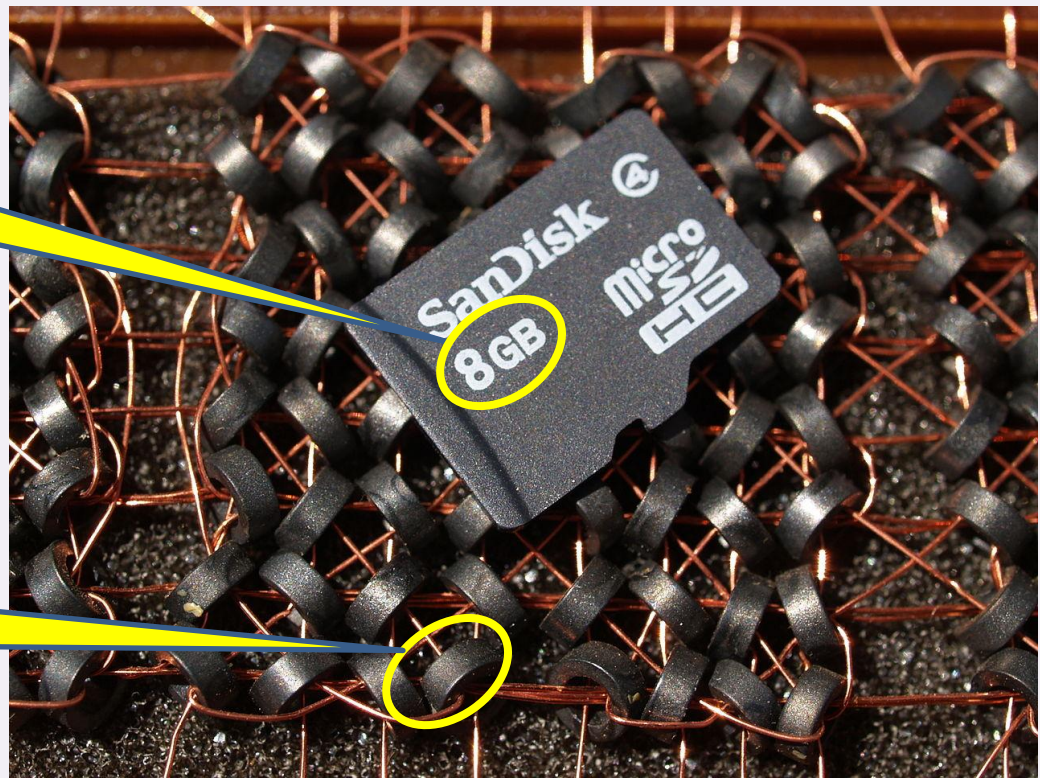
- É possível manter na memória do MS-DOS vários processos simultaneamente, cada um com PSP próprio
- Existem em MS-DOS os programas chamados **TSR (*Terminate and Stay Resident*)**, que permanecem na memória mesmo depois do final de sua execução, para serem ativados através de teclas específicas (***hot keys***)

- Programas **TSR** não podiam, originalmente, usar o MS-DOS para fazer entrada/saída de qualquer tipo porque o MS-DOS não tem rotinas reentrantes de atendimento de interrupções.
- Versões mais recentes *enganavam* o MS-DOS para que isso fosse possível, utilizando para isso, porém, técnicas não-documentadas.

# Memória

Hoje, 64GB  
ou mais

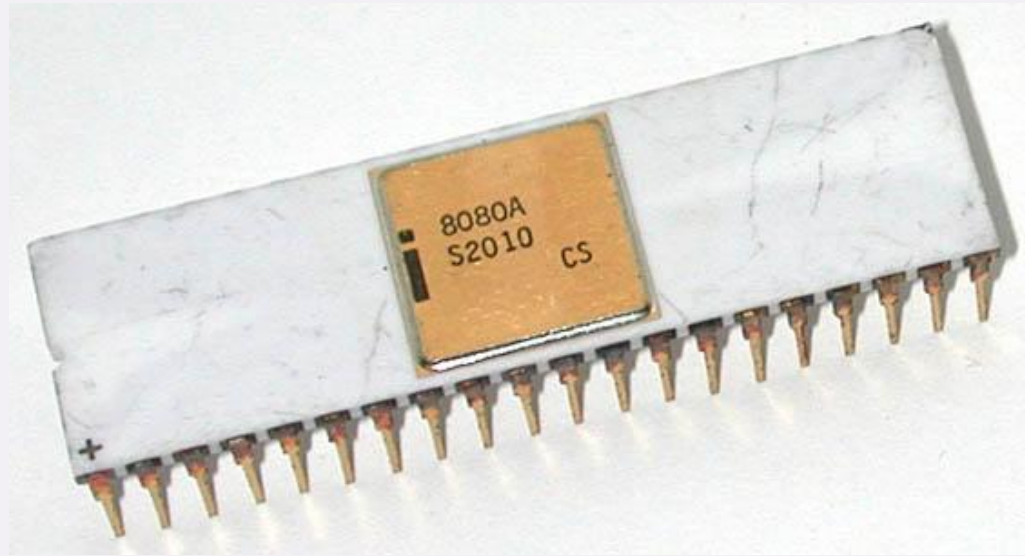
Isto corresponde  
a um bit



- Devido às características do hardware subjacente (IBM PC), o sistema operacional MS-DOS adotou um modelo de memória bastante **complexo**, com **4 regiões, de tamanhos e propriedades diversas**.



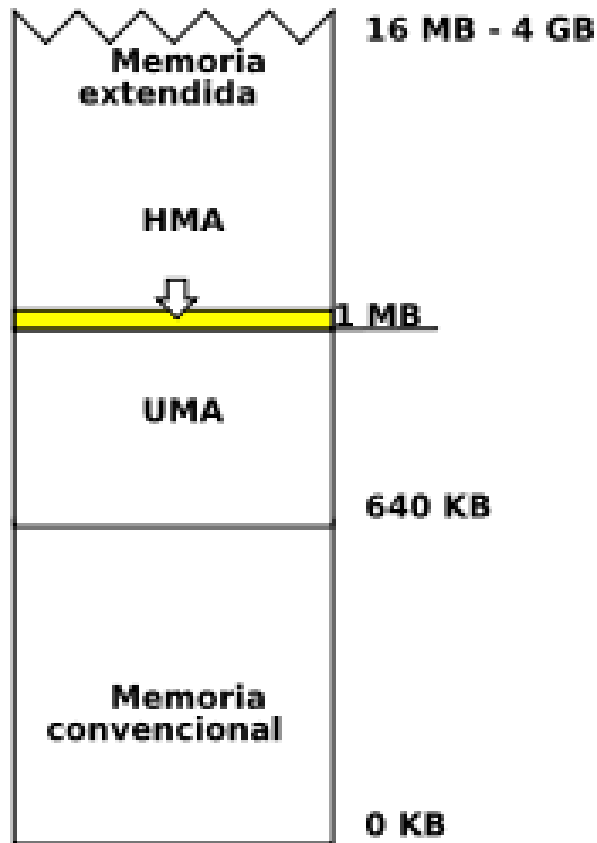
# Arquitetura da memória do 8080



- Proc. Intel, **8 bits**, do início da década de 70
- **Registradores aritméticos de 16 bits** AX,BX,CX,DX, cada qual composto de dois registradores de 8 bits (AH e AL),(BH e BL),(CH e CL),(DH e DL)
- **Registradores verdadeiros de 16 bits** [indexadores, (SI e DI), base da pilha corrente (BP) e ponteiro para o topo da pilha corrente (SP)]

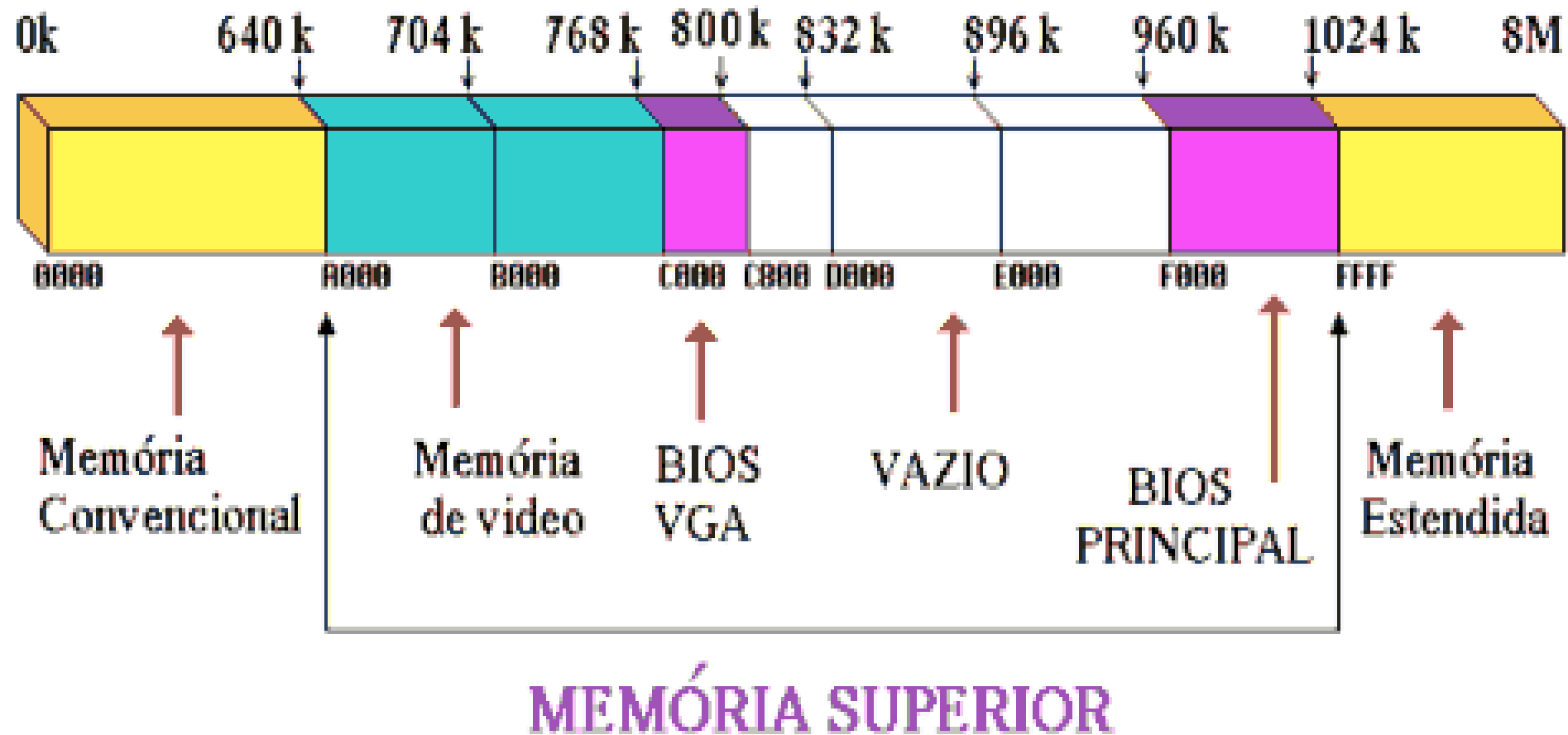
- **registradores de segmentos** (*segment registers*)  
[*code segment* (CS), *data segment* (DS), *stack segment* (SS), *extra segment* (ES)], servem para implementar um mecanismo que faz coexistir os endereços de 16 bits com o espaço de endereçamento de **1 Mbytes**.

- Cada um dos registradores de segmentos guarda os 16 mais significativos bits dos 20 que compõem os campos de endereço propriamente ditos, e com eles endereçam grupos de 16 bytes, os parágrafos.
- Os conteúdos desses registradores são usados para fazer relocação dinâmica dos respectivos segmentos.



- HMA (**high memory area**) – os 64Kbytes entre (hexa) 100000 e 10FFEF, que no 8088 são mapeados a partir de 0, ignorando o 21º bit
- EM (**extended memory**) – endereços acima de 1Mbyte (não acessíveis aos programas em modo real pois o DOS só opera em modo real)
- UMA (**upper memory area**) – os 384Kbytes entre 640K e 1M

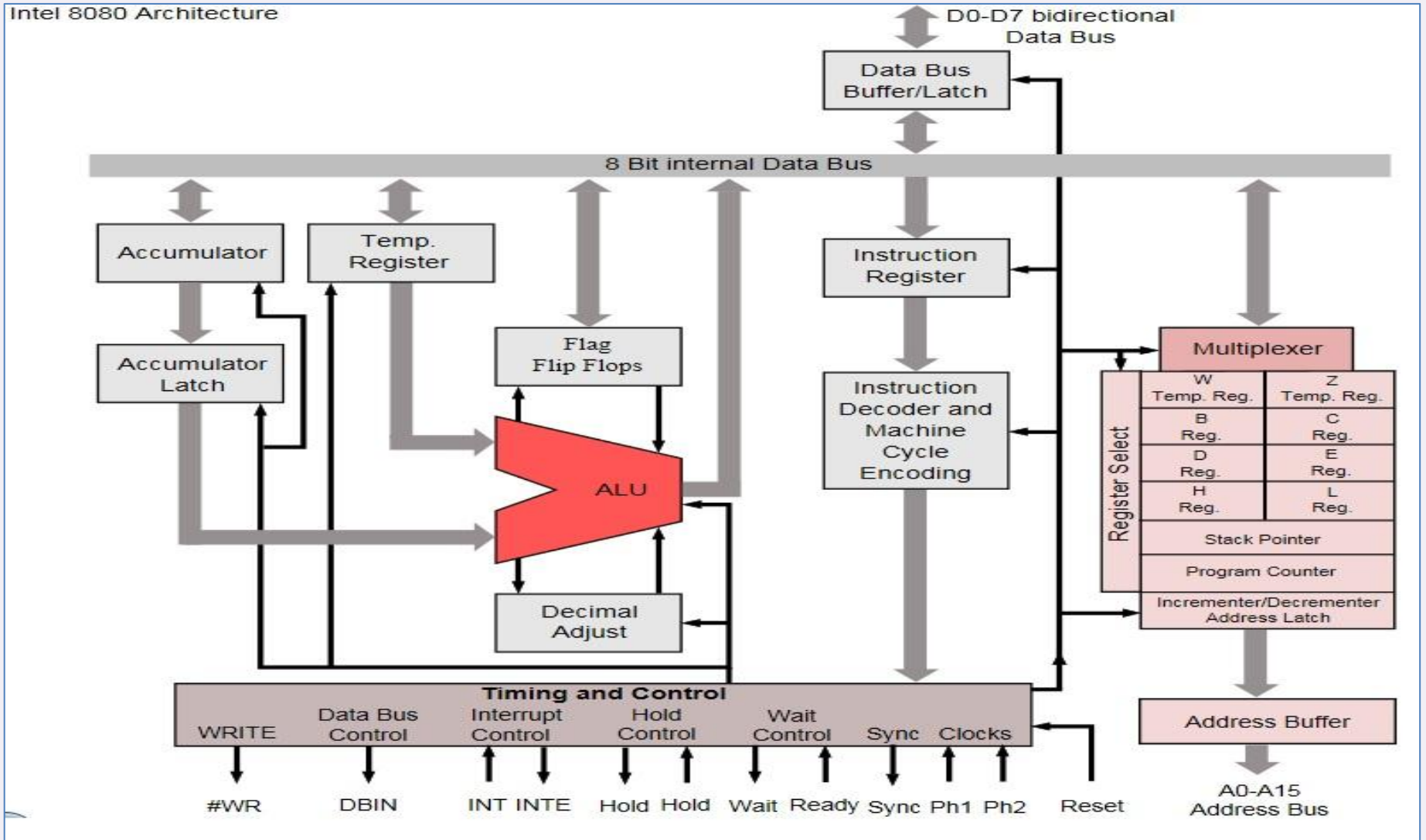
# Organização geral da memória



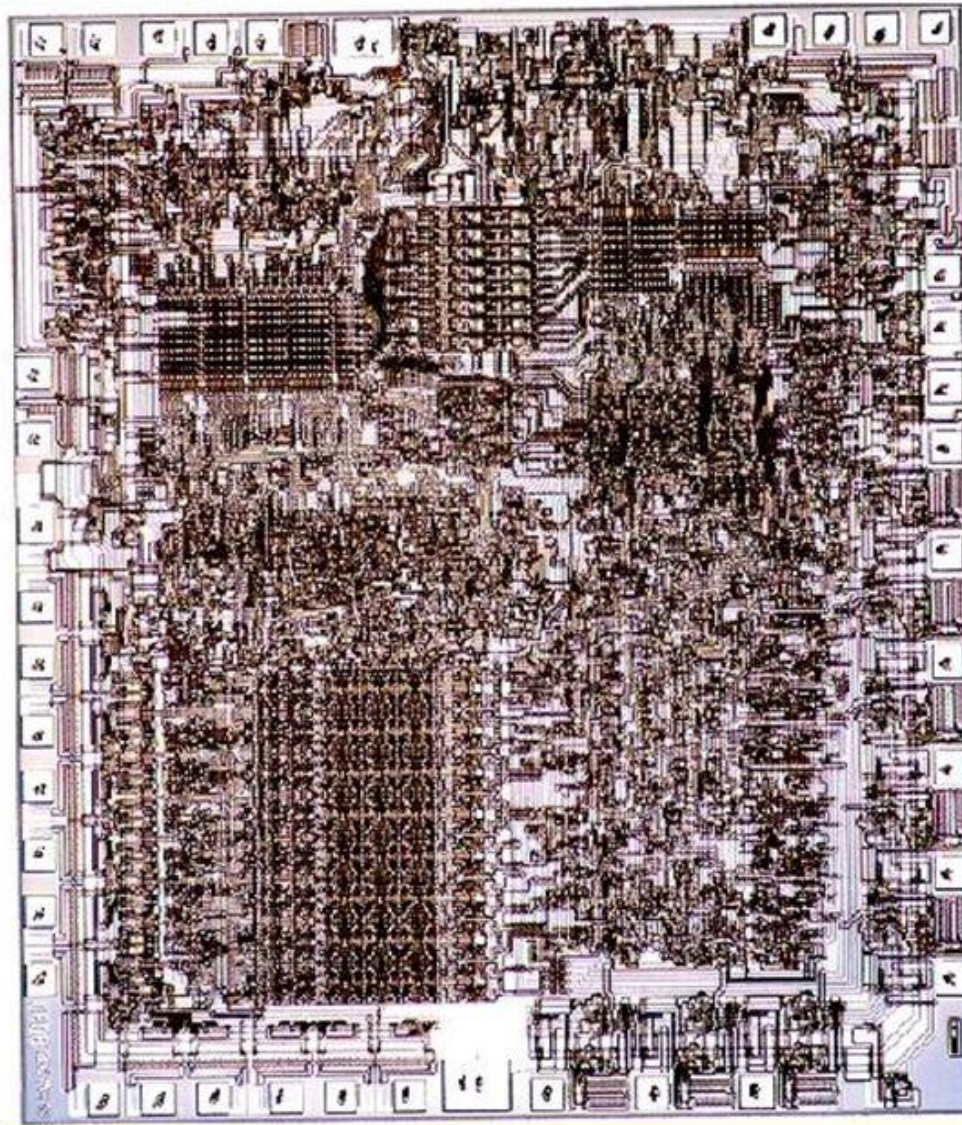
# *Overlays*

- Técnica para permitir que programas muito grandes possam ser executados revezando, de forma não automática, suas partes em uma região comum de memória.

# Arquitetura do Intel 8080







Microprocessador Intel 8080 com 6000 transistores (1974)



# Memória expandida

- EMS (*expanded memory system*) - técnica de *bank switching*, chaveamento de bancos de memória através de um hardware especial com registradores de mapeamento que, para uma área fixa do espaço de endereçamento, mapeia bancos de **1MB** dentre diversos bancos de memória.

# Outras áreas de armazenamento

- ***RAM disk*** – usa a *extended memory* para simular discos rápidos em memória *read/write*
- ***cache*** – usa a *extended memory* para colocar nela um buffer de memória, para acelerar o acesso a endereços de memória muito referenciados

# Sistema de Arquivos

- MS-DOS 1.0 seguiu o padrão dos arquivos CP/M (diretório linear, E/S controlada através de *file control blocks*)
- MS-DOS 2.0 adicionou sistema de arquivos hierárquico parecido com o do UNIX
- Introduziu **descritores** de arquivos e chamadas de sistema baseados no UNIX
- As chamadas de sistema do CP/M ficaram obsoletas e caíram em desuso

# Semelhanças entre UNIX e DOS

- Os sistemas UNIX e o MS-DOS têm em comum, entre outros, os seguintes pontos:
  - diretório hierárquico
  - diretório corrente colocado em evidência
  - uso de caminhos (paths) absolutos e relativos
  - uso de diretórios . e .. (caminho implícito corrente e um nível superior)

# Diferenças entre UNIX e DOS

- O UNIX e o MS-DOS diferiam nos seguintes pontos (entre outros):
  - comprimento dos nomes (DOS 8+3, UNIX 14 ou 255)
  - separador dos nomes nos caminhos (/ , \)
  - maiúsculas e minúsculas (sinônimas, diferentes)
  - proteção, donos de arquivos, grupos de usuários (não, sim)
  - *flags* (somente no DOS):  
**R** [read only], **A**[archive], **S**[system], **H**[hidden]

# Entrada e saída

- Nomes especiais de arquivos designam dispositivos especiais em DOS
- O usuário pode instalar seus drivers de entrada/saída no momento do boot (para instalar um driver, é preciso inserir um comando com seu *nome* [da forma *nome.sys*] no arquivo *config.sys*)

# Chamadas de supervisor

- Muitas se assemelham às correspondentes do sistema UNIX, outras são totalmente distintas em DOS

# Gerenciamento de Processos

- Criar processo filho e executar
- Carregar overlay na memória
- Terminar a execução
- Terminar a execução e permanecer na memória



# Gerenciamento de memória

- Alocar bloco de memória
- Liberar bloco de memória
- Mudar o tamanho de um bloco alocado
- Escolher uma estratégia de alocação
- Permitir ou não o uso da memória superior

# Gerenciamento de arquivos

- Criar arquivo
- Abrir arquivo para entrada ou saída
- Fechar arquivo
- Ler dados de um arquivo para um buffer
- Gravar dados de um buffer para um arquivo
- Posicionar ponteiro do arquivo
- Mudar o nome de um arquivo
- Remover um arquivo
- Criar um diretório
- Remover um diretório
- Alterar o diretório corrente

# Entrada e saída

- Ler e gravar fisicamente dados e parâmetros entre um equipamento de transferência, armazenamento ou comunicação de dados e a memória do computador.

# ASPECTOS DE IMPLEMENTAÇÃO

# Processos

- Simples, porque o DOS não tem multiprogramação
- Para carregar e executar um programa em DOS:
  - Aloca memória suficiente
  - Constrói um PSP nos primeiros 256 bytes alocados
  - Transfere o programa executável para a área alocada
  - Caso seja um programa .exe, relocar todos os endereços
  - Iniciar programa
    - .exe – endereço obtido no cabeçalho
    - .com – endereço constante:  $256_{10}$

- O PSP tem todas as informações necessárias à execução do programa
- Uma lista ligada de PSPs informa o sistema sobre todos os processos
- *Scheduling* da CPU é simples: o programa é executado permanentemente exceto quando um TSR é acionado (assincronamente)
- Durante a execução de TSRs, o sistema perde temporariamente o controle do programa que estava sendo executado

# Memória

- Os blocos de memória no DOS também são organizados como **lista ligada**
- Cada bloco ou **arena** tem um cabeçalho de 16 bytes, contendo ponteiro para o PSP do processo, o tamanho da arena em parágrafos, e o nome do programa executável que a arena contém
- Quando um bloco é liberado, arenas vazias adjacentes não se fundem obrigatoriamente.
- Isso ocorre apenas quando a lista ligada é percorrida desde o início

- Drivers especiais de memória permitem acessar de várias maneiras a memória estendida – *cache*, *RAM disk*, simular memória expandida.
- Não há *swapping* nem paginação.
- O controle do uso da memória é de total responsabilidade do usuário.



# Sistema de Arquivos

- *Layout* do disco: setor de boot primário, diversas partições
- Cada partição tem: FAT (*file allocation table*); uma duplicata da FAT; diretório raiz; diversos blocos de dados
- FAT contém informação sobre toda a área de disco disponível no dispositivo
- Um elemento da FAT para cada bloco no disco
- O tamanho do bloco (1 a 8 setores) é indicado no setor de boot

- *Layout* conceitual da FAT:
  - Os dois primeiros elementos da FAT codificam o tamanho do disco
  - Cada bloco tem um elemento correspondente na FAT
  - Uma lista de blocos ligada via FAT compõe um arquivo
  - Blocos livres também são assim marcados na FAT

- diretório – para cada arquivo tem-se um descritor contendo os seguintes dados:
  - nome
  - extensão
  - *archive/directory/volume label/system/hidden/read-only*
  - hora
  - data
  - primeiro bloco
  - tamanho do arquivo

# Entrada/Saída

- É efetuada através de *drivers* dos dispositivos
- Os *drivers* são ligados entre si, e cada um tem um nome e um código
- Programa do usuário solicita ao sistema uma operação de *read* ou *write*
- DOS responde com as seguintes atividades:
  - Usa um descritor para localizar o arquivo na *system file table*
  - Constrói a mensagem de solicitação correspondente
  - Procura a rotina de interrupção correspondente e a ativa
  - Ativa então o programa de entrada/saída propriamente dito
  - Reporta ao usuário o estado de sucesso ou fracasso da solicitação