

COS 2000

Compatible Operating System



[Présentation](#)

[Comment l'installer](#)

[Mode d'emploi](#)

[Faire un programme pour COS](#)

[Liste des APIs](#)

[Détail des APIs](#)

[En cas de problème](#)

<mailto:cos2003@free.fr>

Présentation

COS2000, par définition, est système d'exploitation. Celui-ci prend la direction des opérations à partir du moment où le PC est mis sous tension (Après le BIOS). Il gère tous les périphériques rattachés au PC et offre aux programmeurs les moyens de développer des applications compatibles en fournissant des APIs (Application Programming Interface). COS2000 est basé sur un concept particulier qui est d'offrir aux programmeurs un maximum de fonctions intégrées pour faciliter le travail des programmeurs et réduire la taille des programmes.

Comment l'installer ?

Pour installer COS2000 :

Sous dos/windows 9x

- Insérez une disquette 1.44 Mo vierge ou inutile dans votre lecteur.

- Lancez le programme SETUP.COM situé dans le dossier de COS2000.
- Si celui-ci ne détecte pas d'erreur, COS2000 est installé !

Sous windows NT/Xp

- Insérez une disquette 1.44 Mo vierge ou inutile dans votre lecteur.
- Lancez le programme SETUP.EXE situé dans le dossier de COS2000.
- Si celui-ci ne détecte pas d'erreur, COS2000 est installé !

Sous Linux

- Insérez une disquette 1.44 Mo vierge ou inutile dans votre lecteur.
- Lancez le programme SETUP.SH situé dans le dossier de COS2000.
- Si celui-ci ne détecte pas d'erreur, COS2000 est installé !

Pour lancer COS2000 :

- Insérez la disquette où COS2000 est installé.
- Veillez que dans le BIOS vous puissiez démarrer à partir de A:.
- Redémarrer votre ordinateur et vous serez sur COS2000.

Mode d'emploi

L'interpréteur de commande COS est le premier logiciel qui est lancé au démarrage. A partir de celui-ci vous pouvez exécuter quelques commandes ou logiciels.

En plus des logiciels, l'interpréteur de commandes peut exécuter 6 commandes :

QUIT

Quitte l'interpréteur.

VERS

Donne la version de COS2000.

CLEAR

Efface l'écran.

REBOOT

Redémarre le PC.

CMDS

Donne la liste des commandes disponibles.

MODE [mode]

Permet de changer de mode vidéo. [mode] doit être un entier compris entre 1 et 9. les modes au delà de 4 sont des modes graphiques à texte émulé. Il est déconseillé de les utiliser car il est parfois impossible de revenir aux modes texte.

DISK

Permet de lire un support disquette 1.44 Mo au format FAT12.

CD

Change le dossier actuel vers celui spécifié.

DIR

Permet de lister le contenu du dossier actuel.

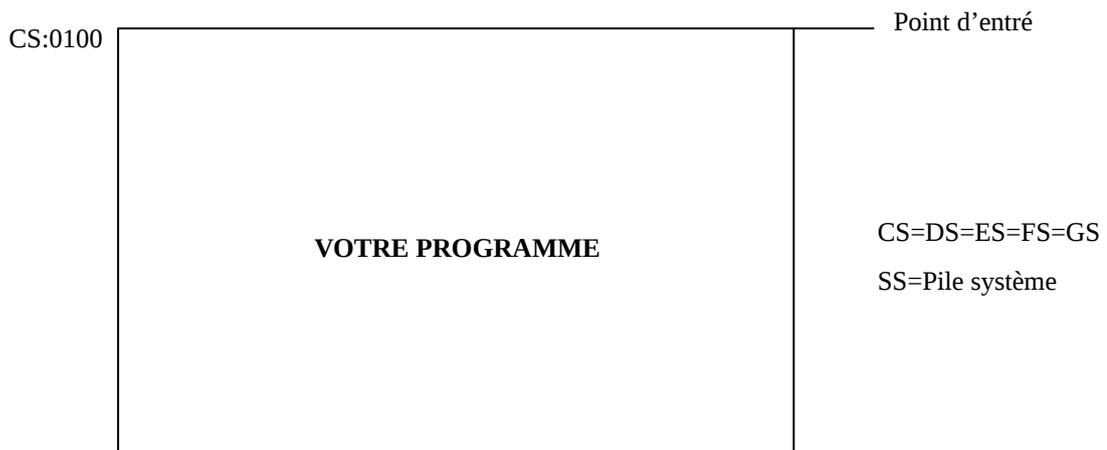
MEM

Permet de lister le contenu du dossier actuel.

Faire un programme pour COS

Toute contribution à COS 2000 en terme de programme est la bienvenue, un répertoire « contribs » contiendra les programmes des différents contributeurs. Aucune modification a ceux-ci ne sera faire sans l'accord explicite de l'auteur. Pour une contribution écrivez moi a l'adresse <mailto:cos2003@free.fr>.

Pour l'instant il n'y a aucun formatage particulier du code à respecter pour faire un programme pour COS 2000. Il faut pour l'instant seulement un point d'entrée en 0100h comme un .COM de dos, 64 Ko sont donc disponible au programme, la pile utilisé est celle du système d'exploitation, c'est donc le seul segment qui ne sera pas initialisé comme les autres.



CS:FFFF

Pour clore le programme il suffit de faire un retour far.
Exemple avec un Hello Word.

Avec tasm

```
.model tiny                ;model tiny (.com)
.486                       ;Pour processeur 80486
Smart                      ;Optimisations
.code                      ;Segment de code

org 0100h                  ;Point d'entré en 0100h

start:
    mov  ah,0Dh
    mov  si,offset msg
    int  47h                ;Afficher le texte (Showstring0)
    xor  ax,ax
    int  16h                ;Attendre l'appuie sur une touche
    db  0CBH ;retour far

msg db 'Hello World',0

end start
```

Avec nasm

```
[bits 16]                  ;16 bits
[org 0x0100]               ;Point d'entré en 0100h
section .text              ;Segment de code

start:
    mov  ah,0x0D
    mov  si,msg
    int  0x47              ;Afficher le texte (Showstring0)
    xor  ax,ax
    int  0x16              ;Attendre l'appuie sur une touche
    retf                   ;retour far

msg db 'Hello World',0
```

Comme vous pouvez le constater l'appel des APIs de Cos se réalise par le biais d'interruptions logiciels dont voici la liste.

Liste des APIs

disque.sys : Gestionnaires FAT12 et Disquette

Interruption 48h (Disquette uniquement)

00h	readsector
01h	writesector
02h	verifysector
03h	initdrive
04h	loadfile
05h	compressrle
06h	decompressrle
07h	findfirstfile
08h	findnextfile
09h	getfreespace
0Ah	searchfile
0Bh	getname
0Ch	getserial
0Dh	changedir
0Eh	readcluster
0Fh	writecluster
10h	getdir
11h	projfile
11h	execfile

port.sys : Gestionnaires port parallèle

Interruption 0Dh (LPT1) ou 0Fh (LPT2)

00h	getlptin
01h	getlptout
02h	getlptinout
03h	setlptin
04h	setlptout
05h	setlptinout
06h	getlpt
07h	getfirstlpt
08h	setemettor
09h	setreceptor
0Ah	settimeout
0Bh	gettimeout
0Ch	receivelpt
0Dh	sendlpt
0Eh	receivelptblock
0Fh	sendlptblock
10h	receivecommand
11h	sendcommand

souris.sys : Gestionnaires souris

Interruption 74h (PS/2)

00h	cmdmouse
01h	cmdmouse2
02h	detectmouse
03h	getmouse
04h	getmousescreen
05h	configmouse

8259a.sys : Gestionnaires contrôleur d'interruption programmable

Interruption 50h (Maître et esclave)

00h	enableirq
01h	disableirq

02h	readmaskirq
03h	readirr
04h	readisr
05h	installhandler
06h	replacehandler
07h	getint
08h	setint
09h	seteoi

video.sys : Gestionnaires de la carte vidéo

Interruption 47h (VGA)

00h	setvideomode
01h	getvideomode
02h	cleartext
03h	changefont
04h	loadfont
05h	space
06h	line
07h	showchar
08h	showint
09h	showsinged
0Ah	showhex
0Bh	showbin
0Ch	showstring
0Dh	showstring0
0Eh	showcharat
0Fh	showintat
10h	showsingedat
11h	showhexat
12h	showbinat
13h	showstringat
14h	showstring0at
15h	setcolor
16h	getcolor
17h	scrolldown
18h	getxy
19h	setxy
1Ah	savescreen
1Bh	restorescreen
1Ch	page2to1
1Dh	page1to2
1Eh	xchgPages
1Fh	savepage1
20h	changelineattr
21h	waitretrace
22h	getinfos
23h	loadbmp
24h	showbmp
25h	viewbmp
26h	savedac
27h	restoredac
28h	savestate
29h	restorestate
2Ah	enablescroll
2Bh	disablescroll
2Ch	showdate
2Dh	showtime
2Eh	showname
2Fh	showattr
30h	showsize
31h	getchar

mcb.sys : Gestionnaires de mémoire vive

Interruption 49h (MB)

00h	mbinit
01h	mbfree
02h	mbcreate
03h	mbresident
04h	mbget
05h	mbfind
06h	mbchown
07h	mballoc
08h	mbclean
09h	mbfindsb

Les possibilités de COS2000 sont aujourd'hui très limitées car il est en cours de développement.

Détail des APIs

- Interruption 48h

Readsector		
Lit le secteur CX et le met en ES:DI. Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=0 CX ES DI
	Données en sortie	Flag Carry

Writesector		
Ecrit le secteur CX avec les données pointés par DS:SI. Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=1 CX DS SI
	Données en sortie	Flag Carry

Verifysector		
Vérifie si le secteur CX n'est pas défectueux. Met le flag carry à 1 si erreur et flag equal à 0 si secteur défectueux.	Paramètres d'entrée	AH=2 CX
	Données en sortie	Flag Carry Flag Equal

Initdrive		
	Paramètres	AH=3

Fonction initialisant le pilote et le matériel afin d'utiliser ultérieurement les fonctions de disque.sys. Met le flag carry à 1 si erreur.	d'entrée	
	Données en sortie	Flag Carry

Loadfile

Charge le fichier dont le nom est pointé par DS:SI en mémoire dans ES:DI et renvoie le nombre d'octets lu en ECX. Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=4 DS SI ES DI
	Données en sortie	ECX Flag Carry

Compressrle

Comprime le contenu de la mémoire pointé par DS:SI (selon une méthode RLE) et dont la taille est spécifié par CX. Le résultat sera mis en ES:DI ainsi que la nouvelle taille mémoire (octets) en BP.	Paramètres d'entrée	AH=5 DS SI ES DI CX
	Données en sortie	

Decompressrle

Décompresse le contenu de la mémoire pointé par DS:SI (selon une méthode RLE) et dont la taille est spécifié par CX. Le résultat sera mis en ES:DI ainsi que la nouvelle taille mémoire (octets) en BP.	Paramètres d'entrée	AH=6 DS SI ES DI CX
	Données en sortie	

Findfirstfile

<p>Renvoie en ES:DI la première entrée du répertoire courant (format BRUT). Met le flag carry à 1 si erreur. Cette fonction prépare aussi l'usage de la fonction findnextfile.</p> <p>Format d'une entrée de répertoire :</p> <pre> Entries Struc FileName db 8 dup (0) FileExt db 3 dup (0) FileAttr db 0 FileReserved db 0 FileTimeCreaMs db 0 ;(*10 ms) FileTimeCrea dw 0 FileDateCrea dw 0 FileDateAcc dw 0 FileNotused dw 0 FileTime dw 0 FileDate dw 0 FileGroup dw 0 </pre>	Paramètres d'entrée	AH=7 ES DI
	Données en sortie	Flag Carry

FileSize dd 0 Entries Ends		
------------------------------------	--	--

Findnextfile

Renvoie en ES:DI l'entrée suivante du répertoire courant (format BRUT). Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=8 ES DI
	Données en sortie	Flag Carry

Getfreespace

Renvoie en EDX l'espace disque libre du volume en octets. Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=9
	Données en sortie	EDX Flag Carry

Searchfile

Renvois dans ES:DI l'entrée de répertoire du fichier pointé par DS:SI. Met le flag equal a 0 si pas existant. Met le flag carry à un si erreur.	Paramètres d'entrée	AH=10 DS SI ES DI
	Données en sortie	Flag Equal Flag Carry

Getname

Renvois dans ES:DI le nom du support courant.	Paramètres d'entrée	AH=11 ES DI
	Données en sortie	

Getserial

Renvois le numéro de série du support courant en EDX.	Paramètres d'entrée	AH=11
	Données en sortie	EDX

Changedir

Change le répertoire courant a celui dont le nom est pointé par DS:SI. Met le flag carry à un si erreur.	Paramètres d'entrée	AH=13 DS SI
	Données en sortie	Flag Carry

Readcluster		
Lit le cluster (groupe) CX et le met en ES:DI. Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=14 ES DI
	Données en sortie	Flag Carry

Writecluster		
Ecrit le cluster (groupe) CX avec les données pointés par DS:SI. Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=15 ES DI
	Données en sortie	Flag Carry

Getdir		
Renvoie en ES:DI sous forme de chaîne a zéro terminal le nom du répertoire courant.	Paramètres d'entrée	AH=16 ES DI
	Données en sortie	

Projfile		
Charge le fichier dont le nom est pointé par DS:SI dans un bloc mémoire. Renvoie en ECX le nombre d'octets lus et en ES l'adresse du bloc de mémoire. Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=17 DS SI
	Données en sortie	ECX ES Flag Carry

Execfile		
Exécute le fichier dont le nom est pointé par DS:SI. Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=18 DS SI
	Données en sortie	Flag Carry

- Interruption 47h

Setvideomode		
Fixe le mode vidéo courant a dont le numéro est AL. Met le flag carry à 1 si erreur. MODES : 0 -> 40x25x16 couleurs en texte 1 -> 80x25x16 couleurs en texte 2 -> 80x50x16 couleurs en texte 3 -> 100x50x16 couleurs en texte 4 -> 100x60x16 couleurs en texte	Paramètres d'entrée	AH=0 AL
	Données en sortie	Flag Carry

<p>5 -> 320x200x256 couleurs en graphique 6 -> 320x400x256 couleurs en graphique 7 -> 320x480x256 couleurs en graphique 8 -> 360x480x256 couleurs en graphique 9 -> 400x600x256 couleurs en graphique</p> <p>Les modes sont généralement utilisable avec une carte VGA 256ko, mais la plupart surexploient le contrôleur vidéo donc ne fonctionne pas toujours. L'utilisation des fonctions caractères est disponible en mode graphique par l'usage de polices émulés mais beaucoup de bugs sont encore présent.</p>		
--	--	--

Getvideomode		
Retourne le mode vidéo courant a dans AL.	Paramètres d'entrée	AH=1
	Données en sortie	AL

Getvideomode		
Efface l'écran en mode graphique ou texte.	Paramètres d'entrée	AH=2
	Données en sortie	

SetFont		
Active la police de numéro CL parmi les 8 disponibles.	Paramètres d'entrée	AH=3 CL
	Données en sortie	

Getfont (PAS ENCORE IMPLEMENTE)		
Récupère en CL le N° de police actif.	Paramètres d'entrée	AH CL
	Données en sortie	

Loadfont		
Charge une police pointée par DS:SI dans la carte vidéo sous le n° de police BL. La taille en hauteur de la police (en pixel) doit être renseigné dans CL Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=4 DS SI CL
	Données en sortie	Flag Carry

Showspace		
		AH=5

Affiche un espace à l'écran après le curseur.	Paramètres d'entrée	
	Données en sortie	

Showline

Affiche un retour a la ligne à l'écran après le curseur.	Paramètres d'entrée	AH=6
	Données en sortie	

Showchar

Affiche un caractère dont le code ASCII est contenu dans DL à l'écran après le curseur.	Paramètres d'entrée	AH=7 DL
	Données en sortie	

Showint

Affiche le nombre entier contenu dans EDX à l'écran après le curseur.	Paramètres d'entrée	AH=8 EDX
	Données en sortie	

Showsigned

Affiche le nombre entier signé contenu dans EDX à l'écran après le curseur.	Paramètres d'entrée	AH=9 EDX
	Données en sortie	

Showhex

Affiche le nombre hexadécimal contenu dans EDX et de taille CX bits à l'écran après le curseur.	Paramètres d'entrée	AH=10 EDX CX
	Données en sortie	

ShowBin

Affiche le nombre binaire contenu dans EDX et de taille CX bits à l'écran après le curseur.	Paramètres d'entrée	AH=11 EDX CX
	Données en sortie	

ShowString

<p>Affiche la chaîne de caractère (type fixe) pointée par DS:SI à l'écran après le curseur.</p> <p>Chaîne de type fixe :</p> <p>Chaîne db 24, 'c un chaîne de type fixe'</p>	Paramètres d'entrée	AH=12 DS SI
	Données en sortie	

ShowString0

<p>Affiche la chaîne de caractère (type zéro terminal) pointée par DS:SI à l'écran après le curseur.</p> <p>Chaîne de type zéro terminal:</p> <p>Chaîne db 'c un chaîne de type zéro terminal',0</p>	Paramètres d'entrée	AH=13 DS SI
	Données en sortie	

Showchartat

<p>Réalise la même fonction que la fonction showchar en spécifiant les coordonnées BL (y), BH (x) ou tout devra être affiché.</p>	Paramètres d'entrée	AH=14 DL BL BH
	Données en sortie	

Showintat

<p>Réalise la même fonction que la fonction showint en spécifiant les coordonnées BL (y), BH (x) ou tout devra être affiché.</p>	Paramètres d'entrée	AH=15 EDX BL BH
	Données en sortie	

Showsignedat

<p>Réalise la même fonction que la fonction showsigned en spécifiant les coordonnées BL (y), BH (x) ou tout devra être affiché.</p>	Paramètres d'entrée	AH=16 EDX BL BH
	Données en sortie	

Showhexat

<p>Réalise la même fonction que la fonction showhex en spécifiant les coordonnées BL (y), BH (x) ou tout devra être affiché.</p>	Paramètres d'entrée	AH=17 EDX CX BL BH
	Données en sortie	

Showbinat

	Paramètres	AH=18 EDX
--	------------	--------------

Réalise la même fonction que la fonction showbin en spécifiant les coordonnées BL (y), BH (x) ou tout devra être affiché.	d'entrée	CX BL BH
	Données en sortie	

Showstringat

Réalise la même fonction que la fonction showstring en spécifiant les coordonnées BL (y), BH (x) ou tout devra être affiché.	Paramètres d'entrée	AH=19 DS SI BL BH
	Données en sortie	

Showstring0at

Réalise la même fonction que la fonction showstring0 en spécifiant les coordonnées BL (y), BH (x) ou tout devra être affiché.	Paramètres d'entrée	AH=20 DS SI BL BH
	Données en sortie	

Setcolor

Change la couleur courante (attributs) pour les opérations textes a celle spécifié dans CL	Paramètres d'entrée	AH=21 CL
	Données en sortie	

Setcolor

Récupère dans CL la couleur courante (attributs) pour les opérations textes.	Paramètres d'entrée	AH=22
	Données en sortie	CL

Setstyle (PAS ENCORE IMPLEMENTE)

Change le style (transparent ou non) courant pour les opérations graphique a celui spécifié dans CL	Paramètres d'entrée	AH CL
	Données en sortie	

Getstyle (PAS ENCORE IMPLEMENTE)

Récupère dans CL le style courant (transparent ou non) pour les opérations graphique.	Paramètres d'entrée	AH
	Données en sortie	CL

ScrollDown

Défile l'écran texte ou graphique de CX caractères vers le haut.	Paramètres d'entrée	AH=23 CX
	Données en sortie	

Getxy

Renvoie en BH les coordonnées x du curseur texte et en BL les coordonnées y du curseur texte.	Paramètres d'entrée	AH=24
	Données en sortie	BH BL

Setxy

Fixe les coordonnées x du curseur texte a BH et les coordonnées y du curseur texte a L.	Paramètres d'entrée	AH=25 BH BL
	Données en sortie	

SaveScreen

Sauvegarde le contenu de l'écran dans un bloc mémoire appelé /vgascreen lié a l'application appelante.	Paramètres d'entrée	AH=26
	Données en sortie	

RestoreScreen

Restaure le contenu de l'écran précédemment sauvegardé dans un bloc mémoire.	Paramètres d'entrée	AH=27
	Données en sortie	

Page1to2

Copie le contenu de la page vidéo n°1 dans la page vidéo n°2. Ne fonctionne qu'en mode texte.	Paramètres d'entrée	AH=28
	Données en sortie	

Page2to1

Copie le contenu de la page vidéo n°2 dans la page vidéo n°1. Ne fonctionne qu'en mode texte.	Paramètres d'entrée	AH=29
	Données en sortie	

Xchgpage

Echange le contenu de la page vidéo n°2 dans la page vidéo n°1. Ne fonctionne qu'en mode texte.	Paramètres d'entrée	AH=30
	Données en sortie	

Savepage1

Sauvegarde le contenu de l'écran dans un bloc mémoire appelé /vgapage1 lié a l'application appelante.	Paramètres d'entrée	AH=31
	Données en sortie	

Changelineattr (VA ETRE SUPPRIMER)

Modifie la couleur de la ligne N°DI a l'écran a celle contenue dans AL.	Paramètres d'entrée	AH=32 AL DI
	Données en sortie	

Waitretrace

Synchronisation avec la retrace verticale.	Paramètres d'entrée	AH=33
	Données en sortie	

Getvgainfos

Renvoie un bloc de donnée en ES:DI contenant l'état de la carte graphique. lines db 0 columns db 0 x db 0 y db 0 xy dw 0 colors db 7 mode db 0FFh pagesize dw 0 pages db 0 font db 0 graphic db 0 xg dw 0 yg dw 0 style dw 0 nbpage db 0 pagesshowed db 0 plane db 0 xyg dw 0 linesize dw 0 adress dw 0 base dw 0	Paramètres d'entrée	AH=34 ES SI
	Données en sortie	

scrolling	db 1	
-----------	------	--

Loadbmppalet		
Charge la palette (DAC) du BMP pointée par DS:SI.	Paramètres d'entrée	AH=35 DS SI
	Données en sortie	

Showbmp		
Affiche le BMP pointée par DS:SI en X:BX, Y:CX.	Paramètres d'entrée	AH=36 DS SI BX CX
	Données en sortie	

Viewbmp		
Affiche le BMP pointée par DS:SI en X:BX, Y:CX avec la préparation de la palette.	Paramètres d'entrée	AH=3 DS SI BX CX
	Données en sortie	

Savedac		
Sauvegarde le contenu de la palette (DAC) dans un bloc mémoire appelé /vgadac lié a l'application appelante.	Paramètres d'entrée	AH=38
	Données en sortie	

Savedac		
Restaure le contenu de la palette (DAC) précédemment sauvegardé dans un bloc mémoire.	Paramètres d'entrée	AH=39
	Données en sortie	

Savestate		
Sauvegarde l'état complet de la carte graphique dans un bloc mémoire appelé /vga lié a l'application appelante. FONCTIONNE PEUT ETRE EN MODE GRAPHIQUE.	Paramètres d'entrée	AH=40
	Données en sortie	

Restorestate		
		AH=41

Restaure l'état complet de la carte graphique précédemment sauvegardé dans un bloc mémoire. FONCTIONNE PEUT ETRE EN MODE GRAPHIQUE.	Paramètres d'entrée	
	Données en sortie	

EnableScroll

Active le défilement automatique de l'écran lors de dépassements.	Paramètres d'entrée	AH=42
	Données en sortie	

DisableScroll

Desactive le défilement automatique de l'écran lors de dépassements.	Paramètres d'entrée	AH=43
	Données en sortie	

Showdate

Affiche la date contenue dans DX à l'écran après le curseur.	Paramètres d'entrée	AH=44 DX
	Données en sortie	

Showtime

Affiche l'heure contenue dans à l'écran après le curseur.	Paramètres d'entrée	AH=45 DX
	Données en sortie	

Showname

Affiche le nom de fichier pointé par DS:SI à l'écran après le curseur.	Paramètres d'entrée	AH=46 DS SI
	Données en sortie	

Showattr

Affiche les attributs fichiers contenus dans DL à l'écran après le curseur.	Paramètres d'entrée	AH=47 DL
	Données en sortie	

Showsize

		AH=48
--	--	-------

Affiche le la taille en octets (et multiples) contenue dans EDX à l'écran après le curseur.	Paramètres d'entrée	EDX
	Données en sortie	

Getchar

Renvoie le caractère situé sous le curseur dans DL.	Paramètres d'entrée	AH=49
	Données en sortie	DL

Setxyg (PAS ENCORE IMPLEMENTE)

Change les coordonnées du curseur graphique a X:BX,Y:CX.	Paramètres d'entrée	AH BX CX
	Données en sortie	

Getxyg (PAS ENCORE IMPLEMENTE NI ECRIT)

Récupère les coordonnées du curseur graphique a X:BX,Y:CX.	Paramètres d'entrée	AH
	Données en sortie	BX CX

Showpixel (PAS ENCORE IMPLEMENTE)

Affiche un pixel de couleur AL en X:BX,Y:CX.	Paramètres d'entrée	AH BX CX AL
	Données en sortie	

Getpixel (PAS ENCORE IMPLEMENTE)

Récupère la couleur du pixel en X:BX,Y:CX dans AL.	Paramètres d'entrée	AH BX CX
	Données en sortie	AL

- Interruption 49h

Mbinit

Initialise les blocs de mémoire pour une utilisation futur des fonction MBs (inutile car le système le réalise au boot). Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=0
	Données en sortie	Flag Carry

Mbfree

Libère le bloc de mémoire GS ainsi que tout les sous blocs de mémoire lié (fils). Un bloc de mémoire considéré résident ou un sous bloc résident ne sera pas libéré. Met le flag carry à 1 si erreur.	Paramètres d'entrée	AH=1 GS
	Données en sortie	Flag Carry

Mbcreate

Crée un bloc de CX caractères (octets) et de nom DS :SI. Retourne en GS le bloc de mémoire alloué et met le flag carry à 1 en cas d'erreur.	Paramètres d'entrée	AH=2 DS SI
	Données en sortie	GS Flag Carry

Mbresident

Met le bloc GS en situation de bloc mémoire résident (non libérable).	Paramètres d'entrée	AH=3 GS
	Données en sortie	

Mbget

Renvoie en GS l'adresse du bloc mémoire situé en CX ème position. Met le flag carry à 1 si introuvable.	Paramètres d'entrée	AH=4
	Données en sortie	GS Flag Carry

Mbfind

Renvoie en GS le bloc de mémoire dont le nom correspond a la chaîne de caractère situé en DS:SI. Met le flag carry à 1 si introuvable.	Paramètres d'entrée	AH=5 DS SI
	Données en sortie	GS Flag Carry

Mbchown

Change le propriétaire (père) du bloc de mémoire GS a celui précisé par DX.	Paramètres d'entrée	AH=6 GS DX
	Données en sortie	Flag Carry

Mballoc

Alloue un bloc de CX caractères (octets) pour le processus (programme) qui le demande. Retourne en GS le bloc de mémoire alloué et met le flag carry à 1 en cas d'erreur.	Paramètres d'entrée	AH=7 CX
	Données en sortie	GS Flag Carry

Mbclean		
Nettoie un peu la mémoire pour fonctionner des blocs de mémoire libre contiguë. Généralement inutile car géré par le système.	Paramètres d'entrée	AH=8
	Données en sortie	Flag Carry

Mbfindsb		
Renvoie en GS le sous bloc de mémoire dont le nom correspond a la chaîne de caractère situé en DS:SI et dont le propriétaire est DX. Met le flag carry à 1 si introuvable.	Paramètres d'entrée	AH=9 DS SI DX
	Données en sortie	Flag Carry

- Interruption 74h

Cmdmouse		
Envoie une commande AL à la souris via contrôleur clavier	Paramètres d'entrée	AH=0 AL
	Données en sortie	

Cmdmouse2		
Envoie une commande type 2 AL à la souris via contrôleur clavier	Paramètres d'entrée	AH=1 AL
	Données en sortie	

Detectmouse		
Déecte et initialise une souris de type PS/2. Met le flag carry à 1 si introuvable.	Paramètres d'entrée	AH=2 AL
	Données en sortie	Flag Carry

Getmouse		
Envoie en BX,CX les coordonnées virtuelles de la souris (respectivement X et Y) ainsi qu'en DL l'état des boutons.	Paramètres d'entrée	AH=3
	Données en sortie	BX CX DL

Getmousescreen		
Envoie en BX,CX les coordonnées écran de la souris (respectivement X et Y)	Paramètres d'entrée	AH=4

ainsi qu'en DL l'état des boutons.	Données en sortie	BX CX DL
------------------------------------	-------------------	----------------

Configmouse		
Configure la vitesse de la souris dans CL et dans AH, AL les sphères X et Y.	Paramètres d'entrée	AH=5 AH AL CL
	Données en sortie	

A suivre pour les autres ressources.... (et avec exemples !)

En cas de problèmes

Si des bugs surviennent ou si COS2000 ne veut pas s'installer, veuillez s'il vous plaît m'envoyer un E-Mail à :

<mailto:cos2003@free.fr>

COS2000 n'exploite pas les disques durs, il est donc improbable qu'il altère vos données !