

**Programmation utilisant les interruptions  
du système d'exploitation :  
le cas de MS-DOS**

Patrick Cégielski

Janvier 2019

*Pour Irène et Marie*

## **Legal Notice**

Copyright © 2019 Patrick Cégielski  
Université Paris XII – IUT de Sénart-Fontainebleau  
Route forestière Hurtaut  
F-77300 Fontainebleau  
[cegielski@u-pec.fr](mailto:cegielski@u-pec.fr)



# Table des matières

<b>Préface</b>	<b>ix</b>
0.1 Bibliographie . . . . .	x
<b>1 Les grandes fonctions d'un système d'exploitation</b>	<b>1</b>
1.1 Étude générale . . . . .	2
1.1.1 Les deux tâches d'un système d'exploitation . . . . .	2
1.1.2 Principe des systèmes d'exploitation . . . . .	3
1.2 Cas de MS-DOS . . . . .	5
1.2.1 Modèle en trois couches . . . . .	5
1.3 Historique . . . . .	6
1.4 Bibliographie . . . . .	9
<b>I Le système d'exploitation comme machine virtuelle</b>	<b>11</b>
<b>2 Programmer avec le DOS</b>	<b>13</b>
2.1 Les entrées-sorties standard . . . . .	14
2.1.1 Fonction 02h d'affichage d'un caractère . . . . .	14
2.1.2 Fonction 01h de saisie d'un caractère avec écho . . . . .	15
2.1.3 Fonction 08h de saisie d'un caractère sans écho . . . . .	17
2.1.4 Saisie d'une chaîne de caractères . . . . .	17
2.2 L'interruption 33h de manipulation de la souris . . . . .	18
2.2.1 Fonction 00h d'initialisation de la souris . . . . .	18
2.2.2 Fonctions 01h et 02h d'affichage et de transparence du pointeur . . . . .	18
2.2.3 Fonction 03h de récupération de la position du pointeur . . . . .	19
2.2.4 Fonction 04h de déplacement du pointeur . . . . .	20
2.3 Fonction 05h d'impression . . . . .	21
2.4 Le système . . . . .	22
2.4.1 Fonction 2Ah d'obtention de la date . . . . .	22
2.4.2 Récupération de l'heure . . . . .	22
2.5 Liste des interruptions du DOS . . . . .	23
2.5.1 Numéros d'interruption . . . . .	23
2.5.2 Les fonctions de l'interruption 21h . . . . .	24
2.6 Bibliographie . . . . .	25

<b>II</b>	<b>Le système d'exploitation comme gestionnaire des ressources</b>	<b>27</b>
<b>3</b>	<b>Les processus de MS-DOS</b>	<b>29</b>
3.1	Philosophie des processus de MS-DOS . . . . .	30
3.1.1	Processus actif . . . . .	30
3.1.2	Programme résident . . . . .	30
3.2	Les programmes exécutables de MS-DOS . . . . .	31
3.2.1	Deux types d'exécutables sous MS-DOS . . . . .	31
3.2.2	Les programmes <code>.com</code> . . . . .	32
3.2.3	Création d'un exécutable <code>.com</code> à la main . . . . .	37
3.2.4	Les programmes <code>.exe</code> . . . . .	38
3.3	Chargement des programmes . . . . .	43
3.3.1	Fonction du processeur de commande . . . . .	43
3.3.2	Préfixe d'un programme transitoire (PSP) . . . . .	44
3.3.3	Chargement d'un programme exécutable <code>.com</code> . . . . .	47
3.3.4	Chargement d'un programme <code>.exe</code> . . . . .	48
<b>4</b>	<b>Gestion de la mémoire</b>	<b>49</b>
4.1	Étude générale de la gestion de la mémoire . . . . .	50
4.1.1	Modèles de programmation . . . . .	50
4.1.2	Modèles d'organisation de la mémoire . . . . .	51
4.2	Mise en place dans MS-DOS . . . . .	54
4.2.1	Organisation de la mémoire adressable . . . . .	54
4.2.2	Partage de la mémoire . . . . .	56
4.2.3	Fonctions <code>48h</code> , <code>49h</code> et <code>4Ah</code> d'allocation dynamique de la mémoire . . . . .	59
<b>5</b>	<b>Les systèmes de fichiers</b>	<b>61</b>
5.1	Étude générale des systèmes de fichiers . . . . .	62
5.1.1	Intérêt des fichiers . . . . .	62
5.1.2	Interface utilisateur . . . . .	63
5.1.3	Mise en oeuvre des systèmes de fichiers . . . . .	69
5.2	Cas de MS-DOS . . . . .	75
5.2.1	Organisation physique de l'espace disque . . . . .	75
5.2.2	Lecture du contenu d'un disque avec <code>debug</code> . . . . .	76
5.2.3	Lecture et écriture des secteurs : fonctions <code>25h</code> et <code>26h</code> . . . . .	77
5.2.4	Organisation logique d'un disque . . . . .	81
5.2.5	Écriture et lecture des fichiers : fonctions <code>3Ch</code> , <code>3Dh</code> , <code>40h</code> et <code>3Eh</code> . . . . .	82
5.2.6	Accès direct : fonction <code>42h</code> . . . . .	89
5.2.7	Gestion des sous-répertoires : fonctions <code>39h</code> et <code>3Ah</code> . . . . .	91
5.2.8	Fonctions <code>41h</code> et <code>56h</code> de destruction et de déplacement des fichiers . . . . .	93
5.3	Bibliographie . . . . .	95

# Table des figures

4.1 Organisation de la mémoire de l'IBM PC . . . . .	54
--	----





# Préface

Nous avons vu la notion de système d'exploitation en tant qu'utilisateur. Nous avons même rencontré plusieurs systèmes d'exploitation, que ce soit en ligne de commande comme MS-DOS et Unix, ou avec interface graphique comme X-Window System, Windows et MacOS.

MS-DOS est un système d'exploitation simple, certains disent même rudimentaire, puisqu'il s'agit d'un système en ligne de commande, mono-utilisateur et mono-tâche. Mais il permet d'accéder à toutes les ressources du microprocesseur et à toute la mémoire centrale.

On peut distinguer cinq niveaux en ce qui concerne les rapports avec un système d'exploitation :

- le *niveau utilisateur*, dans lequel le problème principal est de charger les logiciels dont on a besoin et de les faire exécuter ; cela sous-tend d'utiliser et de manipuler les fichiers ; on se sert pour cela de l'interpréteur de commandes et de commandes telles que `copy`, `rename...` ;

- le *niveau administrateur*, consistant à paramétrer le système et à le tenir à jour ; c'est évidemment très simple pour MS-DOS (par rapport à Unix, par exemple), il suffit de remplir correctement les deux fichiers `autoexec.bat` et `config.sys` ;

- le *niveau écriture de scripts* pour automatiser certaines séquences répétitives de commandes ; ceci est réalisé en MS-DOS avec des fichiers de traitement par lots (*batch* en anglais) d'extension `.bat` ;

- le *niveau programmation système*, en utilisant les *appels systèmes* ; cette programmation se fait pour MS-DOS en langage d'assemblage en utilisant des *interruptions* nouvelles, par rapport au BIOS, et éventuellement en langage évolué tel que le langage C, toujours en utilisant ces mêmes interruptions ;

- le *niveau conception du système*, réservé aux ingénieurs de Microsoft pour ce qui concerne MS-DOS ; le code de MS-DOS n'est pas diffusé, contrairement à celui du BIOS, mais on peut essayer de le retrouver (ou d'en trouver un qui soit analogue) au vu de la description des interruptions existante et des informations éparses, récupérées de ci de là.

Nous allons nous intéresser ici à la programmation en utilisant les interruptions logicielles du système d'exploitation.

# Chapitre 1

## Les grandes fonctions d'un système d'exploitation

Nous allons rappeler dans ce chapitre ce qu'est un système d'exploitation du point de vue de l'utilisateur et quelles en sont les grandes parties, en illustrant ces propos par MS-DOS.

## 1.1 Étude générale

### 1.1.1 Les deux tâches d'un système d'exploitation

Un système d'exploitation effectue fondamentalement deux tâches sans rapport l'une avec l'autre : il émule une machine virtuelle et il gère les ressources. Précisons chacune de ces tâches.

#### 1.1.1.1 Le système d'exploitation en tant que machine virtuelle

L'*architecture* (ensemble d'instructions, organisation de la mémoire, entrées-sorties, structure du bus...) de la plupart des ordinateurs au niveau du langage machine est lourde à gérer pour le programmeur. Bien peu de programmes seraient développés si chaque programmeur devait connaître le fonctionnement, par exemple, de tel ou tel disque et de toutes les erreurs pouvant apparaître lors de la lecture d'un bloc de celui-ci. On a heureusement trouvé un moyen de libérer les programmeurs de la complexité du matériel. Ce moyen consiste à enrober le matériel d'une couche logicielle gérant l'ensemble du système. Cela revient à présenter au programmeur une interface ou **machine virtuelle** plus facile à comprendre et à programmer.

#### 1.1.1.2 Le système d'exploitation en tant que gestionnaire

Les ordinateurs modernes se composent d'un processeur (microprocesseur), de mémoires, d'horloges, de disques, de moniteurs, d'interfaces réseau, d'imprimantes et autres périphériques. La deuxième fonction du système d'exploitation consiste à contrôler l'allocation du processeur, de l'espace mémoires et des périphériques entre les différents programmes y faisant appel.

Imaginez ce qui se produirait si trois programmes s'exécutant sur un ordinateur essayaient d'imprimer simultanément leurs résultats sur la même imprimante. Les premières lignes imprimées pourraient provenir du programme 1, les suivantes du programme 2, puis du programme 3 et ainsi de suite. Il en résulterait le désordre le plus total. Le système d'exploitation peut éviter ce chaos potentiel en transférant les résultats à imprimer dans un fichier tampon dans un répertoire du disque. Lorsqu'un travail d'impression se termine, le système d'exploitation peut alors imprimer un des fichiers se trouvant dans le répertoire tampon. Simultanément, un autre programme peut continuer à générer des résultats sans se rendre compte qu'il ne les envoie pas (encore) à l'imprimante.

## 1.1.2 Principe des systèmes d'exploitation

Comme pour le BIOS, l'interface entre le système d'exploitation et les programmes est constituée d'un ensemble d'« instructions étendues » fournies par le système d'exploitation, qualifiées d'**appels système**.

Les appels système créent, détruisent et utilisent divers objets logiciels gérés par le système d'exploitation, les *processus* et les *fichiers* étant les plus importants de ces objets.

### 1.1.2.1 Les processus

Un **processus** est un programme en train de s'exécuter. Ce n'est pas seulement un programme (le code), mais ses données et les paramètres indiquant où il en est, lui permettant ainsi de continuer s'il est interrompu (grâce à une pile d'exécution, un compteur ordinal...) : on parle de l'**environnement** du processus.

Les appels système les plus importants pour la gestion des processus sont ceux concernant la *création* et la *fin* du processus. Par exemple le processus *interpréteur de commandes* lit les commandes ; si l'utilisateur demande la compilation d'un programme, l'interpréteur doit créer un nouveau processus qui lance l'exécution du compilateur ; ce dernier processus fera, à la fin de la compilation, à un appel système appel permettant de le terminer correctement.

Il existe d'autres appels système pour les processus, en particulier pour demander davantage de mémoire ou pour restituer la mémoire qui n'est plus utilisée.

### 1.1.2.2 Le système de fichiers

Comme nous l'avons déjà dit, une des tâches fondamentales du système d'exploitation est de masquer les spécificités des disques et des autres périphériques d'entrées-sorties et d'offrir au programmeur un modèle agréable et facile d'emploi lui permettant de gérer les **fichiers** indépendamment du matériel utilisé.

Des appels systèmes permettent de créer un fichier, de le supprimer, de le lire et d'écrire dans celui-ci.

La plupart des systèmes d'exploitation font appel au concept de **répertoire** (en anglais *directory*) pour regrouper les fichiers. Il faut donc également des appels système pour créer et détruire un répertoire lorsque celui-ci n'est pas unique, pour y placer ou en retirer un fichier dans tous les cas.

De nombreux systèmes d'exploitation offrent un niveau d'abstraction qui permet aux utilisateurs d'effectuer des entrées-sorties sans entrer dans aucun détail du matériel. Ce niveau d'abstraction considère chaque périphérique comme un **fichier spécial**. Les fichiers spéciaux permettent donc de traiter les périphériques d'entrées-sorties comme des fichiers. On peut ainsi les lire et y écrire en utilisant les mêmes appels système que pour lire et écrire dans un fichier ordinaire.

Il existe deux types de fichiers spéciaux : les *fichiers spéciaux bloc* et les *fichiers spéciaux caractère*. Les **fichiers spéciaux bloc** servent à modéliser les périphériques, tels que les disques, constitués d'un certain nombre de blocs adressables individuellement. Un programme peut, par exemple, accéder au quatrième bloc d'un périphérique, en ouvrant un fichier spécial bloc et en lisant ce quatrième bloc. Les **fichiers spéciaux caractère** servent à modéliser les périphériques qui gèrent les flots de caractères plutôt que les blocs directement adressables. Les terminaux, les imprimantes et les interfaces réseau sont des exemples typiques de ce dernier type de périphérique.

### 1.1.2.3 Les appels système

Les programmes des utilisateurs communiquent donc avec le système d'exploitation (lui demandent des services) en effectuant des *appels système*. À chaque appel système correspond une routine. Les paramètres de l'appel système sont placés en un endroit prédéfini (par exemple les registres du processeur) et la routine de service est appelée. Lorsque le système d'exploitation prend le contrôle, il vérifie la validité des paramètres et, si c'est le cas, effectue le traitement demandé. Lorsqu'il a terminé, le système d'exploitation place un **code de statut** dans un registre, indiquant que le traitement a réussi ou échoué, puis exécute une instruction de retour pour rendre le contrôle à l'appelant.

### 1.1.2.4 L'interpréteur de commandes

Le système d'exploitation est le code exécutant les appels système. Les programmes système tels que les éditeurs de texte, les compilateurs, les assembleurs, les éditeurs de liens et les interpréteurs de commandes ne font pas partie du système d'exploitation. Cependant, l'interpréteur de commandes est souvent considéré comme en faisant partie.

Conclusion.- Les systèmes d'exploitation se composent de quatre parties principales : la gestion des processus, la gestion de la mémoire, la gestion des fichiers et la gestion des périphériques d'entrées-sorties. Il ne faut pas oublier non plus le chargeur du système d'exploitation et l'interpréteur de commandes.

## 1.2 Cas de MS-DOS

### 1.2.1 Modèle en trois couches

Le système d'exploitation MS-DOS est structuré en trois couches :

1. le BIOS (*Basic Input Output System*);
2. le noyau;
3. l'interpréteur de commandes, *command.com*.

Le BIOS.- Le BIOS contient un chargeur du système d'exploitation ainsi qu'une série de pilotes de périphériques de bas niveau servant à isoler MS-DOS des spécificités du matériel. Le BIOS contient, par exemple, des appels système pour lire et écrire à des adresses absolues de disques, pour lire un caractère sur le clavier et pour en afficher un sur l'écran. Il est fourni par le constructeur de l'ordinateur, non par Microsoft, et se trouve en ROM.

Lien entre BIOS et noyau.- Le fichier `IO.SYS` (appelé `IO.COM` dans la version 1.25 de MS-DOS et `IBMBIO.COM` dans la version PC-DOS vendue par IBM) est un fichier (caché) de MS-DOS, chargé lors du démarrage de la machine (plus exactement de MS-DOS), fournissant une interface avec le BIOS par appels de routines. Cela permet au noyau d'accéder aux services du BIOS en appelant des routines d'`IO.SYS` au lieu de faire des déroutements vers la ROM. L'existence de `IO.SYS` permet d'encore mieux isoler le noyau du matériel car, par exemple, le noyau n'a même pas besoin de savoir à quel vecteur d'interruption correspond un service BIOS donné : cette information est enfouie dans `IO.SYS`.

Lors du cycle de démarrage d'un ordinateur compatible PC, à l'amorçage d'un ordinateur, le premier secteur du disque de démarrage (**MBR** pour *Master Boot Record*) est chargé et exécuté. Si celui-ci est un MBR MS-DOS, il charge en mémoire et exécute les 3 premiers secteurs du programme `IO.SYS`.

Ces trois premiers secteurs de `IO.SYS` exécutent les actions suivantes :

- `IO.SYS` est entièrement chargé en mémoire, quel que soit sa taille, et donc s'il occupe plus de trois secteurs;
- Tous les pilotes de périphérique sont initialisés (console écran, disque dur, port série, notamment). À ce moment, les périphériques par défaut sont disponibles;
- Le programme `MSDOS.SYS` (appelé `MSDOS.COM` dans la version 1.25 de MS-DOS) est chargé et sa routine d'initialisation est exécutée. À ce moment, l'accès au système de fichiers est disponible;
- L'interpréteur de commande par défaut `COMMAND.COM` (ou un autre *shell* si spécifié par la commande « `shell=` ») est chargé et exécuté.

Noyau.- Le noyau contient la partie du système d'exploitation indépendante de la machine. Il se trouve dans un autre fichier (caché également), appelé `MSDOS.SYS` (`MSDOS.COM` dans la version 1.25 de MS, `IBMDOS.COM` pour PC-DOS). Il est chargé de la gestion des processus, de la mémoire et du système de fichiers ainsi que de l'interprétation de tous les appels système.

Interpréteur de commandes.- L'interpréteur de commandes `COMMAND.COM` ne fait pas réellement partie du système d'exploitation et peut être remplacé par un autre interpréteur de commandes par l'utilisateur, s'il le désire. L'interpréteur de commandes comprend deux parties : une *partie résidente*, dont le code est toujours situé en mémoire centrale (d'où son nom), et des *parties non résidentes*, qui ne sont situées en mémoire que lorsqu'on en a besoin.

### 1.3 Historique

MS-DOS (abréviation de *Microsoft Disk Operating System*) est le système d'exploitation développé par Microsoft pour l'IBM PC d'abord, puis pour les compatibles PC. Il a été le système le plus utilisé sur compatible PC de juillet 1981 au début des années 1990, avant d'être progressivement remplacé par des systèmes d'exploitation plus évolués, notamment *Windows*.

#### Échec du « contrat du siècle » avec IBM

Gary KILDALL (1942–1994) crée le système d'exploitation *Control Program/Monitor* (CP/M) en 1974 pour les ordinateurs 8 bits, porté plus tard sur des ordinateurs 16 bits (CP/M-86). CP/M est une révolution : dans le monde naissant des micro-ordinateurs, c'est le tout premier système d'exploitation n'étant pas destiné à une marque ou un modèle particuliers ; compatible avec des ordinateurs de microprocesseurs différents, il fonctionne sur Intel 80xx, sur Zilog Z80 et sur Motorola (68xx et 69xx). En 1976, KILDALL et sa femme Dorothy créent leur propre société, *Digital Research Inc.*

En 1980, *Digital Research* rate ce qui est considéré comme le « contrat du siècle » avec IBM. IBM sort en 1980 l'IBM PC, son premier ordinateur personnel. Convaincu que ce type d'ordinateur sera un échec commercial, IBM décide d'acheter un système d'exploitation plutôt que de le développer en interne. CP/M étant le système d'exploitation alors le plus répandu pour les micro-ordinateurs, IBM approche *Digital Research* pour rencontrer Gary KILDALL.

Les pourparlers entre KILDALL et les représentants d'IBM n'aboutissant pas, cela offre à Bill GATES (de *Microsoft*) l'occasion de remporter ce marché.

Bill GATES reçoit d'abord les représentants d'IBM. Pour le système d'exploitation, il les aiguille vers *Digital Research*, ne pensant qu'à fournir ce qui est la spécialité de sa compagnie, les langages de programmation et notamment le BASIC.

KILDALL, passionné d'aviation, aurait fait attendre les représentants d'IBM plusieurs heures (pilote alors son avion personnel).

Ce serait Dorothy, la femme de KILDALL, qui aurait reçu les représentants d'IBM. Ceux-ci, avant de commencer la discussion, auraient exigé que DRI signe un accord de non-divulagation interdisant de révéler quoi que ce soit de leur entretien. Dorothy, avocate, aurait refusé de signer le document.

Gary KILDALL et Tom ROLANDER ont effectivement pris l'avion privé de KILDALL pour livrer de la documentation CP/M à un client du côté de la baie de San Francisco. Ils sont cependant revenus à temps pour participer à la réunion avec IBM. Selon ROLANDER, la réunion s'engagea mal. Tout d'abord, IBM veut faire signer un accord de non-divulagation : toute information communiquée par IBM à *Digital Research* doit être tenue secrète, au point que *Digital Research* aurait à nier l'existence même de la réunion. Par contre, les informations communiquées par DRI à IBM tomberaient dans le domaine public. De plus, IBM veut acheter CP/M à *Digital Research* pour un prix fixe, quel que soit le nombre d'exemplaires vendus, alors que *Digital Research* préfère de loin toucher des redevances pour chaque exemplaire vendu. Finalement IBM ne veut en aucun cas garder le nom de CP/M pour son système d'exploitation, insistant pour l'appeler PC-DOS (même s'il est entièrement écrit par *Digital Research*). Tous ces éléments déplaisent fortement à KILDALL.

Constatant le mécontentement des envoyés d'IBM envers Gary KILDALL et *Digital Research*, et voyant l'occasion de pénétrer le marché des systèmes d'exploitation, Bill GATES se ravise et propose à IBM de lui fournir un système d'exploitation. Il va, pour sa part, le chercher (sans raconter toute l'histoire) dans une société ayant adapté CP/M sous le nom de **86-DOS**.

Après la sortie de l'IBM PC, KILDALL examine MS-DOS et considère que c'est un plagiat de CP/M. Il menace IBM d'un procès. IBM (qui avait probablement été dans l'ignorance de toutes ces manœuvres) trouve l'accord suivant : l'acheteur de l'IBM PC pourra choisir d'installer soit

MS-DOS (sous le nom de PC-DOS), soit CP/M. Mais PC-DOS étant vendu 40 \$ et CP/M 240 \$, le choix de l'utilisateur est vite fait.

### Tim Paterson

Bill GATES s'est donc mis à la recherche d'un système d'exploitation qu'il pourrait effectivement fournir à IBM pour le lancement de l'IBM PC. L'objectif de *Microsoft* était que MS-DOS soit aussi compatible avec CP/M que possible. GATES rachète alors QDOS (*Quick and Dirty OS*, c'est-à-dire « Système d'exploitation vite fait mal fait », aussi appelé 86-DOS) à la petite société SCP (*Seattle Computer Products*). QDOS est en fait un clone à bas prix de CP/M écrit par Tim PATERSON. Bill GATES demande alors à Tim PATERSON d'adapter QDOS à l'IBM PC et de le renommer MS-DOS.

Après l'obtention de son diplôme à l'université de Washington avec les félicitations du jury en juin 1978, Tim PATERSON est embauché en tant qu'ingénieur de conception pour *Seattle Computer Products*.

Un mois plus tard, *Intel* sort son microprocesseur 8086 et PATERSON travaille à la conception d'une carte S-100 8086, commercialisée en novembre 1979. Cette carte étant dépourvue de tout système d'exploitation, le seul logiciel disponible pour cette dernière étant une version autonome du BASIC de Microsoft, les ventes sont plutôt faibles. Pour combler ce manque, PATERSON commence à travailler sur QDOS en avril 1980. Il contient approximativement 4 000 lignes de code en langage d'assemblage 8086 et est totalement compatible avec l'API du populaire système d'exploitation concurrent CP/M. La première version 0.10 sort en juillet 1980.

En décembre de la même année, *Microsoft* achète une licence de QDOS. PATERSON quitte SCP en avril 1981 et part travailler pour Microsoft de mai 1981 à avril 1982. Après un bref retour chez SCP, PATERSON fonde sa propre société, *Falcon Technology*, qui sera rachetée par *Microsoft* en 1986. PATERSON retravaille pour Microsoft de 1986 à 1988, puis de 1990 à 1998. Durant cette troisième période chez *Microsoft*, il travaille sur Visual Basic.

Après avoir de nouveau quitté *Microsoft*, Tim PATERSON fonde une nouvelle société spécialisée en développement logiciel, *Paterson Technology*, dont le site web :

<http://www.patersonstech.com/>

contient, sous l'onglet « *Origins of Dos* », des articles intéressants et les manuels concernant 86-DOS.

### Hiérarchie de répertoires

Une fois la licence achetée, 86-DOS est modifié pour l'IBM PC par Bob O'REAR, de *Microsoft* avec l'assistance de Tim PATERSON. O'REAR parvient à faire tourner 86-DOS sur le prototype du PC en février 1981. Puis 86-DOS doit être adapté pour passer des disquettes de 8 pouces à celles de 5" 1/4 et s'intégrer avec le BIOS, que *Microsoft* aide IBM à écrire. La version 1.0 de PC-DOS est mise en vente avec le premier IBM PC, en août 1981.

Vers fin 1981, PATERSON travaille à une amélioration, appelée PC DOS 1.1. Elle permet aux données d'être écrites sur les deux côtés de la disquette, en doublant ainsi la capacité. Elle est terminée en mars 1982. La version 1.1, supportant donc les lecteurs de disquette 5" 1/4 double face (d'une capacité de 320 KiO), est mise en vente en mai 1982.

Pendant ce temps, une équipe de programmeurs de *Microsoft*, comprenant son co-fondateur Paul ALLEN, Mark ZBIKOWSKI et Aaron REYNOLDS, travaille à une révision majeure pour ajouter une hiérarchie de répertoires au lieu d'un répertoire unique, ainsi que pour prendre en compte les disques durs de l'IBM PC-XT. Le programme DOS 2.0 est plus de deux fois plus gros que DOS 1.0, avec 20 000 lignes de code en langage d'assemblage contre 4 000 lignes dans la première version, utilisant 28 KiO de mémoire vive au lieu de 12 KiO. Il équipe en mars 1983 l'IBM PC-XT sous le nom de PC-DOS 2.0, et est vendu aux fabricants de compatibles sous le nom MS-DOS 2.0.



### Les versions

- MS-DOS 1.00 (juillet 1981) sort avec le tout premier IBM PC. Ne supporte que les disquettes 5" 1/4 simple face de 160 kiO. Il n'y a qu'un seul répertoire, ne supportant que 64 fichiers au maximum par disquette.
- MS-DOS 1.25 (1982) : équivalent de PC-DOS 1.10 mais en version OEM. C'est la première version vendue par Microsoft (MS-DOS) à d'autres constructeurs de compatibles PC. Outre quelques commandes supplémentaires, cette version ajoute le support des disquettes 5" 1/4 double face de 320 kiO.
- MS-DOS 2.00 (mars 1983) : première version à supporter un disque dur (FAT12) et une hiérarchie de répertoires. Supporte aussi les disquettes 5" 1/4 de 180 kiO (simple face) et de 360 kiO (double face).
- MS-DOS 2.10 (mars 1984).
- MS-DOS 3.0 (août 1984) : supporte les disquettes de 1,2 MiO (1 200 kiO) et les disques durs de plus de 15 MiO grâce à la FAT16.
- MS-DOS 3.1 (novembre 1984) : première version à gérer les réseaux.
- MS-DOS 3.2 (janvier 1986) : supporte les disquettes 3,5 pouces de 720 kiO.
- MS-DOS 3.3 (août 1987) : supporte les disquettes 3,5 pouces de 1,44 MiO (1 440 kiO) et les partitions étendues.
- MS-DOS 3.31 (novembre 1987) : supporte les partitions de plus de 32 MiO grâce à un adressage des secteurs logiques passé de 16 à 32 bits; la taille maximum du disque dur passe à 504 MiO.
- MS-DOS 4.00 (juin 1988) : il occupe 110 kiO de RAM, offre une interface graphique (le *shell*) et la taille d'une partition passe à 2 GiO.
- MS-DOS 4.01 : version déboguée de la précédente.
- MS-DOS 5.00 (1991) : version gérant la mémoire haute, et permettant de faire des menus de démarrage; supporte jusqu'à 8 disques durs et les disquettes de 2,88 MiO (2 880 kiO).
- MS-DOS 6.00 (1993) : version intégrant un anti-virus, un outil de vérification de système de fichiers, *ScanDisk*, et un logiciel de compression, *DoubleSpace*.
- MS-DOS 6.20 (novembre 1993) : version déboguée de la précédente; apparition de l'utilitaire *Memmaker*.
- MS-DOS 6.21 (février 1994) : suppression de *DoubleSpace* à la suite d'une plainte pour violation de brevet de la part de *Stac Electronics*, éditeur d'un logiciel concurrent, *Stacker*.
- MS-DOS 6.22 (juin 1994) : *DoubleSpace* est remplacé par *DriveSpace*, logiciel compatible sans le problème de brevet. C'est la dernière version autonome de MS-DOS.  
Microsoft a décidé d'acheter une licence à *Vertisoft* pour la technologie de compression *DriveSpace*.
- MS-DOS 7 (1995) : à partir de cette version MS-DOS est intégré à *Windows 95*. Il permet de démarrer *Windows* et ses propres applications. Elle possède des outils de base pour la gestion des disques durs.
- MS-DOS 7.1 (1996) : cette version est intégrée à *Windows 95b* (OSR 2.0), *Windows 95c* (OSR 2.5), *Windows 98* et *Windows 98 (Deuxième Édition)*. Elle apporte le support du système FAT32 et une meilleure gestion de la mémoire. HIMEM.SYS est automatiquement chargé ainsi que l'outil SCANREG permettant le dépannage du registre.
- MS-DOS 8 (2000). Cette dernière version de MS-DOS intègre et démarre *Windows ME*. C'est une évolution de la version 7.1. Elle intègre un gestionnaire de mémoire étendue, le

mode pur a été désactivé et on peut configurer les spécifications des pays et du clavier à partir de la base du registre. À partir de *Windows XP*, MS-DOS sert essentiellement pour la création d'une disquette de démarrage.

## 1.4 Bibliographie

- [Pat-83] PATERSON, Tim, *An Inside Look at MS-DOS - The design decisions behind the popular operating system*, **BYTE**, June 1983, pp. 230–252.
- [Pat-94] PATERSON, Tim, *From the Mailbox : The Origins of DOS*, **MicroDesign Resources**, Microprocessor Report, Vol. 8, n° 13, October 3, 1994, 3 p.
- [IBM-87] IBM Corporation and Microsoft, Inc., **DOS 3.30 : User's Guide**, IBM Corporation, 1987, Part number 80X0933.
- [IBM-87a] IBM Corporation and Microsoft, Inc., **DOS 3.30 : Reference (Abridged)**, IBM Corporation, 1987, Part number 94X9575.
- [IBM-88] IBM Corporation. **Getting Started with Disk Operating System Version 4.00**, IBM Corporation, 1988, Part number 15F1370.
- [IBM-88a] IBM Corporation, **Using Disk Operating System Version 4.00**, IBM Corporation, 1988, Part number 15F1371.
- [IBM-91] IBM Corporation, **IBM Disk Operating System Version 5.0. User Guide and Reference**, IBM Corporation, 1991, Part number 07G4584.
- [Wei-95] WEIXEL, Suzanne, **IBM PC DOS and Microsoft Windows User's Guide**, Que Corporation, Indianapolis, 2nd ed., 1995, ISBN 0-7897-0276-2.
- [Ave-95] AVERETT, Margaret, **PC DOS 7 User's Guide**, IBM Corporation, 1995, Part number 83G9260 (S83G-9260-00).
- [IBM-95] IBM Corporation, **PC DOS 7 Technical Update**, IBM Redbooks, 1995, ISBN 0-7384-0677-5.

L'ensemble des images des disquettes de distribution de MS-DOS et des manuels en pdf se trouve sur le site :

<https://winworldpc.com/product/pc-dos/3x>