

Chap. III : Le système d'exploitation

Laurent Poinsot

UMR 7030 - Université Paris 13 - Institut Galilée

Cours “Architecture et Système”

Le **système d'exploitation** (ou **O.S.** de l'anglais “ **Operating System** ”) d'un ordinateur est le programme qui permet d'accéder aux ressources matérielles de cet ordinateur. Ces ressources matérielles sont essentiellement les organes d'entrées/sorties : clavier, écran, liaisons réseau, imprimante, disque dur, etc.

Le **système d'exploitation** (ou **O.S.** de l'anglais “ **Operating System** ”) d'un ordinateur est le programme qui permet d'accéder aux ressources matérielles de cet ordinateur. Ces ressources matérielles sont essentiellement les organes d'entrées/sorties : **clavier, écran, liaisons réseau, imprimante, disque dur, etc.**

Le **système d'exploitation** (ou **O.S.** de l'anglais “ **Operating System** ”) d'un ordinateur est le programme qui permet d'accéder aux ressources matérielles de cet ordinateur. Ces ressources matérielles sont essentiellement les organes d'entrées/sorties : clavier, écran, liaisons réseau, imprimante, disque dur, etc.

Les périphériques d'entrées/sorties varient d'un modèle d'ordinateur à l'autre. Même au sein de la famille des " compatibles PC ", on trouve difficilement deux modèles dotés d'exactly les mêmes périphériques (cartes d'extension par exemple). De ce fait, les instructions à exécuter pour piloter tel périphérique (par exemple pour afficher un rectangle rouge à l'écran) diffèrent d'un ordinateur à l'autre.

Le rôle principal du système d'exploitation est d'isoler les programmes des détails du matériel. Un programme désirant afficher un rectangle ne va pas envoyer des instructions à la carte graphique de l'ordinateur, mais plutôt demander au système d'exploitation de le faire. C'est le système d'exploitation qui doit connaître les détails du matériel (dans ce cas le type de carte graphique et les instructions qu'elle comprend). Cette répartition des rôles simplifie grandement l'écriture des programmes applicatifs.

Les périphériques d'entrées/sorties varient d'un modèle d'ordinateur à l'autre. Même au sein de la famille des “ compatibles PC ”, on trouve difficilement deux modèles dotés d'exactly les mêmes périphériques (cartes d'extension par exemple). De ce fait, les instructions à exécuter pour piloter tel périphérique (par exemple pour afficher un rectangle rouge à l'écran) diffèrent d'un ordinateur à l'autre.

Le rôle principal du système d'exploitation est d'isoler les programmes des détails du matériel. Un programme désirant afficher un rectangle ne va pas envoyer des instructions à la carte graphique de l'ordinateur, mais plutôt demander au système d'exploitation de le faire. C'est le système d'exploitation qui doit connaître les détails du matériel (dans ce cas le type de carte graphique et les instructions qu'elle comprend). Cette répartition des rôles simplifie grandement l'écriture des programmes applicatifs.

Les périphériques d'entrées/sorties varient d'un modèle d'ordinateur à l'autre. Même au sein de la famille des “ compatibles PC ”, on trouve difficilement deux modèles dotés d'exactly les mêmes périphériques (cartes d'extension par exemple). De ce fait, les instructions à exécuter pour piloter tel périphérique (par exemple pour afficher un rectangle rouge à l'écran) diffèrent d'un ordinateur à l'autre.

Le rôle principal du système d'exploitation est d'isoler les programmes des détails du matériel. Un programme désirant afficher un rectangle ne va pas envoyer des instructions à la carte graphique de l'ordinateur, mais plutôt demander au système d'exploitation de le faire. C'est le système d'exploitation qui doit connaître les détails du matériel (dans ce cas le type de carte graphique et les instructions qu'elle comprend). Cette répartition des rôles simplifie grandement l'écriture des programmes applicatifs.

Les périphériques d'entrées/sorties varient d'un modèle d'ordinateur à l'autre. Même au sein de la famille des “ compatibles PC ”, on trouve difficilement deux modèles dotés d'exactly les mêmes périphériques (cartes d'extension par exemple). De ce fait, les instructions à exécuter pour piloter tel périphérique (par exemple pour afficher un rectangle rouge à l'écran) diffèrent d'un ordinateur à l'autre.

Le rôle principal du système d'exploitation est d'isoler les programmes des détails du matériel. Un programme désirant afficher un rectangle ne va pas envoyer des instructions à la carte graphique de l'ordinateur, mais plutôt demander au système d'exploitation de le faire. C'est le système d'exploitation qui doit connaître les détails du matériel (dans ce cas le type de carte graphique et les instructions qu'elle comprend). Cette répartition des rôles simplifie grandement l'écriture des programmes applicatifs.

Les périphériques d'entrées/sorties varient d'un modèle d'ordinateur à l'autre. Même au sein de la famille des “ compatibles PC ”, on trouve difficilement deux modèles dotés d'exactly les mêmes périphériques (cartes d'extension par exemple). De ce fait, les instructions à exécuter pour piloter tel périphérique (par exemple pour afficher un rectangle rouge à l'écran) diffèrent d'un ordinateur à l'autre.

Le rôle principal du système d'exploitation est d'isoler les programmes des détails du matériel. Un programme désirent afficher un rectangle ne va pas envoyer des instructions à la carte graphique de l'ordinateur, mais plutôt demander au système d'exploitation de le faire. C'est le système d'exploitation qui doit connaître les détails du matériel (dans ce cas le type de carte graphique et les instructions qu'elle comprend). Cette répartition des rôles simplifie grandement l'écriture des programmes applicatifs.

Les périphériques d'entrées/sorties varient d'un modèle d'ordinateur à l'autre. Même au sein de la famille des " compatibles PC ", on trouve difficilement deux modèles dotés d'exactly les mêmes périphériques (cartes d'extension par exemple). De ce fait, les instructions à exécuter pour piloter tel périphérique (par exemple pour afficher un rectangle rouge à l'écran) diffèrent d'un ordinateur à l'autre.

Le rôle principal du système d'exploitation est d'isoler les programmes des détails du matériel. Un programme désirent afficher un rectangle ne va pas envoyer des instructions à la carte graphique de l'ordinateur, mais plutôt demander au système d'exploitation de le faire. C'est le système d'exploitation qui doit connaître les détails du matériel (dans ce cas le type de carte graphique et les instructions qu'elle comprend). Cette répartition des rôles simplifie grandement l'écriture des programmes applicatifs.

Les périphériques d'entrées/sorties varient d'un modèle d'ordinateur à l'autre. Même au sein de la famille des “ compatibles PC ”, on trouve difficilement deux modèles dotés d'exactly les mêmes périphériques (cartes d'extension par exemple). De ce fait, les instructions à exécuter pour piloter tel périphérique (par exemple pour afficher un rectangle rouge à l'écran) diffèrent d'un ordinateur à l'autre.

Le rôle principal du système d'exploitation est d'isoler les programmes des détails du matériel. Un programme désirent afficher un rectangle ne va pas envoyer des instructions à la carte graphique de l'ordinateur, mais plutôt demander au système d'exploitation de le faire. C'est le système d'exploitation qui doit connaître les détails du matériel (dans ce cas le type de carte graphique et les instructions qu'elle comprend). Cette répartition des rôles simplifie grandement l'écriture des programmes applicatifs.

Le système d'exploitation est donc un programme complexe, lié à la configuration matérielle de la machine. À chaque type de matériel correspond un type de système d'exploitation (MS-DOS, Windows, Mac-OS, Linux, Unix). Tout système d'exploitation est divisé en plusieurs couches. La couche basse est responsable de la gestion du matériel, et change par exemple suivant le type de périphérique installé. Les couches plus hautes sont chargées de fonctions plus évoluées (gestion des fichiers sur disque par exemple), plus ou moins indépendantes du matériel.

Le système d'exploitation est donc un programme complexe, lié à la configuration matérielle de la machine. À chaque type de matériel correspond un type de système d'exploitation (MS-DOS, Windows, Mac-OS, Linux, Unix). Tout système d'exploitation est divisé en plusieurs couches. La couche basse est responsable de la gestion du matériel, et change par exemple suivant le type de périphérique installé. Les couches plus hautes sont chargées de fonctions plus évoluées (gestion des fichiers sur disque par exemple), plus ou moins indépendantes du matériel.

Le système d'exploitation est donc un programme complexe, lié à la configuration matérielle de la machine. À chaque type de matériel correspond un type de système d'exploitation (MS-DOS, Windows, Mac-OS, Linux, Unix). Tout système d'exploitation est divisé en plusieurs couches. La couche basse est responsable de la gestion du matériel, et change par exemple suivant le type de périphérique installé. Les couches plus hautes sont chargées de fonctions plus évoluées (gestion des fichiers sur disque par exemple), plus ou moins indépendantes du matériel.

Le système d'exploitation est donc un programme complexe, lié à la configuration matérielle de la machine. À chaque type de matériel correspond un type de système d'exploitation (MS-DOS, Windows, Mac-OS, Linux, Unix). Tout système d'exploitation est divisé en plusieurs couches. La couche basse est responsable de la gestion du matériel, et change par exemple suivant le type de périphérique installé. Les couches plus hautes sont chargées de fonctions plus évoluées (gestion des fichiers sur disque par exemple), plus ou moins indépendantes du matériel.

Le système d'exploitation est donc un programme complexe, lié à la configuration matérielle de la machine. À chaque type de matériel correspond un type de système d'exploitation (MS-DOS, Windows, Mac-OS, Linux, Unix). Tout système d'exploitation est divisé en plusieurs couches. La couche basse est responsable de la gestion du matériel, et change par exemple suivant le type de périphérique installé. Les couches plus hautes sont chargées de fonctions plus évoluées (gestion des fichiers sur disque par exemple), plus ou moins indépendantes du matériel.

Le système d'exploitation est donc un programme complexe, lié à la configuration matérielle de la machine. À chaque type de matériel correspond un type de système d'exploitation (MS-DOS, Windows, Mac-OS, Linux, Unix). Tout système d'exploitation est divisé en plusieurs couches. La couche basse est responsable de la gestion du matériel, et change par exemple suivant le type de périphérique installé. Les couches plus hautes sont chargées de fonctions plus évoluées (gestion des fichiers sur disque par exemple), plus ou moins indépendantes du matériel.

Le système d'exploitation est donc un programme complexe, lié à la configuration matérielle de la machine. À chaque type de matériel correspond un type de système d'exploitation (MS-DOS, Windows, Mac-OS, Linux, Unix). Tout système d'exploitation est divisé en plusieurs couches. La couche basse est responsable de la gestion du matériel, et change par exemple suivant le type de périphérique installé. Les couches plus hautes sont chargées de fonctions plus évoluées (gestion des fichiers sur disque par exemple), plus ou moins indépendantes du matériel.

Un système d'exploitation offre différents services essentiels au bon fonctionnement d'un ordinateur :

- Il gère l'ensemble des échanges entre le microprocesseur, la mémoire centrale et les divers périphériques (écran, clavier, souris, etc.) ;
- Il met à disposition de l'utilisateur un certain nombre de commandes et il gère leur exécution (formatage des disquettes, copie de fichiers, création de répertoires, ...)
- Il détecte et récupère (lorsque c'est possible) les erreurs et en informe l'utilisateur.

Un système d'exploitation offre différents services essentiels au bon fonctionnement d'un ordinateur :

- Il gère l'ensemble des échanges entre le microprocesseur, la mémoire centrale et les divers périphériques (écran, clavier, souris, etc.) ;
- Il met à disposition de l'utilisateur un certain nombre de commandes et il gère leur exécution (formatage des disquettes, copie de fichiers, création de répertoires, ...) ;
- Il détecte et récupère (lorsque c'est possible) les erreurs et en informe l'utilisateur.

Un système d'exploitation offre différents services essentiels au bon fonctionnement d'un ordinateur :

- Il gère l'ensemble des échanges entre le microprocesseur, la mémoire centrale et les divers périphériques (écran, clavier, souris, etc.) ;
- Il met à disposition de l'utilisateur un certain nombre de commandes et il gère leur exécution (formatage des disquettes, copie de fichiers, création de répertoires, ...)
- Il détecte et récupère (lorsque c'est possible) les erreurs et en informe l'utilisateur.

Un système d'exploitation offre différents services essentiels au bon fonctionnement d'un ordinateur :

- Il gère l'ensemble des échanges entre le microprocesseur, la mémoire centrale et les divers périphériques (écran, clavier, souris, etc.) ;
- Il met à disposition de l'utilisateur un certain nombre de commandes et il gère leur exécution (formatage des disquettes, copie de fichiers, création de répertoires, ...)
- Il détecte et récupère (lorsque c'est possible) les erreurs et en informe l'utilisateur.

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Un système d'exploitation est composé

- d'un **noyau** (ou **kernel** en anglais), qui assure la gestion des processus (programmes exécutés par le microprocesseur). C'est réellement le cœur de l'O.S.
- d'autres composants logiciels assurant les fonctions du système d'exploitation :
 - gestion de fichiers ;
 - gestion de la mémoire ;
 - gestion des entrées-sorties ;
 - interpréteur de commandes ;
 - ...

Types d'O.S. : systèmes mono-tâches

Il ne gère qu'une seule tâche à la fois (un seul programme). Quand le programme est lancé, il utilise seul les ressources de la machine (processeur et mémoire notamment) et ne rend la main à l'O.S. qu'en fin d'exécution ou en cas d'erreur.

Par exemple, le MS-DOS (MicroSoft Disk Operating System).

Types d'O.S. : systèmes mono-tâches

Il ne gère qu'une seule tâche à la fois (un seul programme). Quand le programme est lancé, il utilise seul les ressources de la machine (processeur et mémoire notamment) et ne rend la main à l'O.S. qu'en fin d'exécution ou en cas d'erreur.

Par exemple, le MS-DOS (MicroSoft Disk Operating System).

Types d'O.S. : systèmes mono-tâches

Il ne gère qu'une seule tâche à la fois (un seul programme). Quand le programme est lancé, il utilise seul les ressources de la machine (processeur et mémoire notamment) et ne rend la main à l'O.S. qu'en fin d'exécution ou en cas d'erreur.

Par exemple, le MS-DOS (MicroSoft Disk Operating System).

Types d'O.S. : systèmes multi-tâches

Le système multi-tâche (mais mono-utilisateur) gère simultanément plusieurs programmes sur une même machine. Il permet de partager le temps du processeur pour plusieurs programmes, de façon à ce que ceux-ci semblent s'exécuter simultanément. Le principe est d'allouer du temps (de traitement par le processeur) à différents programmes (tâches, processus) fonctionnant en même temps. Ces tâches sont tour à tour actives (exécutées par le processeur), en attente, suspendues ou détruites, suivant une priorité qui leur est associée. Le temps alloué peut être fixe ou variable suivant le type de partage géré par l'O.S.

Types d'O.S. : systèmes multi-tâches

Le système multi-tâche (mais mono-utilisateur) gère simultanément plusieurs programmes sur une même machine. Il permet de partager le temps du processeur pour plusieurs programmes, de façon à ce que ceux-ci semblent s'exécuter simultanément. Le principe est d'allouer du temps (de traitement par le processeur) à différents programmes (tâches, processus) fonctionnant en même temps. Ces tâches sont tour à tour actives (exécutées par le processeur), en attente, suspendues ou détruites, suivant une priorité qui leur est associée. Le temps alloué peut être fixe ou variable suivant le type de partage géré par l'O.S.

Types d'O.S. : systèmes multi-tâches

Le système multi-tâche (mais mono-utilisateur) gère simultanément plusieurs programmes sur une même machine. Il permet de partager le temps du processeur pour plusieurs programmes, de façon à ce que ceux-ci semblent s'exécuter simultanément. Le principe est d'allouer du temps (de traitement par le processeur) à différents programmes (tâches, processus) fonctionnant en même temps. Ces tâches sont tour à tour actives (exécutées par le processeur), en attente, suspendues ou détruites, suivant une priorité qui leur est associée. Le temps alloué peut être fixe ou variable suivant le type de partage géré par l'O.S.

Types d'O.S. : systèmes multi-tâches

Le système multi-tâche (mais mono-utilisateur) gère simultanément plusieurs programmes sur une même machine. Il permet de partager le temps du processeur pour plusieurs programmes, de façon à ce que ceux-ci semblent s'exécuter simultanément. Le principe est d'allouer du temps (de traitement par le processeur) à différents programmes (tâches, processus) fonctionnant en même temps. Ces tâches sont tour à tour actives (exécutées par le processeur), en attente, suspendues ou détruites, suivant une priorité qui leur est associée. Le temps alloué peut être fixe ou variable suivant le type de partage géré par l'O.S.

Types d'O.S. : systèmes multi-tâches

Le système multi-tâche (mais mono-utilisateur) gère simultanément plusieurs programmes sur une même machine. Il permet de partager le temps du processeur pour plusieurs programmes, de façon à ce que ceux-ci semblent s'exécuter simultanément. Le principe est d'allouer du temps (de traitement par le processeur) à différents programmes (tâches, processus) fonctionnant en même temps. Ces tâches sont tour à tour actives (exécutées par le processeur), en attente, suspendues ou détruites, suivant une priorité qui leur est associée. Le temps alloué peut être fixe ou variable suivant le type de partage géré par l'O.S.

Types d'O.S. : systèmes multi-tâches

Le système multi-tâche (mais mono-utilisateur) gère simultanément plusieurs programmes sur une même machine. Il permet de partager le temps du processeur pour plusieurs programmes, de façon à ce que ceux-ci semblent s'exécuter simultanément. Le principe est d'allouer du temps (de traitement par le processeur) à différents programmes (tâches, processus) fonctionnant en même temps. Ces tâches sont tour à tour actives (exécutées par le processeur), en attente, suspendues ou détruites, suivant une priorité qui leur est associée. Le temps alloué peut être fixe ou variable suivant le type de partage géré par l'O.S.

Types d'O.S. : systèmes multi-tâches

Le système multi-tâche (mais mono-utilisateur) gère simultanément plusieurs programmes sur une même machine. Il permet de partager le temps du processeur pour plusieurs programmes, de façon à ce que ceux-ci semblent s'exécuter simultanément. Le principe est d'allouer du temps (de traitement par le processeur) à différents programmes (tâches, processus) fonctionnant en même temps. Ces tâches sont tour à tour actives (exécutées par le processeur), en attente, suspendues ou détruites, suivant une priorité qui leur est associée. Le temps alloué peut être fixe ou variable suivant le type de partage géré par l'O.S.

La phase de chargement

Le **chargement** est la phase de démarrage de l'ordinateur (mise sous tension), également appelée **boot sequence**. À la mise sous tension, le programme **boot loader** est chargé en mémoire, et s'exécute. Ce programme est une partie de l'O.S. et il procède au chargement des différents composants de celui-ci. L'ensemble du système d'exploitation se trouve donc chargé et en cours d'exécution.

La phase de chargement

Le **chargement** est la phase de démarrage de l'ordinateur (mise sous tension), également appelée **boot sequence**. À la mise sous tension, le programme **boot loader** est chargé en mémoire, et s'exécute. Ce programme est une partie de l'O.S. et il procède au chargement des différents composants de celui-ci. L'ensemble du système d'exploitation se trouve donc chargé et en cours d'exécution.

La phase de chargement

Le **chargement** est la phase de démarrage de l'ordinateur (mise sous tension), également appelée **boot sequence**. À la mise sous tension, le programme **boot loader** est chargé en mémoire, et s'exécute. Ce programme est une partie de l'O.S. et il procède au chargement des différents composants de celui-ci. L'ensemble du système d'exploitation se trouve donc chargé et en cours d'exécution.

La phase de chargement

Le **chargement** est la phase de démarrage de l'ordinateur (mise sous tension), également appelée **boot sequence**. À la mise sous tension, le programme **boot loader** est chargé en mémoire, et s'exécute. Ce programme est une partie de l'O.S. et il procède au chargement des différents composants de celui-ci. L'ensemble du système d'exploitation se trouve donc chargé et en cours d'exécution.

La phase de chargement

Le **chargement** est la phase de démarrage de l'ordinateur (mise sous tension), également appelée **boot sequence**. À la mise sous tension, le programme **boot loader** est chargé en mémoire, et s'exécute. Ce programme est une partie de l'O.S. et il procède au chargement des différents composants de celui-ci. L'ensemble du système d'exploitation se trouve donc chargé et en cours d'exécution.

La communication homme-machine

Dès que le système est chargé, l'utilisateur peut envoyer des ordres à l'ordinateur, que l'O.S. exécute. Ces ordres sont envoyés soit en mode **console** (ligne de commande), soit en mode **graphique** (par exemple en cliquant sur un bouton). Certains O.S. ne fonctionnent qu'en mode console (par exemple DOS). Les ordres peuvent également être lancés à travers l'exécution d'application ou de fichiers de commande (fichiers **batch**).

La communication homme-machine

Dès que le système est chargé, l'utilisateur peut envoyer des ordres à l'ordinateur, que l'O.S. exécute. Ces ordres sont envoyés soit en mode **console** (ligne de commande), soit en mode **graphique** (par exemple en cliquant sur un bouton). Certains O.S. ne fonctionnent qu'en mode console (par exemple DOS). Les ordres peuvent également être lancés à travers l'exécution d'application ou de fichiers de commande (fichiers **batch**).

La communication homme-machine

Dès que le système est chargé, l'utilisateur peut envoyer des ordres à l'ordinateur, que l'O.S. exécute. Ces ordres sont envoyés soit en mode **console** (ligne de commande), soit en mode **graphique** (par exemple en cliquant sur un bouton). Certains O.S. ne fonctionnent qu'en mode console (par exemple DOS). Les ordres peuvent également être lancés à travers l'exécution d'application ou de fichiers de commande (fichiers **batch**).

La communication homme-machine

Dès que le système est chargé, l'utilisateur peut envoyer des ordres à l'ordinateur, que l'O.S. exécute. Ces ordres sont envoyés soit en mode **console** (ligne de commande), soit en mode **graphique** (par exemple en cliquant sur un bouton). Certains O.S. ne fonctionnent qu'en mode console (par exemple DOS). Les ordres peuvent également être lancés à travers l'exécution d'application ou de fichiers de commande (fichiers **batch**).

La communication homme-machine

Dès que le système est chargé, l'utilisateur peut envoyer des ordres à l'ordinateur, que l'O.S. exécute. Ces ordres sont envoyés soit en mode **console** (ligne de commande), soit en mode **graphique** (par exemple en cliquant sur un bouton). Certains O.S. ne fonctionnent qu'en mode console (par exemple DOS). Les ordres peuvent également être lancés à travers l'exécution d'application ou de fichiers de commande (fichiers **batch**).

Services d'accès aux ressources

En général on n'utilise pas un ordinateur pour son système d'exploitation, mais on se sert de celui-ci pour utiliser des applications logicielles. Ces applications utilisent quant à elles des services de l'O.S. pour écrire ou lire un fichier sur le disque dur, pour connaître l'heure et la date du système, pour imprimer des caractères à l'écran, pour émettre de la musique via les enceintes acoustiques, etc. L'ensemble de ces services est fourni par l'O.S. et les développeurs d'applications n'ont pas besoin de les re-coder ; il suffit de savoir les appeler correctement. Ces services forment une **API** (pour **Application Programming Interface**). Par exemple, sous Windows on parle parfois de l'API Win 32. Du point de vue du programmeur, l'API est un ensemble de fonctions qu'il est possible d'appeler dans un programme afin d'obtenir des services de la part de l'O.S. C'est ainsi que chaque logiciel est conçu pour fonctionner sur un O.S. particulier.

Services d'accès aux ressources

En général on n'utilise pas un ordinateur pour son système d'exploitation, mais on se sert de celui-ci pour utiliser des applications logicielles. Ces applications utilisent quant à elles des services de l'O.S. pour écrire ou lire un fichier sur le disque dur, pour connaître l'heure et la date du système, pour imprimer des caractères à l'écran, pour émettre de la musique via les enceintes acoustiques, etc. L'ensemble de ces services est fourni par l'O.S. et les développeurs d'applications n'ont pas besoin de les re-coder ; il suffit de savoir les appeler correctement. Ces services forment une **API** (pour **Application Programming Interface**). Par exemple, sous Windows on parle parfois de l'API Win 32. Du point de vue du programmeur, l'API est un ensemble de fonctions qu'il est possible d'appeler dans un programme afin d'obtenir des services de la part de l'O.S. C'est ainsi que chaque logiciel est conçu pour fonctionner sur un O.S. particulier.

Services d'accès aux ressources

En général on n'utilise pas un ordinateur pour son système d'exploitation, mais on se sert de celui-ci pour utiliser des applications logicielles. Ces applications utilisent quant à elles des services de l'O.S. pour écrire ou lire un fichier sur le disque dur, pour connaître l'heure et la date du système, pour imprimer des caractères à l'écran, pour émettre de la musique via les enceintes acoustiques, etc. L'ensemble de ces services est fourni par l'O.S. et les développeurs d'applications n'ont pas besoin de les re-coder ; il suffit de savoir les appeler correctement. Ces services forment une API (pour **Application Programming Interface**). Par exemple, sous Windows on parle parfois de l'API Win 32. Du point de vue du programmeur, l'API est un ensemble de fonctions qu'il est possible d'appeler dans un programme afin d'obtenir des services de la part de l'O.S. C'est ainsi que chaque logiciel est conçu pour fonctionner sur un O.S. particulier.

Services d'accès aux ressources

En général on n'utilise pas un ordinateur pour son système d'exploitation, mais on se sert de celui-ci pour utiliser des applications logicielles. Ces applications utilisent quant à elles des services de l'O.S. pour écrire ou lire un fichier sur le disque dur, pour connaître l'heure et la date du système, pour imprimer des caractères à l'écran, pour émettre de la musique via les enceintes acoustiques, etc. L'ensemble de ces services est fourni par l'O.S. et les développeurs d'applications n'ont pas besoin de les re-coder ; il suffit de savoir les appeler correctement. Ces services forment une API (pour **Application Programming Interface**). Par exemple, sous Windows on parle parfois de l'API Win 32. Du point de vue du programmeur, l'API est un ensemble de fonctions qu'il est possible d'appeler dans un programme afin d'obtenir des services de la part de l'O.S. C'est ainsi que chaque logiciel est conçu pour fonctionner sur un O.S. particulier.

Services d'accès aux ressources

En général on n'utilise pas un ordinateur pour son système d'exploitation, mais on se sert de celui-ci pour utiliser des applications logicielles. Ces applications utilisent quant à elles des services de l'O.S. pour écrire ou lire un fichier sur le disque dur, pour connaître l'heure et la date du système, pour imprimer des caractères à l'écran, pour émettre de la musique via les enceintes acoustiques, etc. L'ensemble de ces services est fourni par l'O.S. et les développeurs d'applications n'ont pas besoin de les re-coder ; il suffit de savoir les appeler correctement. Ces services forment une **API** (pour **Application Programming Interface**). Par exemple, sous Windows on parle parfois de l'API Win 32. Du point de vue du programmeur, l'API est un ensemble de fonctions qu'il est possible d'appeler dans un programme afin d'obtenir des services de la part de l'O.S. C'est ainsi que chaque logiciel est conçu pour fonctionner sur un O.S. particulier.

Services d'accès aux ressources

En général on n'utilise pas un ordinateur pour son système d'exploitation, mais on se sert de celui-ci pour utiliser des applications logicielles. Ces applications utilisent quant à elles des services de l'O.S. pour écrire ou lire un fichier sur le disque dur, pour connaître l'heure et la date du système, pour imprimer des caractères à l'écran, pour émettre de la musique via les enceintes acoustiques, etc. L'ensemble de ces services est fourni par l'O.S. et les développeurs d'applications n'ont pas besoin de les re-coder ; il suffit de savoir les appeler correctement. Ces services forment une **API** (pour **Application Programming Interface**). Par exemple, sous Windows on parle parfois de l'API Win 32. Du point de vue du programmeur, l'API est un ensemble de fonctions qu'il est possible d'appeler dans un programme afin d'obtenir des services de la part de l'O.S. C'est ainsi que chaque logiciel est conçu pour fonctionner sur un O.S. particulier.

Services d'accès aux ressources

En général on n'utilise pas un ordinateur pour son système d'exploitation, mais on se sert de celui-ci pour utiliser des applications logicielles. Ces applications utilisent quant à elles des services de l'O.S. pour écrire ou lire un fichier sur le disque dur, pour connaître l'heure et la date du système, pour imprimer des caractères à l'écran, pour émettre de la musique via les enceintes acoustiques, etc. L'ensemble de ces services est fourni par l'O.S. et les développeurs d'applications n'ont pas besoin de les re-coder ; il suffit de savoir les appeler correctement. Ces services forment une **API** (pour **Application Programming Interface**). Par exemple, sous Windows on parle parfois de l'API Win 32. Du point de vue du programmeur, l'API est un ensemble de fonctions qu'il est possible d'appeler dans un programme afin d'obtenir des services de la part de l'O.S. C'est ainsi que chaque logiciel est conçu pour fonctionner sur un O.S. particulier.

Services d'accès aux ressources

En général on n'utilise pas un ordinateur pour son système d'exploitation, mais on se sert de celui-ci pour utiliser des applications logicielles. Ces applications utilisent quant à elles des services de l'O.S. pour écrire ou lire un fichier sur le disque dur, pour connaître l'heure et la date du système, pour imprimer des caractères à l'écran, pour émettre de la musique via les enceintes acoustiques, etc. L'ensemble de ces services est fourni par l'O.S. et les développeurs d'applications n'ont pas besoin de les re-coder ; il suffit de savoir les appeler correctement. Ces services forment une **API** (pour **Application Programming Interface**). Par exemple, sous Windows on parle parfois de l'API Win 32. Du point de vue du programmeur, l'API est un ensemble de fonctions qu'il est possible d'appeler dans un programme afin d'obtenir des services de la part de l'O.S. C'est ainsi que chaque logiciel est conçu pour fonctionner sur un O.S. particulier.

Fonctions complexes : gestion de l'unité centrale

La gestion de l'unité centrale (processeur) consiste à distribuer le (temps de traitement du) processeur entre les différents travaux à exécuter. Cette fonction est simple dans le cas d'un système mono-tâche. En ce qui concerne un système multi-tâche, la concurrence pour l'accès au processeur entre les différents travaux est gérée par l'attribution d'un niveau de priorité et d'une portion de temps de l'unité centrale.

Fonctions complexes : gestion de l'unité centrale

La gestion de l'unité centrale (processeur) consiste à distribuer le (temps de traitement du) processeur entre les différents travaux à exécuter. Cette fonction est simple dans le cas d'un système mono-tâche. En ce qui concerne un système multi-tâche, la concurrence pour l'accès au processeur entre les différents travaux est gérée par l'attribution d'un niveau de priorité et d'une portion de temps de l'unité centrale.

Fonctions complexes : gestion de l'unité centrale

La gestion de l'unité centrale (processeur) consiste à distribuer le (temps de traitement du) processeur entre les différents travaux à exécuter. Cette fonction est simple dans le cas d'un système mono-tâche. En ce qui concerne un système multi-tâche, la concurrence pour l'accès au processeur entre les différents travaux est gérée par l'attribution d'un niveau de priorité et d'une portion de temps de l'unité centrale.

Fonctions complexes : gestion des processus

Dans le cas de systèmes multi-tâches, on parle de **gestion des processus**. Un **processus** est un programme en cours d'exécution. Un processus peut être **actif** (en cours d'exécution) ou **inactif** (en attente d'exécution). Il ne peut y avoir qu'un seul processus actif à la fois. Le changement d'état (actif \leftrightarrow inactif) est ordonné par l'O.S. suite à une interruption (un signal) reçue par un dispositif d'entrée-sortie, soit à intervalle régulier de manière à répartir le temps entre les processus. Un système est dit **préemptif** lorsqu'il possède un **ordonnanceur** (ou **planificateur**) qui répartit, selon des critères de priorité, le temps processeur entre les différents processus qui en font la demande.

Fonctions complexes : gestion des processus

Dans le cas de systèmes multi-tâches, on parle de **gestion des processus**. Un **processus** est un programme en cours d'exécution. Un processus peut être **actif** (en cours d'exécution) ou **inactif** (en attente d'exécution). Il ne peut y avoir qu'un seul processus actif à la fois. Le changement d'état (actif \leftrightarrow inactif) est ordonné par l'O.S. suite à une interruption (un signal) reçue par un dispositif d'entrée-sortie, soit à intervalle régulier de manière à répartir le temps entre les processus. Un système est dit **préemptif** lorsqu'il possède un **ordonnanceur** (ou **planificateur**) qui répartit, selon des critères de priorité, le temps processeur entre les différents processus qui en font la demande.

Fonctions complexes : gestion des processus

Dans le cas de systèmes multi-tâches, on parle de **gestion des processus**. Un **processus** est un programme en cours d'exécution. Un processus peut être **actif** (en cours d'exécution) ou **inactif** (en attente d'exécution). Il ne peut y avoir qu'un seul processus actif à la fois. Le changement d'état (actif \leftrightarrow inactif) est ordonné par l'O.S. suite à une interruption (un signal) reçue par un dispositif d'entrée-sortie, soit à intervalle régulier de manière à répartir le temps entre les processus. Un système est dit **préemptif** lorsqu'il possède un **ordonnanceur** (ou **planificateur**) qui répartit, selon des critères de priorité, le temps processeur entre les différents processus qui en font la demande.

Fonctions complexes : gestion des processus

Dans le cas de systèmes multi-tâches, on parle de **gestion des processus**. Un **processus** est un programme en cours d'exécution. Un processus peut être **actif** (en cours d'exécution) ou **inactif** (en attente d'exécution). Il ne peut y avoir qu'un seul processus actif à la fois. Le changement d'état (actif \leftrightarrow inactif) est ordonné par l'O.S. suite à une interruption (un signal) reçue par un dispositif d'entrée-sortie, soit à intervalle régulier de manière à répartir le temps entre les processus. Un système est dit **préemptif** lorsqu'il possède un **ordonnanceur** (ou **planificateur**) qui répartit, selon des critères de priorité, le temps processeur entre les différents processus qui en font la demande.

Fonctions complexes : gestion des processus

Dans le cas de systèmes multi-tâches, on parle de **gestion des processus**. Un **processus** est un programme en cours d'exécution. Un processus peut être **actif** (en cours d'exécution) ou **inactif** (en attente d'exécution). Il ne peut y avoir qu'un seul processus actif à la fois. Le changement d'état (actif \leftrightarrow inactif) est ordonné par l'O.S. suite à une interruption (un signal) reçue par un dispositif d'entrée-sortie, soit à intervalle régulier de manière à répartir le temps entre les processus. Un système est dit **préemptif** lorsqu'il possède un **ordonnanceur** (ou **planificateur**) qui répartit, selon des critères de priorité, le temps processeur entre les différents processus qui en font la demande.

Fonctions complexes : gestion des processus

Dans le cas de systèmes multi-tâches, on parle de **gestion des processus**. Un **processus** est un programme en cours d'exécution. Un processus peut être **actif** (en cours d'exécution) ou **inactif** (en attente d'exécution). Il ne peut y avoir qu'un seul processus actif à la fois. Le changement d'état (actif \leftrightarrow inactif) est ordonné par l'O.S. suite à une interruption (un signal) reçue par un dispositif d'entrée-sortie, soit à intervalle régulier de manière à répartir le temps entre les processus. Un système est dit **préemptif** lorsqu'il possède un **ordonnanceur** (ou **planificateur**) qui répartit, selon des critères de priorité, le temps processeur entre les différents processus qui en font la demande.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Fonctions complexes : gestion de la mémoire centrale

Les applications occupent et utilisent de la place en mémoire. L'O.S. a aussi pour rôle de gérer en permanence un **plan d'occupation de la mémoire**, c'est-à-dire qu'il gère l'organisation et l'optimisation de la mémoire. L'O.S. gère également le chargement des programmes en mémoire centrale. En cas de besoin, l'O.S. répartit ses besoins entre la mémoire vive et la mémoire virtuelle (place allouée sur le disque dur jouant le rôle d'une mémoire vive) : c'est le **swapping**. Pour répartir la mémoire entre différentes applications, plusieurs techniques sont employées :

- La **pagination** : les programmes sont découpés en blocs de longueur fixe correspondant aux segments de mémoire alloués.
- La **segmentation** permet un découpage plus fin et adapté aux besoins. Les programmes ne sont pas découpés en blocs de taille fixe mais en segments dont la taille dépend du programme.

Un des problèmes importants quant à la gestion de la mémoire est la **réallocation** de la mémoire lorsque celle-ci est libérée ou la réallocation de la mémoire non libérée par des applications. Certains systèmes disposent d'un dispositif de **ramasse miettes** (ou **garbage collector**) qui récupère la mémoire non libérée.

Un des problèmes importants quant à la gestion de la mémoire est la **réallocation** de la mémoire lorsque celle-ci est libérée ou la réallocation de la mémoire non libérée par des applications. Certains systèmes disposent d'un dispositif de **ramasse miettes** (ou **garbage collector**) qui récupère la mémoire non libérée.

Fonctions complexes : gestion des entrées-sorties

Cette fonction consiste à prendre en charge l'échange d'information entre l'ordinateur et le monde extérieur (les périphériques). Cette gestion comprend à la fois l'installation (ajout, suppression, mise à jour) et le pilotage des périphériques (comment peut-on communiquer avec eux ?). Ces différentes fonctions sont assurées par des pilotes (ou drivers) appropriés à chaque matériel.

Fonctions complexes : gestion des entrées-sorties

Cette fonction consiste à prendre en charge l'échange d'information entre l'ordinateur et le monde extérieur (les périphériques). Cette gestion comprend à la fois l'installation (ajout, suppression, mise à jour) et le pilotage des périphériques (comment peut-on communiquer avec eux ?). Ces différentes fonctions sont assurées par des pilotes (ou drivers) appropriés à chaque matériel.

Fonctions complexes : gestion des entrées-sorties

Cette fonction consiste à prendre en charge l'échange d'information entre l'ordinateur et le monde extérieur (les périphériques). Cette gestion comprend à la fois l'installation (ajout, suppression, mise à jour) et le pilotage des périphériques (comment peut-on communiquer avec eux ?). Ces différentes fonctions sont assurées par des **pilotes** (ou **drivers**) appropriés à chaque matériel.

Fonctions complexes : gestion des fichiers

Le système de gestion de fichiers (une partie de l'O.S.) prend en charge les opérations de lecture et écriture de fichiers sur des ressources autre que la mémoire centrale (disque dur, clef USB, etc.). Il gère à la fois les fichiers, les répertoires et tient à jour la **table d'allocation** (**File Allocation Table**) qui permet d'associer le fichier à sa localisation physique sur le disque dur, mais également de gérer l'espace disponible sur celui-ci. Un certain nombre de commandes et d'utilitaires sont mis à disposition de l'utilisateur pour gérer ses fichiers et ses répertoires (copie, renommage, création de répertoire, formatage, etc.).

Fonctions complexes : gestion des fichiers

Le système de gestion de fichiers (une partie de l'O.S.) prend en charge les opérations de lecture et écriture de fichiers sur des ressources autre que la mémoire centrale (disque dur, clef USB, etc.). Il gère à la fois les fichiers, les répertoires et tient à jour la **table d'allocation (File Allocation Table)** qui permet d'associer le fichier à sa localisation physique sur le disque dur, mais également de gérer l'espace disponible sur celui-ci. Un certain nombre de commandes et d'utilitaires sont mis à disposition de l'utilisateur pour gérer ses fichiers et ses répertoires (copie, renommage, création de répertoire, formatage, etc.).

Fonctions complexes : gestion des fichiers

Le système de gestion de fichiers (une partie de l'O.S.) prend en charge les opérations de lecture et écriture de fichiers sur des ressources autre que la mémoire centrale (disque dur, clef USB, etc.). Il gère à la fois les fichiers, les répertoires et tient à jour la **table d'allocation (File Allocation Table)** qui permet d'associer le fichier à sa localisation physique sur le disque dur, mais également de gérer l'espace disponible sur celui-ci. Un certain nombre de commandes et d'utilitaires sont mis à disposition de l'utilisateur pour gérer ses fichiers et ses répertoires (copie, renommage, création de répertoire, formatage, etc.).