République du Sénégal

***Un peuple-Un but -Une Foi***

**Ministère de l’Enseignement Supérieur de la Recherche et de l’Innovation**

****

**GROUPE ISM**

**SOUTENANCE LICENCE 1**

**Théorie des systèmes d’exploitation**

**L1B GLRS**

**MODÉRATEURS:**

**Kokou Godwin TCHAKPANA**

**MABUILA NKOUNGA Franck-Ba**

**Moutagha Abou DIA**

**Okana Gildas MANDOU**

**PROFESSEUR :**

M. **Massamba LO**

1. **GESTIONS DES FICHIERS**
2. **La notion du système de fichiers**

**Un système de fichier** encore appelé ***système de gestion de fichiers*** ou ***file system*** en anglais, est une façon de stocker des informations et de les organiser dans des fichiers sur ce qu'on appelle en génie logiciel des mémoire secondaire (pour le matériel informatique il s'agit de mémoire de masse comme un disque dur). Le volume des données traités par les applications pouvant être énorme (pouvant atteindre plusieurs téraoctets) ne peuvent être stocké à long terme dans la mémoire centrale, il est nécessaire de les stocker dans des mémoires secondaires sous forme de suites de blocs ou fichiers. Un bloc est la plus petite unité que le périphérique de stockage est capable de gérer.

Pour l'utilisateur un système de fichier est vu comme un ensemble de fichiers qui sont regrouper dans des répertoires contenus eux même dans d'autres répertoires avec ou sans d'autres fichiers. Dans un système de fichier on a donc une arborescence avec une racine, des dossiers et des fichiers qui peuvent être organisé de manière indépendante. Une telle gestion de fichiers permet de traiter et de conserver des quantités importantes de données ainsi que de les partager entre plusieurs programmes informatiques. Il offre à l'utilisateur une vue abstraite sur ses données et permet de la localiser à partir d'un chemin d'accès.

1. **Comment nommer un fichier ?**

En informatique, **un fichier** est au sens commun, une collection, un ensemble de données numériques réunies sous un même nom, enregistrées sur un support de stockage permanent, appelé mémoire de masse, tel qu'un disque dur. Il faut noter que le nom d'un fichier est une chaîne de caractère souvent de taille limitée. De nos jours quasiment tous les caractères sont utilisables mais certains sont interdits ou déconseillés car ayant un sens pour le système d'exploitation. C'est par exemple le cas des ***deux points*** (:) et de ***la barre oblique*** (/).

Avec l’essor de la technologie, les systèmes d'exploitation disposent de nos jours de différentes méthodes pour associer un nom de fichier a son contenu. Par exemple :

* Pour Windows chaque répertoire contient une table associant les noms des fichiers à leur taille et un index vers la table d'allocation de fichiers, une zone réservée du disque indiquant pour chaque bloc de données l'index du bloc suivant du même fichier.
* Pour Unix les fichiers et les répertoires sont identifiés par un numéro unique (inode) ce qui permet d'accéder à une structure de données regroupant toutes les informations sur un fichier (comme la protection d'accès en lecture) à l'exception du nom.

En outre, les systèmes d'exploitation ont différentes méthodes de codage :

* Sur Unix les noms des fichiers sont basés sur une suite d'octets. Les couches de base de logiciel (noyau, systèmes de fichiers) manipulent les noms des fichiers sans connaître le jeu de caractère utilisé. Afin d'éviter des erreurs de compatibilité l'UTF-8 est de plus en plus utilisée notamment dans les distributions Linux.
* Sous Windows et dans les environnements graphiques le nom d'un fichier possède en général un suffixe (extension) séparé par un point qui est fonction du contenu de fichiers. Par exemple ***.txt*** pour le texte.

1. **L’utilité du système de fichier**

Le système de fichiers assure plusieurs fonctions :

* ***La manipulation des fichiers*** : créer, détruire des fichiers, insérer, supprimer ou modifier un article dans un fichier.
* ***L'allocation de place mémoire secondaire*** : le SGF alloue à chaque fichier un nombre valable de granule de mémoire secondaire de taille fix.
* ***La localisation des fichiers*** : chaque fichier possède un ensemble d'informations descriptives (nom, adresse...) regroupé dans un inode.
* **La sécurité et le contrôle les fichiers** : un nom et une clé de protection sont associés à chaque fichier afin de le protéger contre tout accès non autorisé lors du partage de fichier et le SGF doit garantir la conservation des fichiers en cas de panne du matériel ou du logiciel.
* ***Le bon fonctionnement et l'organisation du fichier*** : l'utilisateur peut organiser ses données en permanence et les distribuer dans différents fichiers. Le contenu des fichiers est déterminé par leur format qui dépend de l'application utilisée.

Le choix du système de gestion de fichiers se fait principalement en fonction du système d'exploitation. Les plus récents supportent un grand nombre de système de fichier. On distingue différents types de système de fichiers :

* **Non journalisés** comme ext (EXtented file system) pour Linux, Windows ;
* **Journalisés** : ces systèmes de fichiers enregistrent les modifications dans un journal avant de les effectuer sur les fichiers eux-mêmes, ce qui permet la récupération de modification "en cours" en cas d'arrêt intempestif. On a par exemple ext3 pour Linux et NTFS (New Technology File System) pour Windows ;
* **À Snapshot** : les systèmes de fichiers à snapshot ou instantané en français offre la possibilité d'enregistrer l'état du système de fichiers en un instant donné. Par exemple APFS (Apple File System) pour macIOS et iOS et ACFS pour Linux et Windows ;
* **Réseau** : comme AFS (Andrew File System) pour Linux, Windows et Mac OS ;
* **Cluste**r : comme GPFS (General Parallèle File System) pour Linux et Windows ;
* **Spécialisés** : comme SQUASHFS, système de fichier compressé en lecture seule pour Linux ;
* **Temporaires** : comme RAMFS le système de fichiers temporaire en mémoire RAM le plus simple et efficace qui soit basé sur la gestion du cache du noyau Linux ;
* **Méta systèmes de fichiers** : tel que UNIONFS, une implémentation dans le noyau Linux consistant à fusionner différents systèmes de fichiers en un unique point de montage ;
* **Pseudo systèmes de fichiers** : il s'agit d’une interface particulière au 7e logiciel généralement le système d'exploitation. Il montre sur l'apparence d'une pratique des fichiers et de répertoire, des informations de nature variée (l'état de l'ordinateur, des périphériques...). Par exemple UDEV qui a remplacé DEVFS pour Linux 2.6
* **À classer** tel que ADFS (Ascorn Disc File System) pour Linux

**Quelques exemples de systèmes de fichiers utilisés**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Système de fichiers** | **Créateur** | **Année d'introduction** | **Système d’exploitation initial** | **License** |
| **LTFS** | **IBM** | **2010** | **Linux, Mac OS X, planned Microsoft Windows,** | **LGPL** |
| **IlesfayFS** | **Ilesfay Technology Group** | **2011** | **Microsoft Windows, planned Red Hat Enterprise Linux** |  |
| **VMFS5** | **VMware** | **2011** | **VMware ESXi 5.0tux 3 stats** |  |
| **ReFS** | **Microsoft** | **2012, 2013** | **Windows Server 2012** |  |
| **Lanyard Filesystem** | **Dan Luedtke** | **2012** | **Linux** |  |
| **F2FS** | **Samsung** | **2012** | **Linux** | **GPLv2** |
| **APFS** | **Apple** | **2016** | **MacOs** |  |

1. **GESTION DE LA MEMOIRE**

En ce qui concerne la gestion de la mémoire, on ne peut parler de cette dernière sans parler du gestionnaire de mémoire alors, en ce qui nous concerne nous commencerons par aborder la problématique du gestionnaire de mémoire puis s’en suivra la suite du processus.

1. **Le gestionnaire de mémoire**

Le **gestionnaire de mémoire** est un sous-ensemble du système d'exploitation. Son rôle est de partager la mémoire entre l'OS et les diverses applications. Le terme "mémoire" fait surtout référence la mémoire principale, c'est à dire à la RAM, mais la gestion de celle-ci demande la contribution de la mémoire auxiliaire (mémoire de masse, spacieuse mais lente) et à la mémoire cache (rapide mais de taille restreinte).

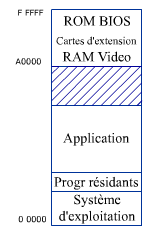
Nous avons 2 types de gestion de mémoire :

1. **La gestion de mémoire pour un systèmes monontâches**

Dans le cas des systèmes monotâche, la gestion de la mémoire est assez simple. Il suffit de réserver une partie de la mémoire au système d'exploitation. L'application est ensuite casée dans l'espace restant qui est libéré sitôt que l'application est terminée.

Cela se complique un peu si l'application nécessite plus d'espace que ce que peut fournir la mémoire vive. On segmente alors l'application en ***segments de recouvrements*** ou *"****Overlays****"*. Cette technique n'a plus cours maintenant. Elle était utilisée à l'époque du DOS pour des applications volumineuses. Le programmeur devait prévoir le découpage de son application en imaginant comment ces overlays serait chargés en mémoire les uns à la suite des autres pour qu'au cours de son exécution l'application puisse atteindre toutes les fonctions nécessaires.

C'était en quelque sorte comme cela qu'on concevait la mémoire virtuelle à l'époque des systèmes d'exploitation monotâche. Notez que la zone mémoire occupée par le système d'exploitation n'était pas protégée par ce type de gestion de mémoire.



1. **La gestion de mémoire à systèmes multitâches**

Plusieurs processus doivent se partager la mémoire sans empiéter sur l'espace réservé au système d'exploitation ni aux autres processus. Quand un processus se termine, le S.E. doit libérer l'espace mémoire qui lui était alloué pour pouvoir y placer de nouveaux processus.

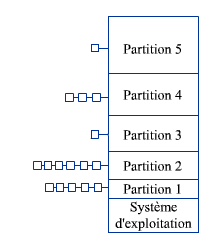
Pour se faire, le plus simple est de diviser la mémoire en partitions fixes dès le démarrage du système. Les partitions sont de différentes tailles pour éviter que de grandes partitions ne soient occupées que par de petits processus. Le gestionnaire de mémoire, en fonction de la taille des processus, décide quelle partition lui allouer pour ne pas gaspiller trop de mémoire.

Une file d'attente est associée à chaque partition. Quand vient une nouvelle tâche, le gestionnaire détermine quelle est la plus petite partition qui peut la contenir puis place cette tâche dans la file correspondante.

Le fait d'éviter d'allouer une partition trop grande à un petit processus conduit parfois à des aberrations. Il arrive que des partitions plus grandes restent inutilisées alors que se forment ailleurs des files interminables de petits processus. La mémoire est donc mal utilisée.

Une autre solution est de créer une file unique. Lorsqu'une partition se libère, on consulte la file pour trouver la tâche qui l'occuperait de manière optimale.

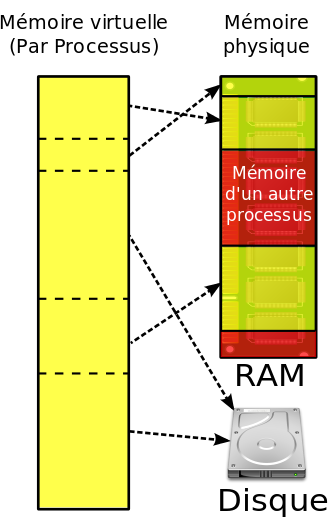
Le risque est que les petites tâches soient pénalisées. Une parade est de conserver une petite partition au moins qui ne sera accessible qu'aux petites tâches. Une autre solution, serait de dire qu'un processus ne peut être ignorer qu'au maximum un certain nombre de fois. Après n refus, il prendra place dans une partition même si la partition est bien plus grande qu'il ne faut.



1. **La mémoire virtuelle**

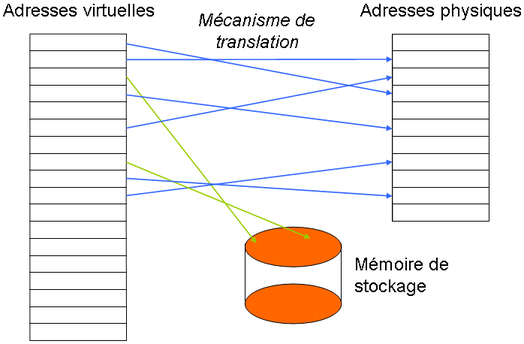
En informatique, le mécanisme de **mémoire virtuelle** a été mis au point dans les [années 1960](https://fr.wikipedia.org/wiki/Ann%C3%A9es_1960). Il repose sur l'utilisation de traduction à la volée des adresses (virtuelles) vues du logiciel, en adresses physiques de [mémoire vive](https://fr.wikipedia.org/wiki/M%C3%A9moire_vive). La mémoire virtuelle permet :

* D’utiliser de la [mémoire de masse](https://fr.wikipedia.org/wiki/M%C3%A9moire_de_masse) comme extension de la [mémoire vive](https://fr.wikipedia.org/wiki/M%C3%A9moire_vive) ;
* D’augmenter le [taux de multiprogrammation](https://fr.wikipedia.org/wiki/Taux_de_multiprogrammation) ;
* De mettre en place des mécanismes de [protection de la mémoire](https://fr.wikipedia.org/wiki/M%C3%A9moire_virtuelle#Protection) ;
* De partager la mémoire entre [processus](https://fr.wikipedia.org/wiki/Processus_informatique).



Les adresses mémoires émises par le processeur sont des adresses virtuelles, indiquant la position d'un mot dans la mémoire virtuelle. Cette mémoire virtuelle est formée de zones de même taille, appelées pages. Une adresse virtuelle est donc un couple (numéro de page, déplacement dans la page). La mémoire vive est également composée de zones de même taille, appelées cadres (frames en anglais), dans lesquelles prennent place les pages (un cadre contient une page : taille d'un cadre = taille d'une page). La taille de l'ensemble des cadres en mémoire vive utilisés par un processus est appelée Resident set size.

Un mécanisme de traduction (translation, ou génération d'adresse) assure la conversion des adresses virtuelles en adresses physiques, en consultant une table des pages (page table en anglais) pour connaître le numéro du cadre qui contient la page recherchée. L'adresse physique obtenue est le couple (numéro de cadre, déplacement). Il peut y avoir plus de pages que de cadres (c'est là tout l'intérêt) : les pages qui ne sont pas en mémoire sont stockées sur un autre support (disque), elles seront ramenées dans un cadre quand on en aura besoin.



1. **Le swapping**

L'**espace d'échange**, en [anglais](https://fr.wikipedia.org/wiki/Anglais) ***swap***, est une partie de la [mémoire de masse](https://fr.wikipedia.org/wiki/M%C3%A9moire_de_masse) d'un [ordinateur](https://fr.wikipedia.org/wiki/Ordinateur) utilisée par le [système d'exploitation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) pour stocker des données qui, du point de vue des [applications](https://fr.wikipedia.org/wiki/Application_(informatique)), se trouvent en [mémoire vive](https://fr.wikipedia.org/wiki/M%C3%A9moire_vive). L'espace d'échange peut prendre la forme d'une [partition](https://fr.wikipedia.org/wiki/Partition_(informatique)) dédiée (la **partition d'échange**, courante sous les systèmes [Unix](https://fr.wikipedia.org/wiki/Type_Unix)) ou d'un simple [fichier](https://fr.wikipedia.org/wiki/Fichier_informatique) (le **fichier d'échange**, C:\pagefile.sys sous [Windows](https://fr.wikipedia.org/wiki/Windows) par exemple), ou de plusieurs partitions et/ou fichiers. La mémoire vive et l'espace d'échange constituent ensemble la [mémoire virtuelle](https://fr.wikipedia.org/wiki/M%C3%A9moire_virtuelle) du système.

Sur certains systèmes tels que [GNU/Linux](https://fr.wikipedia.org/wiki/GNU/Linux), la partition d'échange est aussi utilisée pour la mise en [hibernation](https://fr.wikipedia.org/wiki/Hibernation_(informatique)), ou veille sur disque, du système.

Il est parfois nécessaire de supprimer *toutes* les pages ou segments d'un processus de la mémoire centrale. Dans ce cas le processus sera dit swappé, et toutes les données lui appartenant seront stockées en mémoire de masse. Cela peut survenir pour des processus dormant depuis longtemps, alors que le système d'exploitation a besoin d'allouer de la mémoire aux processus actifs. Les pages ou segments de code (programme) ne seront jamais swappés, mais tout simplement réassignés, car on peut les retrouver dans le fichier correspondant au programme (le fichier de l'exécutable). Pour cette raison, le système d'exploitation interdit l'accès en écriture à un fichier exécutable en cours d'utilisation ; symétriquement, il est impossible de lancer l'exécution d'un fichier tant qu'il est tenu ouvert pour un accès en écriture par un autre processus.

1. **Le mapping**

La mémoire cache ne pouvant contenir toute la mémoire principale il faut définir une méthode mémoire principale, il faut définir une méthode indiquant à quelle adresse de la mémoire cache doit être écrite une ligne de la mémoire principale. Cette méthode s'appelle le mapping.

Il existe trois types de mapping :

1. **La mémoire cache complètement associative (fully associative cache)**

Chaque ligne de la mémoire de niveau supérieur peut être écrite sur n'importe quelle adresse du cache. Cette méthode demande beaucoup de logique car elle offre de nombreuses possibilités. Cela explique pourquoi l'associativité complète n'est utilisée que dans les petits caches.

1. **La mémoire cache directe (direct mapped cache)**

La ligne de cache est partagée par de nombreux niveaux d'adresses mémoire Prédominant. Par conséquent, nous avons besoin d'un moyen de comprendre le courant caché. Ces informations sont fournies par l'étiquette, qui est stockée dans le cache. L'index correspond à la ligne où les données sont enregistrées. De plus, le contrôleur Le cache doit savoir si une ligne contient des données. Un peu Le supplément (appelé bit de validité) indique si la ligne est libre.

1. **Mémoire cache N-associative**

Il s'agit d'un compromis entre la cartographie directe et la cartographie entièrement associative, en essayant de combiner la simplicité d'une cartographie avec l'efficacité d'une autre. Le cache est Divisé en une collection de N lignes de cache. Une ligne de la mémoire de niveau supérieur a été allouée à l'assembly, elle peut donc être écrite sur n'importe quel canal.

Ceci permet d'éviter de nombreux échecs de cache conflictuels. Au sein d'un ensemble, le mapping est complètement associatif. Habituellement, sélectionnez le groupe de la manière suivante : Set = Mod memory address

1. **LA GESTION DES INTERRUPTIONS**

La gestion des interruptions est la partie du noyau système chargée de répondre aux signaux(événements). Sa fonction est de deux ordres : déterminer la source d’interruption et répondre à l’interruption

1. **Qu’est-ce qu’une interruption ?**

En informatique, une interruption est une suspension temporaire de l'exécution normale d'un programme informatique par le microprocesseur afin d'exécuter un programme prioritaire appelé « *service d'interruption* ». C’est également est un évènement qui provoque l'arrêt du programme en cours et provoque le branchement du microprocesseur à un sous-programme particulier dit de "*traitement de l'interruption*". En effet, un programme peut entrer dans une boucle permanente. Dans ce cas seule l’intervention de l’usager ou du système au bout d’un délai plus ou moins long peut permettre d’interrompre l’exécution du programme fautif. Il faut pour cela disposer d’un mécanisme matériel permettant d’interrompre l’exécution de la boucle par un signal externe au programme. C’est une des utilisations de la notion d’interruption. Introduite pour la première fois en 1953, les interruptions sont utilisées par les machines pour gérer diverses fonctions. Elles sont principalement utilisées pour permettre des communications non bloquantes avec des périphériques externes. De plus, elles servent à commuter les tâches dans un ordonnanceur.

1. **Les Types d’interruption**

En informatique, il existe principalement 3 types d’interruptions :

1. **Les interruptions matérielles ou externes**

Les interruptions matérielles (ou externes, comme dans externes au processeur) sont des signaux émis par des composantes physiques qui brisent le flot d’exécution du système d’exploitation de telle sorte qu’il doive s’interroger à savoir quoi faire pour bien leur répondre. Les tics d’horloge, les données arrivant de la souris ou du clavier, un signal du disque rigide signifiant que celui-ci est prêt à transférer des données peuvent tous être source d’interruptions. En outre, Une interruption matérielle est causée par un périphérique matériel tel qu’une demande de démarrage d’une E/S, une défaillance matérielle ou quelque chose de similaire. Les interruptions matérielles ont été introduites comme un moyen d’éviter de perdre le temps précieux du processeur dans les boucles d’interrogation, en attendant des événements externes. Elles sont classées en deux types :

* **Interruptions masquables :** Les processeurs doivent interrompre le registre de masque qui permet d'activer et de désactiver les interruptions matérielles. Chaque signal a un bit placé dans le registre de masque. Si ce bit est défini, une interruption est activée et désactivée lorsqu'un bit n'est pas défini, ou vice versa. Les signaux qui interrompent les processeurs via ces masques sont appelés interruptions masquées.
* **Interruptions non masquables (NMI)**: Les NMI sont les activités de priorité la plus élevée qui doivent être traitées immédiatement et dans n'importe quelle situation, comme un signal de temporisation généré par un temporisateur de surveillance**.**

1. **Les interruptions logicielles ou internes**

Les interruptions logicielles (ou internes, comme dans internes au processeur) sont semblables aux interruptions matérielles. L'unique différence réside dans le fait que les interruptions logicielles sont émises par des programmes. Les interruptions logicielles sont générées volontairement par un processus en exécution (un de vos programmes, par exemple), donc à même le processeur. On compte parmi celles-ci toute la gamme des erreurs (instructions invalides, division par zéro, erreurs d'accès à la mémoire, ...). Une interruption logicielle est invoquée par l’utilisation de l’instruction INT. Cet événement arrête immédiatement l’exécution du programme et transmet l’exécution au gestionnaire INT. Le gestionnaire INT fait généralement partie du système d’exploitation et détermine l’action à entreprendre. Ce sont des interruptions provoquées par le programmeur. Le programmeur utilise une instruction qui déclenche une interruption. Les interruptions logicielles ont un effet similaire à un appel de fonction avec une différence fondamentale.

1. **Les Exceptions**

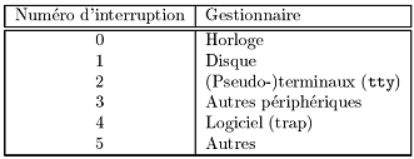
Une exception (abréviation de "***événement exceptionnel***") est une erreur ou un événement inattendu se produisant lorsqu’un programme est en cours d'exécution. Lorsqu'une exception se produit, le flux du programme est interrompu. Si le programme peut gérer et traiter l'exception, il peut continuer à s'exécuter. Si une exception n'est pas gérée, le programme peut être obligé de quitter. Les exceptions surviennent quand un évènement logiciel spécial arrive, une erreur d’exécution de programme. Par exemple : instruction invalide ; division par 0 ; référence à une adresse invalide ; accès invalide à une adresse protégée. Les exceptions ont un très haut niveau de priorité parce le microprocesseur est dans une impasse : il ne peut exécuter l’instruction en cours en raison d’une erreur de programmation. Lorsqu'une exception se produit, l'exécution normale du programme est interrompue et l'exception est traitée. L'utilisation des gestionnaires d'exceptions s'est généralisée sur PC avec l'utilisation du mode protégé sous DOS puis avec les systèmes d'exploitation multitâches. Auparavant, une erreur de programmation.

1. **Différences entre les interruptions matérielles et logicielles**

|  | ***Interruption matérielle*** | ***Interruption logicielle*** |
| --- | --- | --- |
| 1 | Une interruption matérielle est une interruption générée à partir d’un périphérique ou d’un matériel externe. | L’interruption logicielle est l’interruption générée par tout système interne de l’ordinateur. |
| 2 | Il n’incrémente pas le compteur de programme. | Il incrémente le compteur de programme. |
| 3 | L’interruption matérielle peut être invoquée avec un périphérique externe tel qu’une demande de démarrage d’E/S ou l’occurrence d’une défaillance matérielle. | L’interruption logicielle peut être invoquée à l’aide de l’instruction INT. |
| 4 | Il a la priorité la plus basse que les interruptions logicielles | Il a la priorité la plus élevée parmi toutes les interruptions. |
| 5 | L’interruption matérielle est déclenchée par du matériel externe et est considérée comme l’un des moyens de communiquer avec les périphériques externes, le matériel. | L’interruption logicielle est déclenchée par le logiciel et considérée comme l’un des moyens de communiquer avec le noyau ou de déclencher des appels système, en particulier lors de la gestion des erreurs ou des exceptions. |
| 6 | C’est un événement asynchrone. | C’est un événement synchrone. |
| 7 | Les interruptions matérielles peuvent être classées en deux types :  1. Interruption masquable.  2. Interruption non masquable. | Les interruptions logicielles peuvent être classées en deux types :  1. Interruptions normales.  2. Exception |
| 8 | Les enfoncements des touches et les mouvements de la souris sont des exemples d’interruption matérielle. | Tous les appels système sont des exemples d’interruptions logicielles |

1. **Traitement des interruptions**

Le traitement d’une interruption par le système s’effectue en appelant une ***routine*** de traitement associée à cette interruption, dont l’adresse est stockée dans les cases d’un tableau indexé par le type d’interruption : ***la table de vecteurs d’interruption.***

******

Des numéros d’interruptions différents sont couramment affectés aux différents périphériques du système, et permettront d’accéder aux routines de traitement à travers le vecteur d’interruption. Le traitement d’une interruption s’effectue par le système de la façon suivante :  
 - Le processeur sauvegarde la valeur de son compteur ordinal, détermine le type d’interruption, passe en mode noyau et charge la nouvelle valeur du compteur ordinal à partir du vecteur d’interruption;  
 - La routine de traitement de l’interruption sauvegarde les autres registres du processeur puis appelle la procédure principale de gestion de l’interruption;  
 - Au retour de la procédure de gestion, la routine de traitement restaure les registres du processeur, puis recharge la valeur du compteur ordinal.

Ainsi lorsqu’une interruption survient :

**Le dispatcher** s’occupe de la sauvegarde du contexte du processus en cours, de l’extraction du contexte du processus le plus adéquat et du transfert du contrôle à l’adresse désignée par le compteur ordinal du processus restauré.

**Le scheduler** détermine quel processus doit être exécuté. Il suffit pour cela d’ordonner les processus prêts selon un critère donné qui détermine leur priorité. Le processus élu sera celui qui se situe, selon l’ordre de priorité, en tête de la file des processus prêts. Lorsqu’une interruption survient le processus actif peut perdre la main au profit (des modules du système d’exploitation et par la suite) d’un processus parmi la liste des processus prêts.  
Les liaisons entre le gestionnaire d’interruption et le dispatcher sont représentées par l’ensemble d’actions suivantes :

