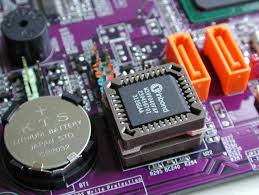
Le BIOS (Basic Input Output System)

Le BIOS est un programme que l’on retrouve sur une puce de mémoire morte.

Physiquement :



**Rôle du BIOS au démarrage :**

1. POST (Power On Self Test)

Vérification du fonctionnement de quelques composantes de l’ordinateur (CPU, RAM (lecture et écriture dans les cellules de la RAM), cache, clavier…).

Si le POST rencontre un problème, il envoie une suite de signaux sonores de durées différentes (un peu comme le code Morse). Cette suite de signaux est différente de celle envoyée si tout fonctionne normalement.

Pour connaître la signification de la suite de signaux : Beep code dans la documentation de la carte mère ou en tapant la marque du Bios dans un moteur de recherche (ex Award Bios Beep Codes). (Il existe différents codes selon la marque du BIOS (AMI, Award,…)).

Les signaux sont produits par un petit haut-parleur sur la carte-mère. (Voir plus haut, à côté de la pile)

1. Configuration du matériel

Après le POST, le Bios s’occupe des configurations du matériel.

Ex : RAM ( taux de rafraîchissement…), clavier (vitesse de répétition des caractères…), cœurs (activés ou non…), ports (activés ou non)…

Ces caractéristiques sont gardées dans un type spécial de mémoire : le CMOS (on dit aussi la NVRAM). Le CMOS est une puce distincte de celle qui contient le Bios.

On a donc :

Bios sur ROM Données du Bios sur CMOS (NVRAM)

Pile du CMOS

Pourquoi deux types de mémoire ? (ROM et CMOS)

Les données du Bios doivent être facilement modifiable (ex : si veut désactiver un port)

ROM : pas assez facilement modifiable.

On pense donc à de la RAM. Mais la RAM perdra son contenu lorsqu’on éteindra l’ordinateur. On utilisera donc une RAM spéciale qui gardera tout le temps son contenu grâce à une pile.

Les utilisateurs peuvent changer les valeurs des données du Bios en allant dans le CMOS (SETUP). On y reviendra plus loin dans ce texte.

Le système (l’ordinateur) doit aussi pouvoir modifier les données du CMOS. On a vu que le CMOS contient la liste des caractéristiques (configurations) du matériel. La liste des composantes matérielles est donc contenue dans le CMOS. À chaque fois qu’on fait une modification matérielle (ex : ajout d’un disque dur), cette modification doit apparaître dans le CMOS. L’ordinateur détecte en général les changements de matériel (CPU, disques durs…) et modifie lui-même les données dans le CMOS (Plug-and-Play).

En résumé, dans le CMOS, on retrouve entre autres la liste des composantes matérielles de l’ordinateur ainsi que leur configuration.

1. Chargement en RAM de certain pilotes (drivers)

Pour voir ce qu’est un pilote, faisons un rappel tiré du fichier *cpu Partie 1 – générations 1 à 5* :

Chaque micro-processeur a son propre langage, le langage-machine. Ce langage contient les seules instructions que le micro-processeur peut exécuter. Ce sont des instructions très simples.

Lorsque vous écrivez un programme en langage évolué (par exemple, le C), un compilateur fera la traduction du langage évolué vers le langage machine.

Reprenons l’exemple de l’instruction C=A+B. Cette instruction est en langage évolué. Elle est trop complexe pour le micro-processeur. Le compilateur la traduira pour donner plusieurs instructions plus simples que le micro-processeur pourra exécuter. Il remplacera aussi les noms de variables par des adresses.

C= A+B devient :

* Aller chercher le contenu de la variable A en RAM et la mettre dans le registre 1
* Aller chercher le contenu de la variable B en RAM et la mettre dans le registre 2
* Faire l’addition des deux registres et placer le résultat dans le registre 3
* Aller porter le contenu du registre 3 en RAM dans la variable C

Et en remplaçant les variables par leur adresse en RAM, on obtient :

(En supposant que la variable A a été placée à l’adresse 0068, la variable B à l’adresse 0069 et la variable C à l’adresse 0070)

* Aller chercher le contenu de l’adresse 0068 en RAM et la mettre dans le registre 1
* Aller chercher le contenu de l’adresse 0069 en RAM et la mettre dans le registre 2
* Faire l’addition des deux registres et placer le résultat dans le registre 3
* Aller porter le contenu du registre 3 en RAM à l’adresse 0070

En langage machine, on a :

* LOAD Reg1 0068
* LOAD Reg2 0069
* ADD Reg1, Reg2, Reg3
* STORE Reg3 0070

|  |  |  |
| --- | --- | --- |
| Instructions | Description des signaux pour exécuter les instructions | registres |
| ADD |  |  |
| SUB |  |  |
| MUL |  |  |
| DIV |  |  |
| CMP |  |  |
| LOAD |  |  |
| LDB |  | Cache L1 |
| STORE |  |
| … |  |
|  |  |  |
|  |  |

Donc, si on résume :

Programme en langage évolué (ex : C) 🡪COMPILATEUR 🡪Programme en langage machine du cpu 🡪 Exécution

FIN DU RAPPEL…

Un contrôleur étant aussi un processeur a aussi son langage machine.

Instructions venant du système d’exploitation 🡪PILOTE (Driver) 🡪Commandes en langage machine du contrôleur 🡪 Exécution

Exemple : l’instruction à exécuter par le CPU est une instruction pour le disque dur (par exemple, lire dans le fichier f1.txt).

CPU RAM

|  |  |  |
| --- | --- | --- |
| lire dans le fichier f1.txt |  | Programme en C |
|  |  |
| Unité de contrôle |  |

L’unité de contrôle donne au système d’exploitation la commande à exécuter (lire dans le fichier f1.txt).

CPU RAM

|  |  |  |
| --- | --- | --- |
| lire dans le fichier f1.txt |  | Programme en C |
| Système d’exploitation |  |
| Unité de contrôle | Système d’exploitation |

Le système d’exploitation envoie l’instruction au bon pilote (ici, celui du disque dur).

CPU RAM

|  |  |  |
| --- | --- | --- |
| lire dans le fichier f1.txt |  | Programme en C |
| Système d’exploitation | Pilote du disque dur |
| Unité de contrôle | Système d’exploitation |

Le pilote reçoit l’instruction à exécuter (ouvrir le fichier F1.docx )et la traduit dans le langage du contrôleur qui lui, génère les signaux pour le faire.

CPU RAM

|  |  |  |
| --- | --- | --- |
| lire dans le fichier f1.txt |  | Programme en C |
|  | Pilote du disque dur |
| Unité de contrôle | Système d’exploitation |

|  |  |  |
| --- | --- | --- |
|  | Contrôleur de disque dur |  |

Disque dur

En résumé :

Instructions venant du système d’exploitation (Lire dans le fichier f1.txt…)

🡪PILOTE du contrôleur de disque dur

🡪Commandes en langage machine du contrôleur

🡪 Exécution

Vous voyez maintenant pourquoi, lorsqu’on télécharge un nouveau pilote, on vous pose 2 questions : avec quel système d’exploitation et avec quel contrôleur ce pilote sera-t-il utilisé…

La plupart des pilotes sont exécutés une fois le système d’exploitation chargé.

Par contre, on a besoin de certain périphériques (ex : écran) avant que le système d’exploitation soit chargé.

Solution : le Bios contient certain pilotes de base, (ex : écran : juste assez pour faire afficher quelques informations (pas d’interface graphique))

L’étape (iii) du démarrage est donc le chargement de pilotes de base (écran, clavier, disque dur ).

1. Chargement du SE

Le Bios consulte ses données dans le CMOS pour voir à quel endroit (disque dur local, réseau, clé USB…) il doit aller chercher le système d’exploitation.

Une fois qu’il a cette information, le Bios charge le système d’exploitation en RAM.

Les pilotes plus performants (dont l’écran) sont alors exécutés.

**Accès au CMOS**

On a vu qu’on retrouve dans le CMOS la liste des composantes physiques et leur configuration ainsi que l’ordre pour le chargement du SE.

On retrouve aussi la possibilité de mettre un mot de passe au Bios.

Pour accéder au CMOS (par abus de langage, on dit quelquefois accéder au Bios…), on doit appuyer sur une touche lors démarrage (la touche dépend du Bios de votre carte mère). Ce peut être par exemple F2.

Voici des exemples de ce que vous pouvez voir à l’écran à ce moment :





Voir le manuel (User’s guide) de la carte Gygabyte **B460M DS3H.**

**Modification du BIOS (Flash du Bios)**

Il est possible de télécharger une version plus récente d’un Bios (par exemple, Bios qui ne gère pas la totalité de notre disque dur).

Procédure : sur le site de la carte-mère, aller chercher la version voulue de Bios. (ref; site de la carte-mère ).

ATTENTION : ne faire cette procédure que si nécessaire et avec beaucoup de précautions. (Copie de sécurité)

Certaines cartes-mères possèdent deux Bios pour plus de sécurité lors d’un éventuel téléchargement de Bios



Le BIOS UEFI (celui qu’on vient de voir)

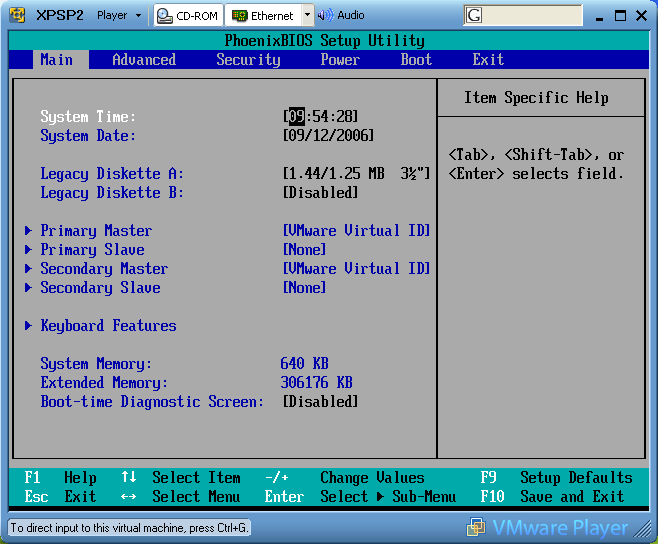
* Interface graphique (+souris)
* Peut gérer jusqu’à 128 partitions
* Peut gérer un disque dur de 9.4 Zeta-octets (Tera-octets, Peta-octets, Exa-octets, Zeta-octets)

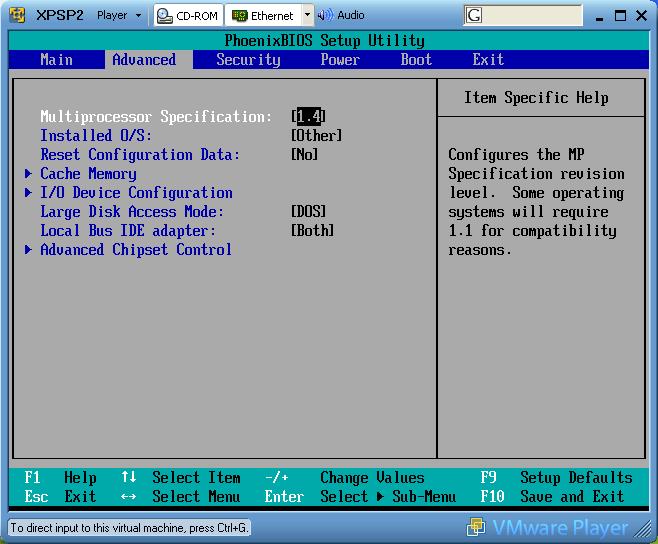
<https://www.youtube.com/watch?v=nOu7k9wDwoo>

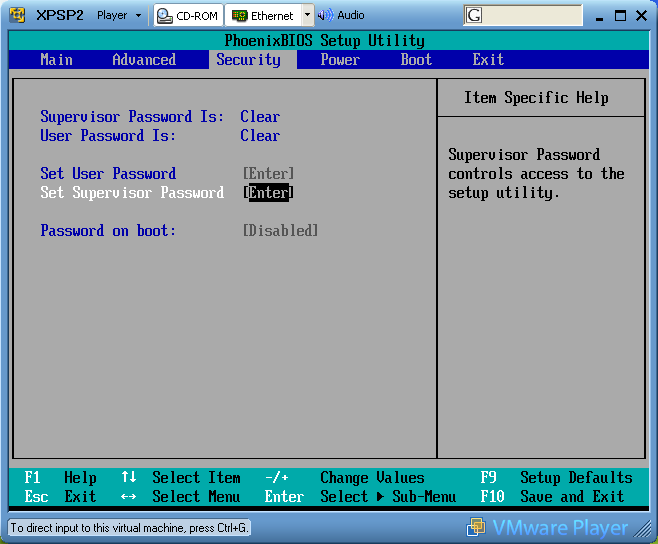
<https://www.youtube.com/watch?v=8EfjJubHTqw> (3D)

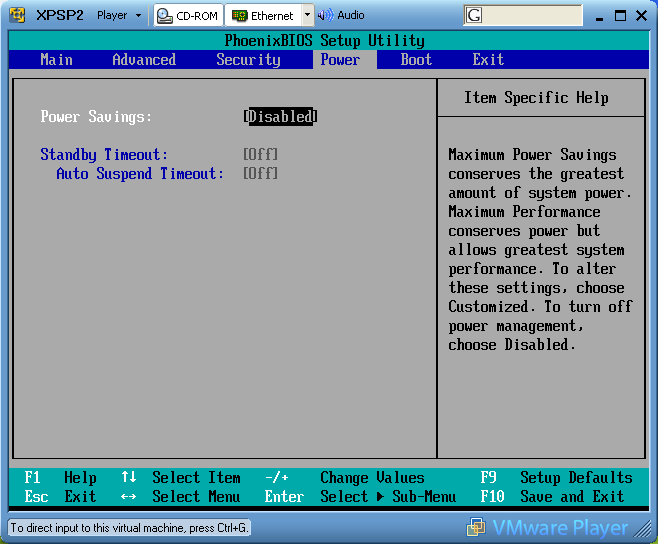
<http://www.gigabyte.fr/MicroSite/304/images/3d-bios.html>

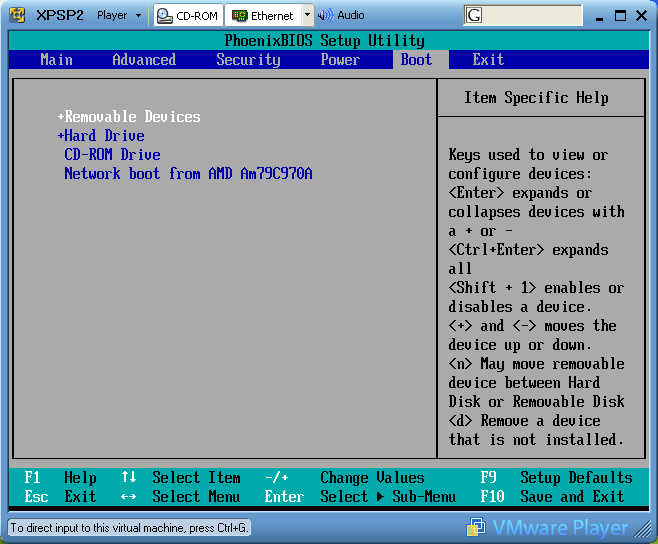
Le Bios Legacy (Le Bios avant le Bios UEFI)

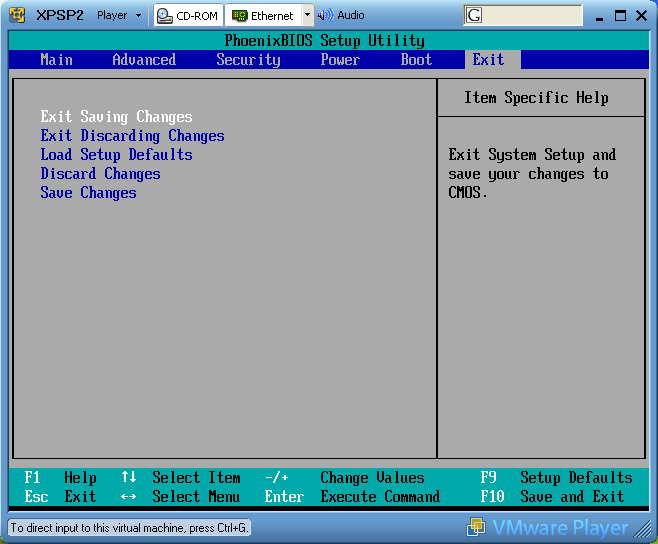












**BIOS UEFI**

*Unified Extensible Firmware Interface*

Des setup de BIOS nouvelles générations!

Résumé :

|  |  |  |
| --- | --- | --- |
|  | BIOS | BIOS UEFI |
| Visuel | Basique | Amélioré |
| Partitions | 4 | 128 |
| Grosseur disque max. | 2.2To | 9.4 millards To (9.4Zo) |
|  |  |  |