

Le processus de démarrage

par [Romain Guy \(Gfx\)](#)

Date de publication : 6 Juin 2006

Dernière mise à jour :

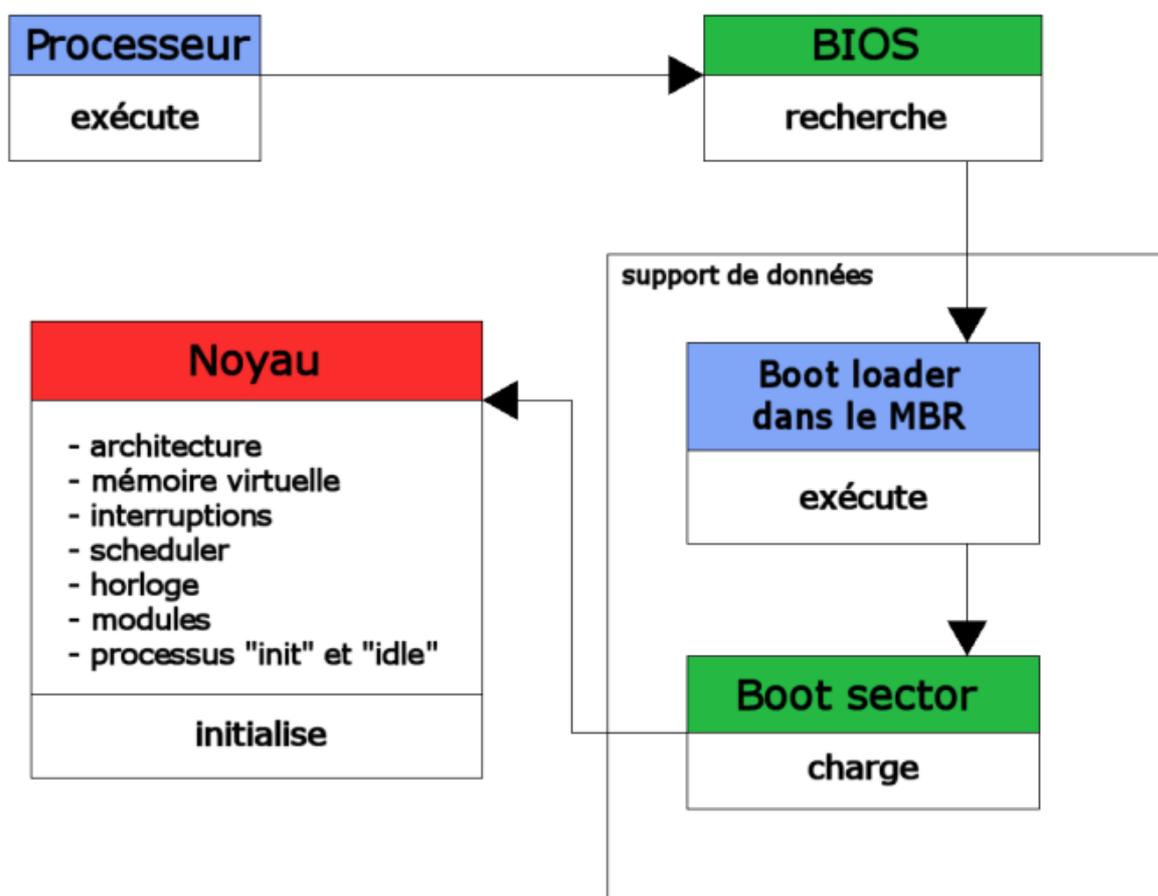
Linux, à l'instar de tout système d'exploitation, effectue une série d'opérations lors de son chargement. L'étude de celles-ci vous permettra de les configurer et de mieux appréhender cette étape importante.

- I - Introduction
- II - Chargement du noyau
- III - Les premiers processus
- IV - Niveaux d'exécution
- V - Scripts de niveaux

I - Introduction

Après avoir étudié la génération, et donc son contenu, d'un nouveau noyau adapté à vos besoins, nous allons maintenant examiner sa phase d'amorçage. Une fois l'ordinateur mis sous tension, le processeur lance l'exécution du BIOS (Basic Input Output System). Le rôle du BIOS consiste à analyser la configuration matérielle de l'ordinateur et de la déclarer apte pour le service. Une fois le système initialisé correctement, le BIOS déclenche une interruption (0x19 ou 25 en décimal) précédant la recherche d'un système d'exploitation.

La recherche s'effectue suivant un ordre prédéfini au sein du " setup " (configuration du BIOS). En règle générale, le BIOS commence sa recherche sur le lecteur de disquette ou le lecteur CD-Rom puis sur le disque dur. Sur chaque lecteur susceptible de contenir un système d'exploitation, le BIOS lit le premier secteur, appelé MBR (Master Boot Record) pour un disque dur ou secteur d'amorçage (boot sector) dans le cas d'une disquette. Ce premier secteur contient un petit programme et une table des partitions. Le petit programme, ou " boot loader ", est chargé de désigner la partition active du média. Cependant le lecteur contenant le système d'exploitation recèle un programme un peu plus complexe qui va non seulement désigner la partition active, mais également désigner la partition d'amorçage et l'activer. Linux emploie généralement LILO comme boot loader. Finalement, le secteur de début (c'est à dire le premier secteur de la partition d'amorçage autre que le MBR) contient à son tour un programme qui chargera le noyau.



Séquence de démarrage

II - Chargement du noyau

Avant de démarrer le noyau, Linux effectue deux choses d'une importance capitale (nous traitons ici de processeurs de la famille x86 Intel). Premièrement, le processeur est commuté en mode protégé. En effet, les processeurs Intel démarrent toujours, à l'allumage de l'ordinateur, en mode réel. Ensuite, un code initialisant l'exploitation du processeur en mode 32 bits est exécuté. Le démarrage du noyau peut alors commencer#

Les différentes étapes du processus de démarrage du noyau de Linux sont les suivantes :

- initialisation en fonction de l'architecture matérielle
- initialisation de la mémoire virtuelle
- initialisation des interruptions, du planificateur (le " scheduler ") et de l'horloge
- traitement des arguments de la ligne de commande
- initialisation des modules
- exécution d'un nouveau processus appelant la fonction " init " du noyau

La fonction " init " se charge ensuite, entre autres opérations, d'exécuter successivement les programmes portant le nom " init " dans les répertoires /etc, /bin et /sbin. Notons que si le lancement de init s'achève sans erreur dans le premier ou le second répertoire, alors le noyau ne tentera pas de réitérer l'opération dans les autres répertoires. Nous reviendrons plus tard sur les agissements de cette fonction.

III - Les premiers processus

Les deux premiers processus démarrés se nomment " idle " et " init " et portent, par une fantaisie d'un certain Linus T., les numéros d'identification 0 et 1. La seule tâche affectée à idle est de consommer le temps processeur qui n'est pas exploité par les autres processus. idle se voit aidé dans son entreprise par le scheduler qui répartit en permanence le temps processeur entre tous les processus. Et si d'aventure aucun processus ne demande de temps de calcul, le scheduler s'en remet à idle pour occuper un peu votre foudre de guerre.

Heureusement, " init " se révèle un peu plus intéressant et efficace que cet imbécile de " idle ". Ce processus s'avère être le parent de tous les processus du système (hormis le singulier " idle "). Les processus sont en effet classés suivant une arborescence dont chaque branche définit un processus possédant obligatoirement un parent et éventuellement un ou plusieurs enfants. Le processus init effectue un contrôle des partitions puis effectue le montage du système de fichiers principal sous /.

IV - Niveaux d'exécution

Le programme " init " exécuté va ensuite permettre de faire la distinction entre divers modes de fonctionnement de Linux, appelés niveaux d'exécution. Chaque niveau d'exécution exécute une série de scripts de démarrage. Vous trouverez l'ensemble de ces scripts dans l'arborescence " /etc/rc.d/init.d ". Le fonctionnement de ces scripts est décrit plus en détails dans la partie suivante. Le choix du niveau d'exécution sous lequel Linux doit être démarré est défini par le fichier de configuration nommé /etc/inittab.

La première ligne de ce fichier spécifie le niveau d'exécution. En règle générale, il s'agit du niveau 3 qui correspond au mode multi-utilisateurs :

Les 7 niveaux d'exécution classiques :

- 0 - halte
- 1 - mode utilisateur unique
- 2 - mode multi-utilisateurs sans gestion NFS
- 3 - mode mutli-utilisateurs
- 4 - sans effet
- 5 - mode X11
- 6 - redémarrage

Le mode d'initialisation ne doit sous aucun prétexte être assigné aux modes 0 et 6 sans quoi vous rencontreriez de gros problèmes pour démarrer le système d'exploitation. Le mode 5 quant à lui permet de démarrer le système en niveau 3 avec en sus un gestionnaire de login graphique tel que kdm ou xdm. La ligne définissant le niveau par défaut ressemble à celle-ci :

```
id:3:initdefault:
```

Chaque ligne du fichier inittab suit une syntaxe très précise de la forme : <id>:<niveau>:<mode>:<commande>

Le premier champ est un identifiant unique de 1 à quatre caractères. Certains identifiants permettent d'effectuer des opérations particulières.

Le champ " niveau " définit les actions à effectuer suivant le niveau d'initialisation choisi. Ainsi, la ligne " l1:1:wait:/sbin/init.d/rc 1 " ne sera exécutée que si le niveau d'initialisation est 1.

Le troisième champ permet de définir le mode d'exécution de la commande :

Modes d'exécution (liste non exhaustive) :

once : la commande n'est exécutée qu'une fois

off : la commande n'est pas exécutée

wait : attente de l'exécution de la commande

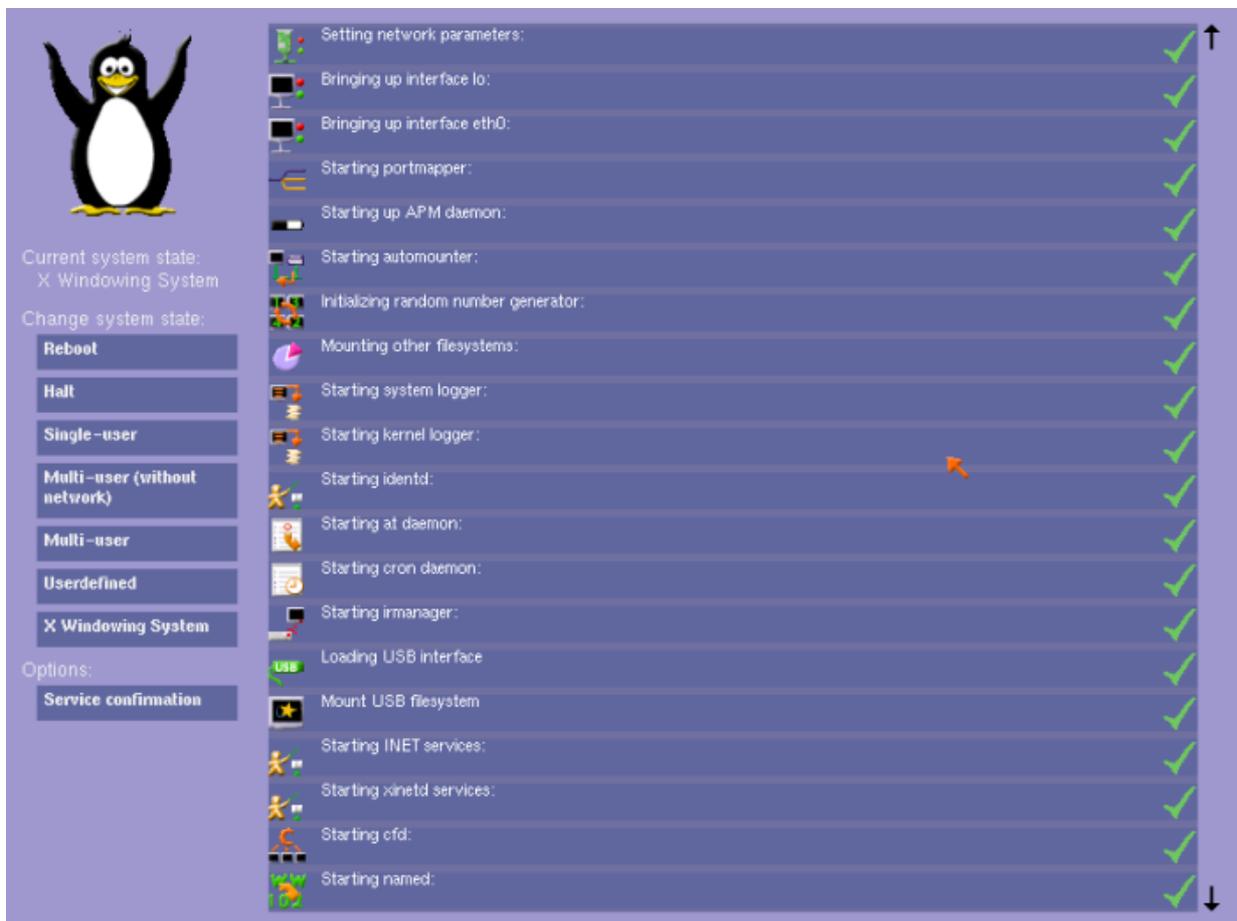
boot : la commande doit s'exécuter durant l'amorçage, le niveau est ignoré

bootwait : comme précédemment, mais le boot ne se poursuit qu'après la fin de la commande

ctrlaltdel : la commande est appelée lors de l'appui sur Ctrl+Alt+Del

initdefault : mode d'exécution de Linux

Pour finir, le quatrième champ contient une commande qui sera exécutée par le bash. Au préalable, le fichier /etc/initscript, s'il existe, est lancé car il contient des commandes qui sont traitées en priorité.



Une interface graphique à init : Aurora

V - Scripts de niveaux

Lors du passage à un niveau d'exécution donné, fichier nommé " rc " pilote ce passage. L'emplacement de ce fichier varie d'une distribution à l'autre. Dans notre cas, le fichier est nommé /etc/rc, qui est en réalité un lien symbolique vers /etc/rc.d/rc (il en va de même pour les répertoires /etc/rc*.d qui sont des liens vers /etc/rc.d/rc*.d). Ce fichier rc va définir deux variables d'environnement nommées " runlevel " et " previous " (ces noms peuvent également varier) contenant respectivement le niveau d'exécution demandé et le niveau d'exécution précédent.

Vient ensuite le changement de niveau à proprement parler. Ce changement consiste à exécuter une série de scripts destinés à démarrer et à arrêter des services. Les scripts sont placés dans les répertoires /etc/rc*.d (où * est le niveau d'exécution). Lors du changement de niveau, tous les scripts commençant par " K " (pour " kill ") dans le répertoire correspondant au niveau précédent seront exécutés avec l'argument " stop ". Ensuite, les scripts commençant par un " S " dans le répertoire du nouveau niveau seront lancés avec l'argument " start ". Les scripts K et S sont en fait des liens vers des scripts placés dans /etc/rc.d/init.d et qui exécutent des tâches différentes suivant l'argument (start ou stop).