

---

# systemd processus de démarrage

## Processus de démarrage du système

Plusieurs composants sont impliqués dans le démarrage du système. Immédiatement après la mise sous tension, le BIOS fait une initialisation hardware minimale, et donne la main au chargeur de boot stocké dans un périphérique de stockage. Ce chargeur va ainsi invoquer un Kernel. Dans le cas de Linux, ce kernel extrait (optionnellement) et exécute une image disque initiale en RAM (initrd), tel que généré par dracut(8), qui recherche le système de fichier racine (possiblement en utilisant systemd). Une fois trouvé et monté, l'initrd donne le contrôle au gestionnaire du système (tel que systemd), qui est ensuite responsable pour gérer tout le hardware restant, en montant tous les systèmes de fichiers nécessaires et en lançant tous les services configurés.

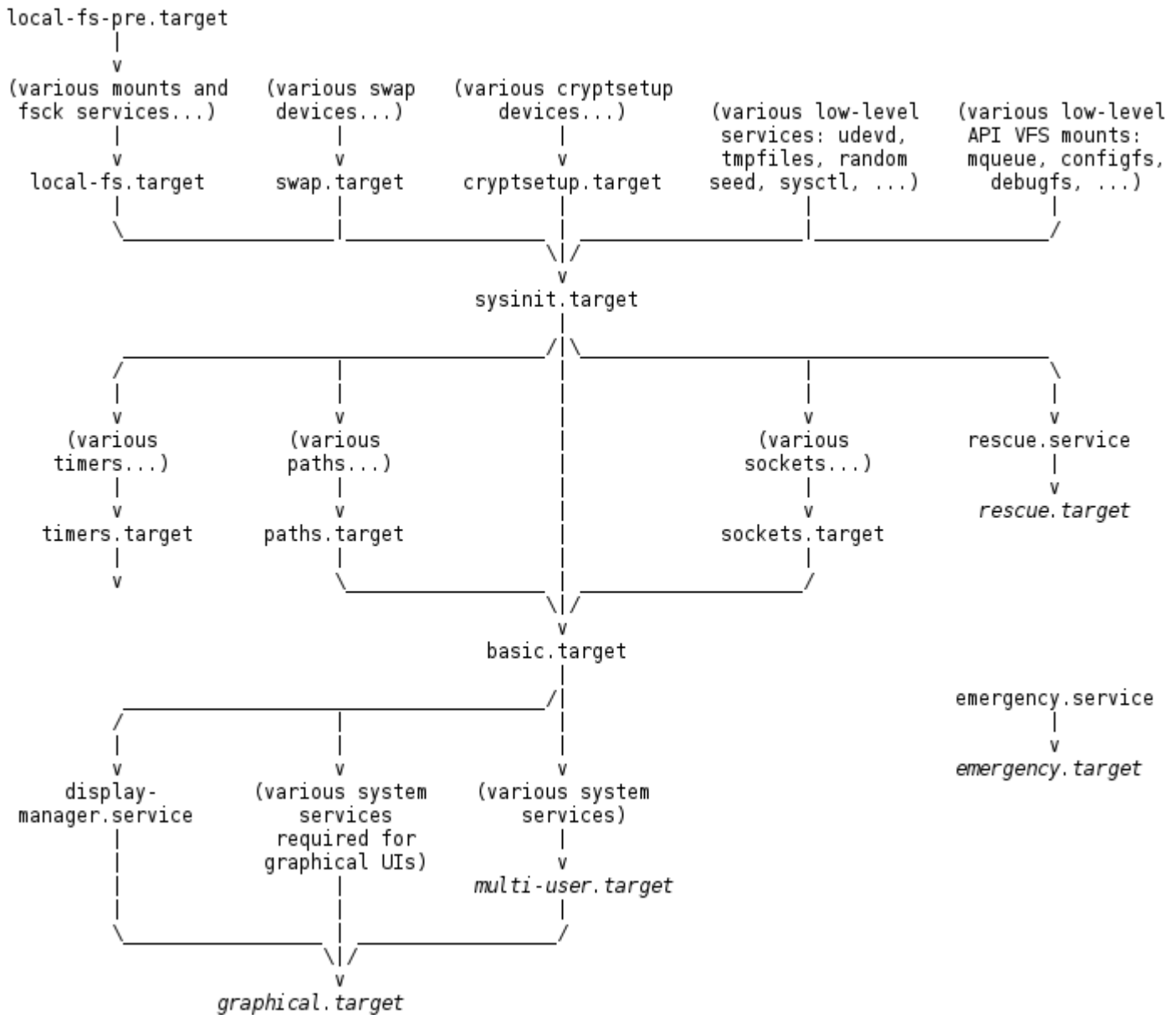
Lors de l'arrêt, le gestionnaire stop tous les services, démonte tous les systèmes de fichiers, et optionnellement saute dans le code initrd qui démonte le système racine. Enfin, le système est arrêté.

## Gestionnaire système au démarrage

Au démarrage, le gestionnaire système dans l'image OS est responsable d'initialiser les systèmes de fichiers, services et pilotes requis, qui sont nécessaires pour le système. Dans les systèmes systemd, ce processus est splitté en divers étapes qui sont exposés en tant qu'unités cible. Le processus de boot est hautement parallélisé pour que l'ordre dans lequel les unités cible soient atteints ne soit pas déterministe, mais adhère à une structure ordonnée limitée.

Quand systemd démarre le système, il active toutes les unités qui sont des dépendances de `default.target`. Généralement, `default.target` est simplement un alias de `graphical.target`, ou `multi-user.target`, en fonction de la configuration du système. Pour forcer un ordre minimal entre les unités, un nombre d'unité cible sont disponibles.

Le tableau suivant est une vue structurelle des unités et leur position dans le boot. Les unités en haut sont démarrées avant les unités plus bas.



Les unités cible qui sont communément utilisées comme cible de boot sont en italique. Ces unités sont un bon choix comme cible principal, par exemple en passant au kernel l'option `systemd.unit=` en en lien `default.target`.

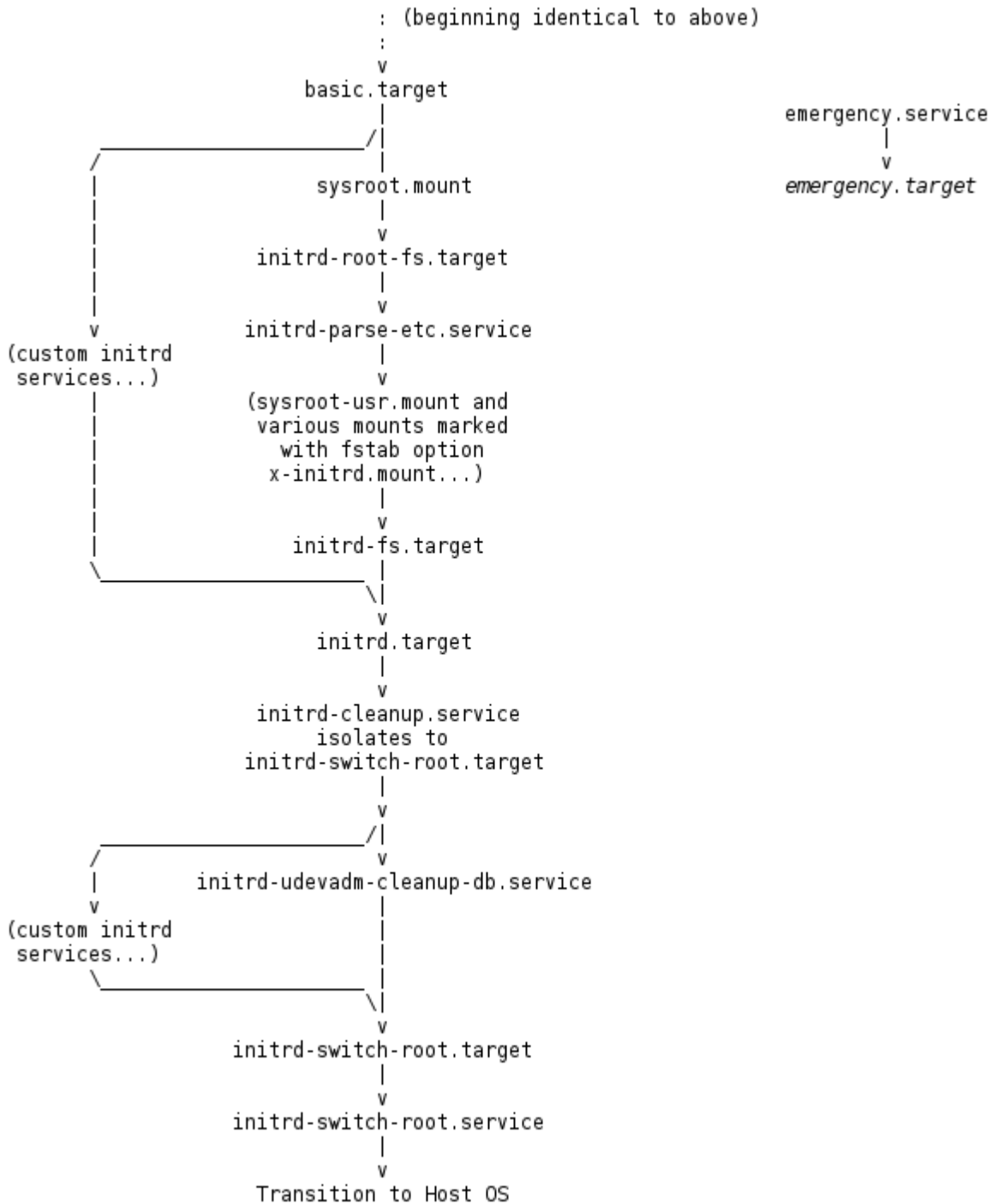
`timers.target` est poussé par `basic.target` de manière asynchrone. Cela permet aux unités timer de dépendre de services qui deviennent disponible plus tard.

## Démarrage dans l'initrd

L'implémentation du Ram disk initial peut être définis en utilisant `systemd`. Dans ce cas, le démarrage dans l'initrd suit la structure suivante.

La cible par défaut dans l'initrd est `initrd.target`. Le processus de boot commence de la même manière que le démarrage du gestionnaire système jusqu'à ce qu'il atteigne `basic.target`. De là, `systemd` approche la cible spéciale `initrd.target`. Si le périphérique root peut être monté dans `/sysroot`, l'unité `sysroot.mount` devient active et `initrd-root-fs.target` est atteinte. Le service `initrd-parse-etc.service` scanne `/sysroot/etc/fstab` pour un point de montage possible de `/usr` et les entrées additionnelles marquées avec l'option `x-initrd.mount`. Toutes les entrées trouvées sont montées sous `sysroot`, et `initrd-fs.target` est atteint. Le service `initrd-cleanup.service` isole vers `initrd-switch-root.target`, où les services de nettoyage peuvent être lancés. En dernier, `initrd-switch-root.service` est activé, causant le

système à switch sont root.



## Arrêt du système

L'extinction du système avec systemd consiste également de divers unités cible avec une structure minimale :

