



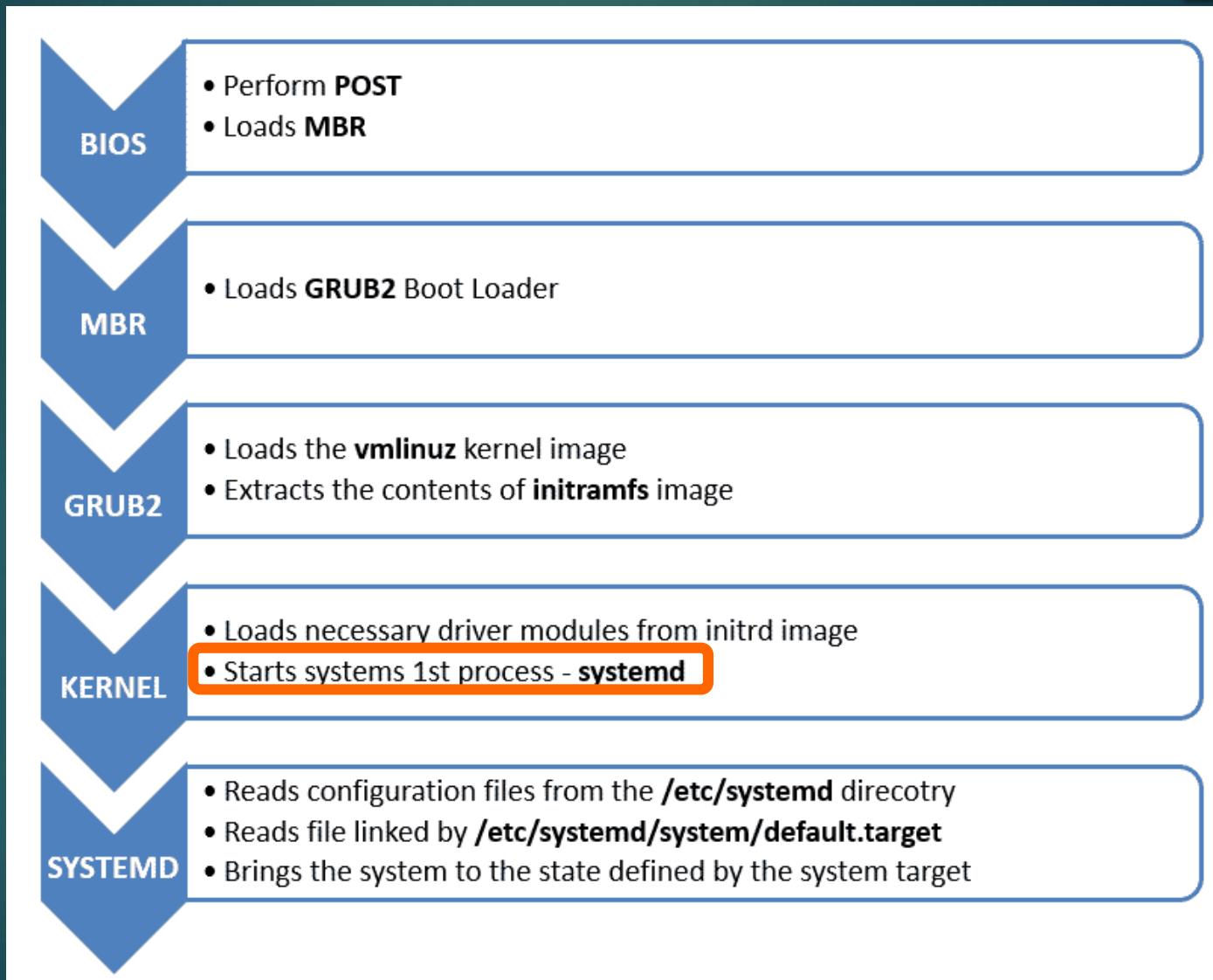
I106B

13. Processus

[HTTPS://WEB.ARCHIVE.ORG/WEB/20161027104203/HTTP://RYANSTUTORIALS.NET/LINUXTUTORIAL/PROCESSES.PHP](https://web.archive.org/web/20161027104203/http://ryanstutorials.net/linuxtutorial/processes.php)

Séquence de démarrage de Linux

2



Un processus

3

- ▶ Est un programme en mémoire en cours d'exécution.
 - ▶ PID : numéro unique identifiant un processus en mémoire
 - ▶ Au démarrage : un premier processus est démarré automatiquement, PID=1 (*init* ou *systemd*)
 - C'est ce processus qui lance ensuite les autres processus
 - Chaque processus possède donc un processus parent (celui qui l'a lancé) : PPID = PID du processus parent
- ➔ Arborescence de processus (avec pour racine: PID=1)

ps tree

4

- Affiche une structure en arbre pour les processus (options: -p pour afficher les PIDs ; -u pour afficher les propriétaires)

[→ nécessite une installation séparée:
`sudo apt-get install psmisc`]

```
anthony.legrand@courslinux:~$ pstree
systemd--VGAAuthService
        --2*[agetty]
        --apache2--2*[apache2--26*[{apache2}]]
        --cron
        --dbus-daemon
        --ntpd--{ntpd}
        --rsyslogd--{in:imklog}
                --{in:imuxsock}
                --{rs:main Q:Reg}
        --sshd--7*[sshd--sshd--bash]
                --2*[sshd--sshd--bash--nano]
                --sshd--sshd--bash--pstree
                --sshd--sshd--bash--ssh
```

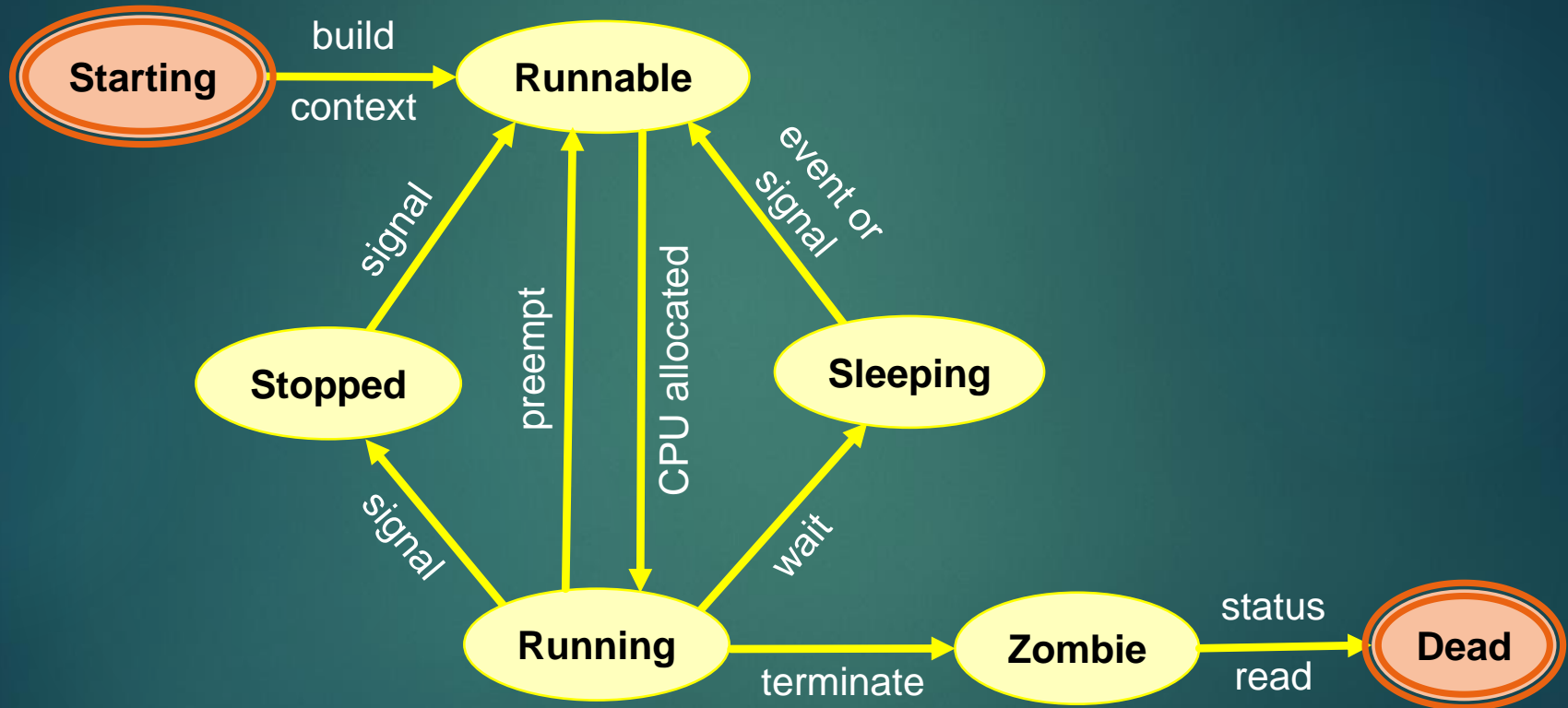
Commandes utiles

5

- ▶ **ps** : liste les processus
(option: `-u` pour afficher vos propres processus)
- ▶ **top** : « gestionnaire de tâches »
- ▶ **htop** (*human top*) : comme `top` en plus joli
[→ nécessite une installation séparée:
`sudo apt-get install htop`]

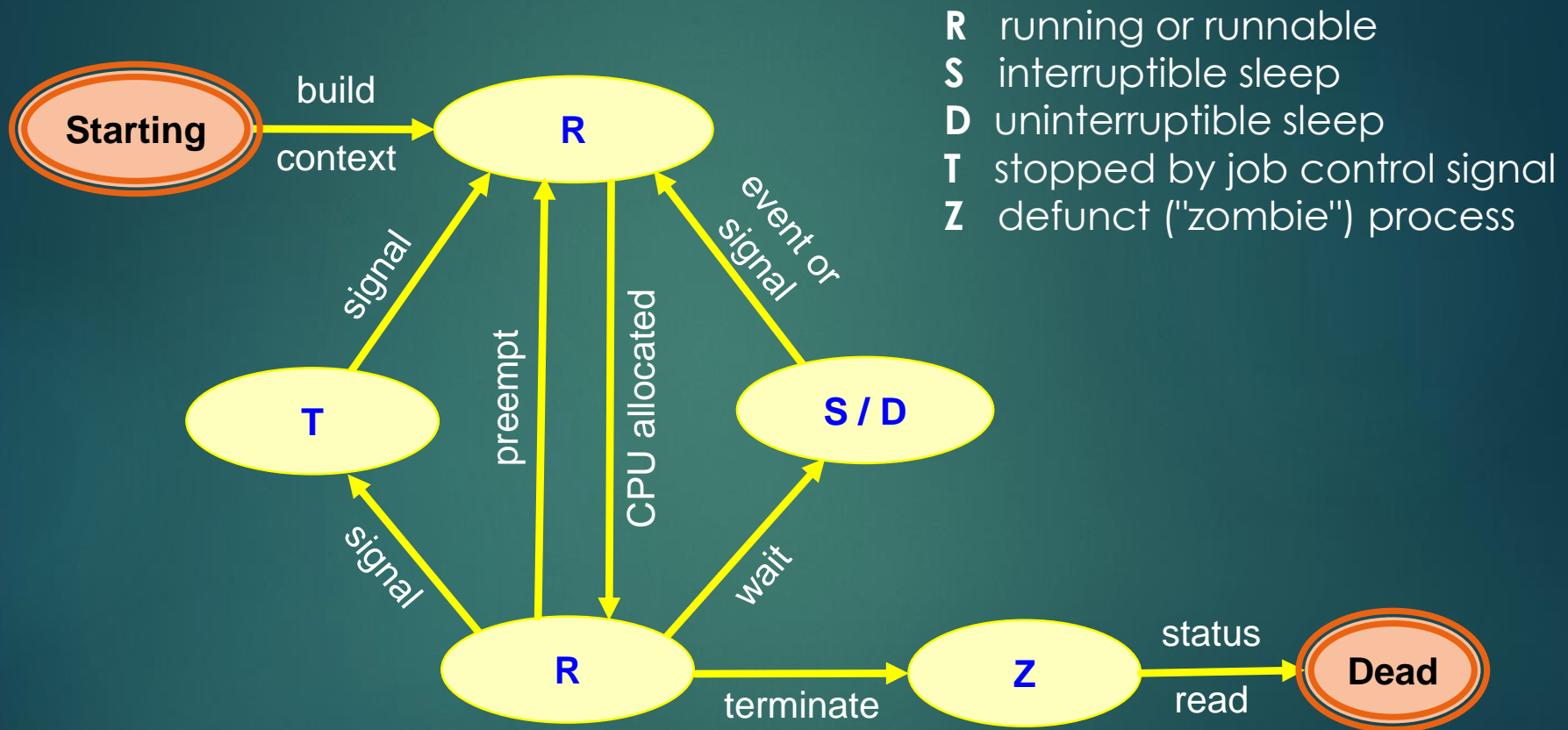
Etats d'un processus

6



Etats d'un processus

7



ps -l : la colonne S fournit l'état des processus (cf. man ps)

Signaux

8

kill -SIGNAL PID0

- ▶ Mécanisme permettant d'envoyer un byte à un processus.
 - Indépendant de `stdin`, `stdout` et `stderr`
 - La valeur du byte a une signification précise
(cf. **kill -l** pour afficher les signaux définis sur une machine)
- ▶ Utilité : terminer un processus
 - SIGTERM (15) : en lui demandant, afin qu'il puisse sauver ses fichiers par exemple
 - SIGINT (2) : équivalent à SIGTERM, généré quand on entre Ctrl+C au clavier
 - SIGKILL (9) : en lui forçant la main

- ▶ Une commande exécutée dans un terminal reçoit aussi un numéro de job.
 - %1 est le premier job actif, %2 le second, etc.
- ▶ Par défaut, un job lancé s'empare du terminal.
 - S'il lit sur `stdin`, il lit ce qui est entré sur le terminal.
 - Bloque le terminal tant qu'il s'exécute.
- ▶ On peut l'envoyer à l'arrière-plan.
 - Le terminal est relâché et on peut entrer de nouvelles commandes à lancer.
 - S'il lit sur `stdin`, il ne peut plus rien y lire.
 - S'il écrit sur `stdout`, cela continue de s'afficher sur le terminal, ce qui peut se mélanger avec l'affichage d'autres commandes !

Gestion d'un job

10

commande



**S'exécute en
avant-plan**

commande &

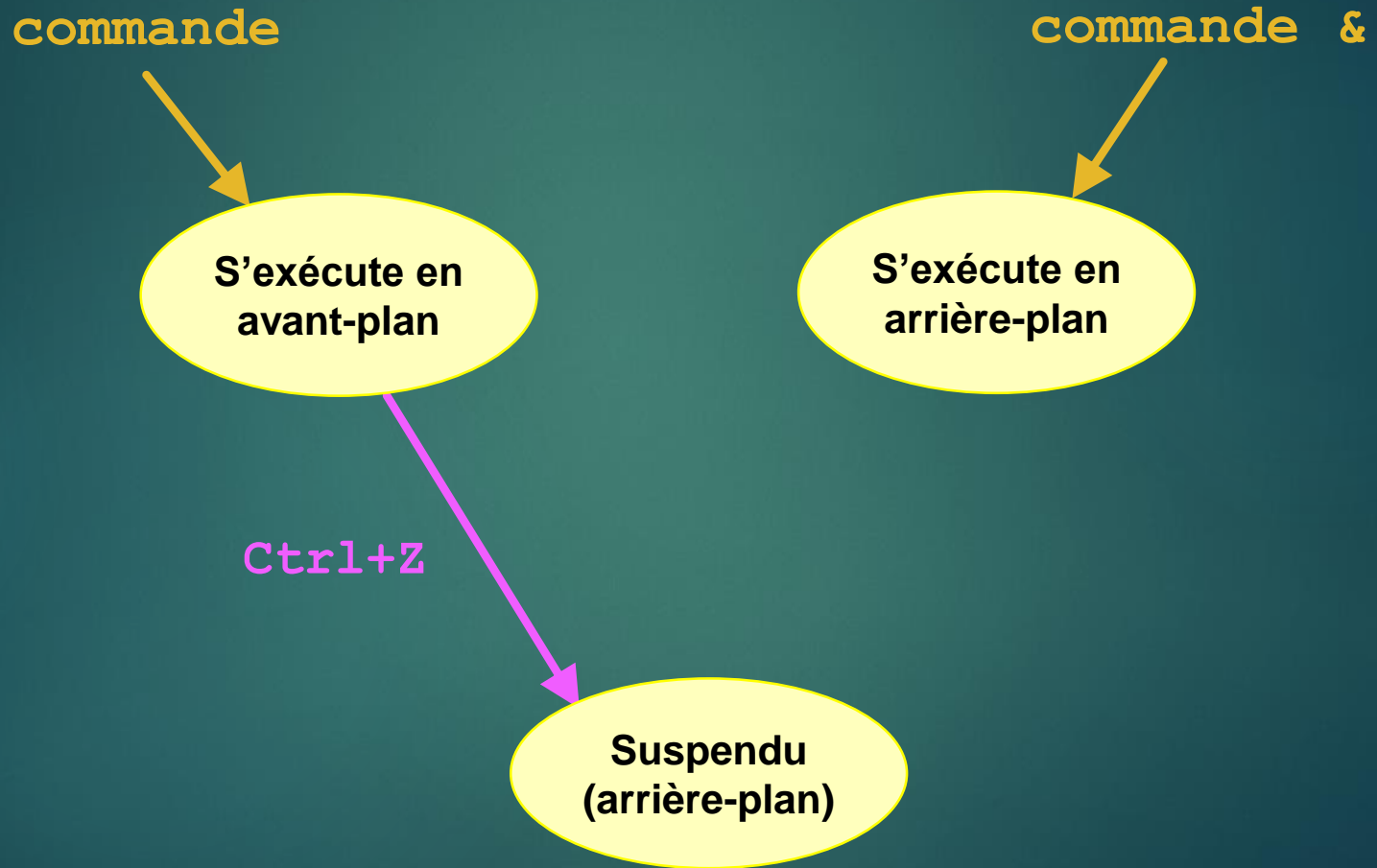


**S'exécute en
arrière-plan**

**Suspendu
(arrière-plan)**

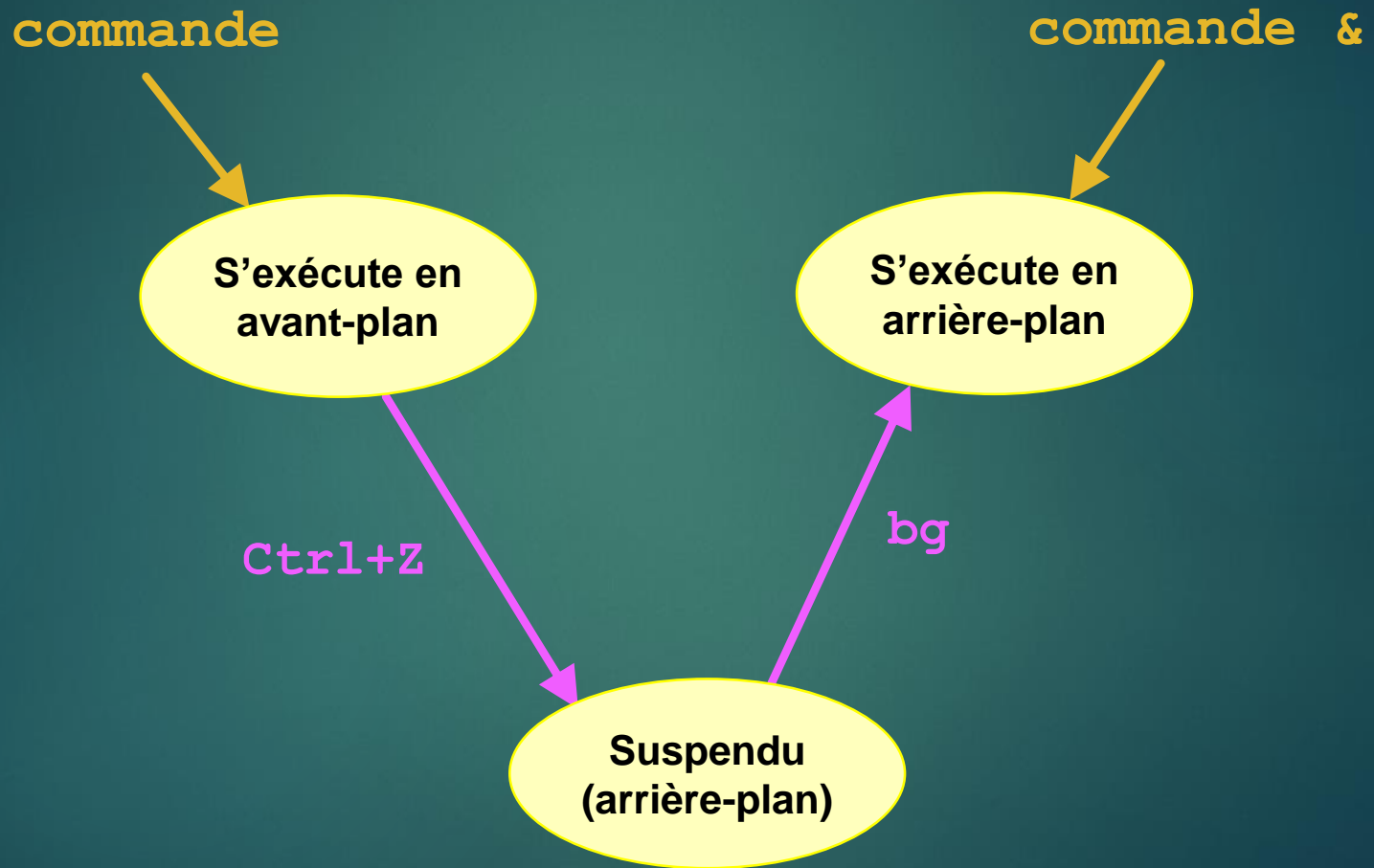
Gestion d'un job

10



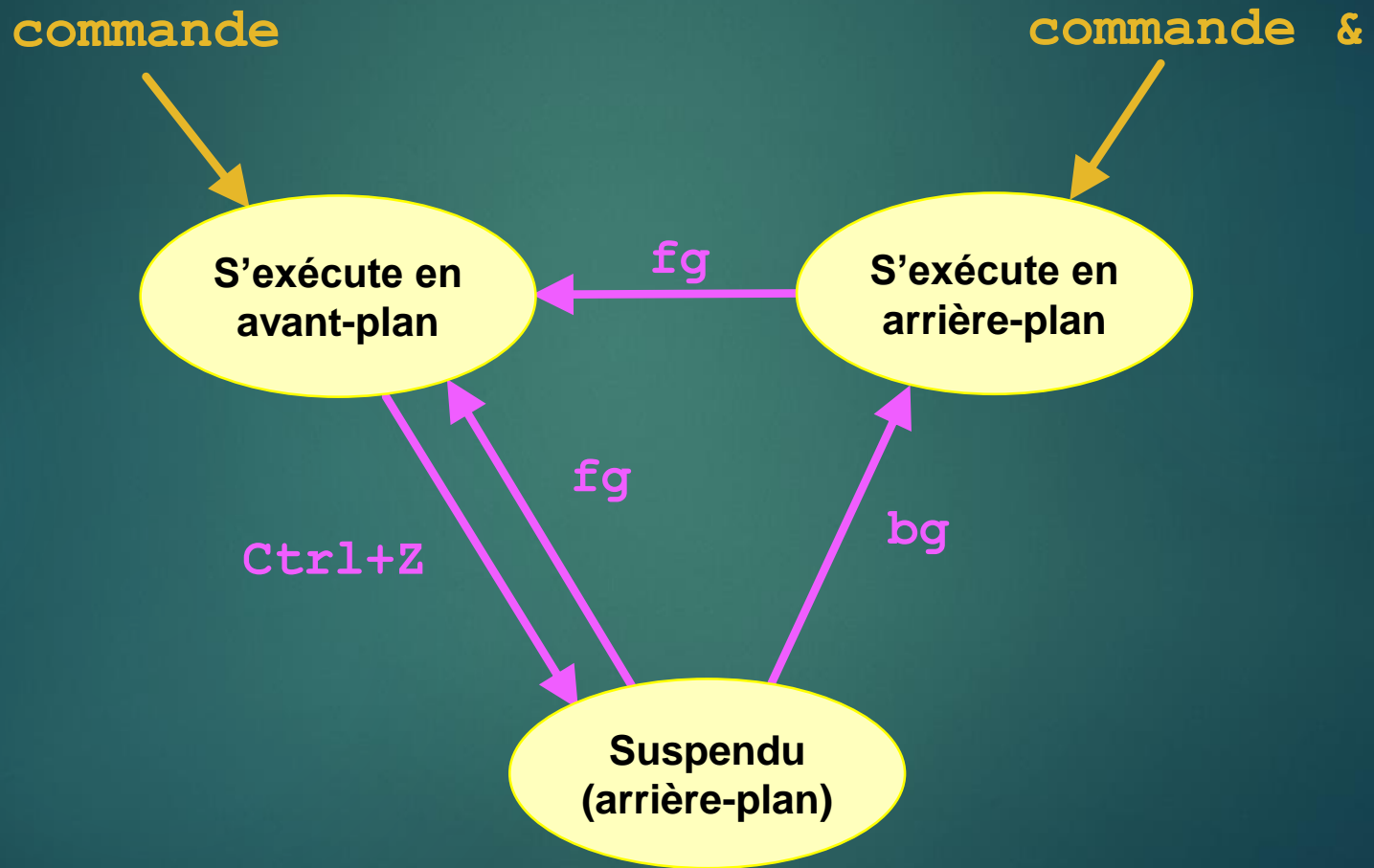
Gestion d'un job

10



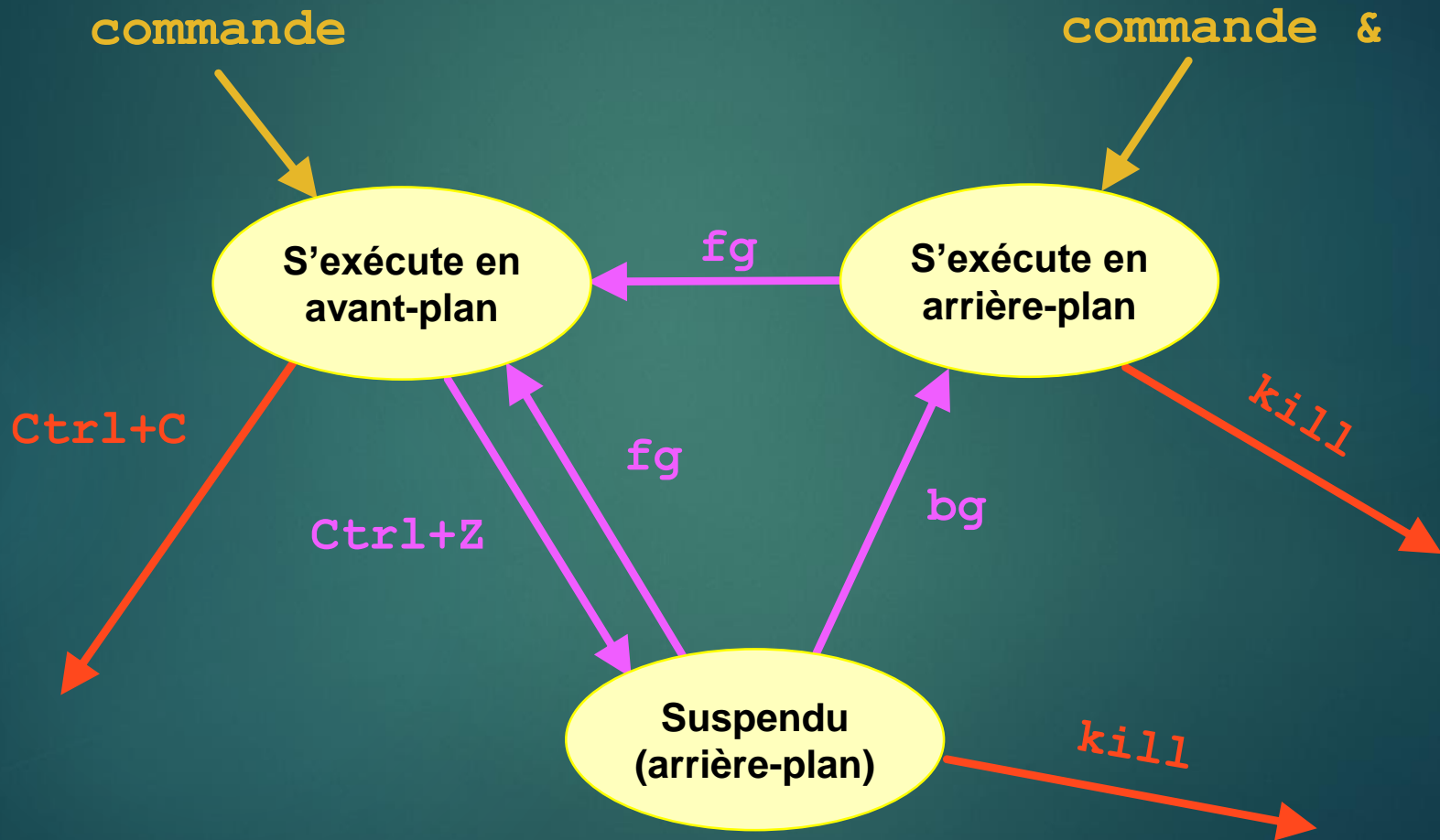
Gestion d'un job

10



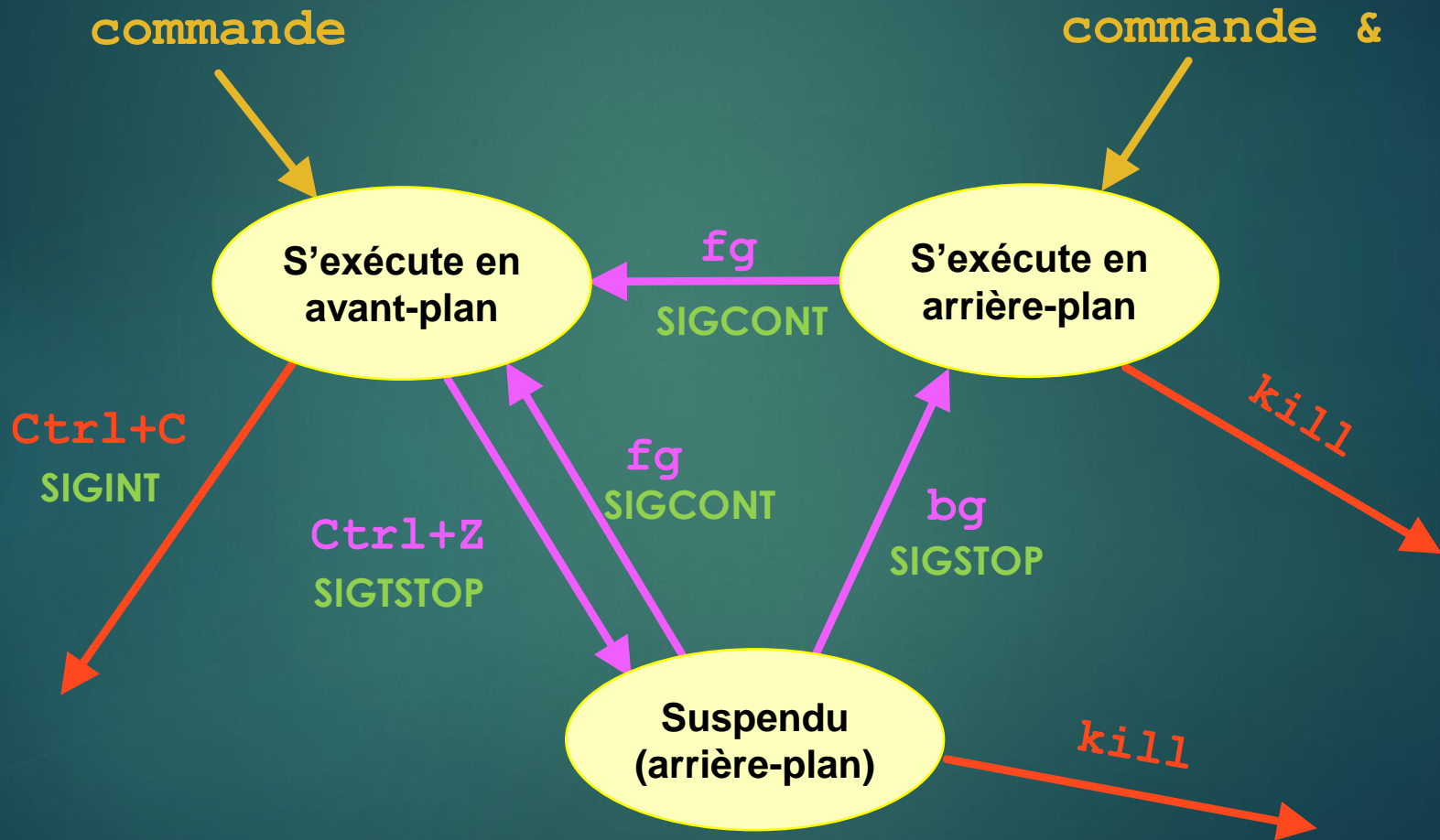
Gestion d'un job

10



Gestion d'un job

10



Commandes utiles

16

- ▶ **ps** : liste les processus
- ▶ **cmd &** : lance un processus en arrière-plan
- ▶ **fg** : remet un job à l'avant-plan
- ▶ **bg** : reprend l'exécution d'un job et l'envoie à l'arrière-plan (p.e. après Ctrl-Z)
- ▶ **jobs** : liste les processus en cours d'exécution dans le terminal où la commande est lancée
- ▶ **kill** : envoie un signal à un processus