

## C13 Processus

### Définition

Un **processus** est un programme en cours d'exécution.

### Remarques

## C13 Processus

### Définition

Un **processus** est un programme en cours d'exécution.

### Remarques

Ne pas confondre *processus* et *programme*. Un même programme, exécuté par différents utilisateurs et/ou à différents moments dans le temps correspond à des processus différents. On dit qu'un processus est associé à un **contexte d'exécution**.

## C13 Processus

### Définition

Un **processus** est un programme en cours d'exécution.

### Remarques

Ne pas confondre *processus* et *programme*. Un même programme, exécuté par différents utilisateurs et/ou à différents moments dans le temps correspond à des processus différents. On dit qu'un processus est associé à un **contexte d'exécution**.

Sur un système de type Linux, la commande `ps` permet de lister les processus actifs, la commande `htop` permet d'en avoir une vue dynamique. Ces commandes affichent notamment un numéro appelé `pid` (pour process identifier) qui identifie de façon unique chaque processus.

## C13 Processus

### Définition

Un **processus** est un programme en cours d'exécution.

### Remarques

Ne pas confondre *processus* et *programme*. Un même programme, exécuté par différents utilisateurs et/ou à différents moments dans le temps correspond à des processus différents. On dit qu'un processus est associé à un **contexte d'exécution**.

Sur un système de type Linux, la commande `ps` permet de lister les processus actifs, la commande `htop` permet d'en avoir une vue dynamique. Ces commandes affichent notamment un numéro appelé `pid` (pour process identifier) qui identifie de façon unique chaque processus.

Les ressources de l'ordinateur ne peuvent pas être utilisées de façon simultanée par les nombreux processus. Le temps d'utilisation de ces ressources est donc partagé en intervalle très courts et attribués aux processus par une partie spécifique de l'OS : **l'ordonnanceur**.

### Etats d'un processus

Les systèmes d'exploitation gèrent l'exécution « simultanée » de plusieurs processus en leur attribuant différents états :

### Etats d'un processus

Les systèmes d'exploitation gèrent l'exécution « simultanée » de plusieurs processus en leur attribuant différents états :

**élu** : le processus est en train de s'exécuter (il utilise le cpu).

### Etats d'un processus

Les systèmes d'exploitation gèrent l'exécution « simultanée » de plusieurs processus en leur attribuant différents états :

**élu** : le processus est en train de s'exécuter (il utilise le cpu).

**bloqué** : le processus était dans l'état élu, mais il attend une ressource qui n'est pas disponible.

### Etats d'un processus

Les systèmes d'exploitation gèrent l'exécution « simultanée » de plusieurs processus en leur attribuant différents états :

**élu** : le processus est en train de s'exécuter (il utilise le cpu).

**bloqué** : le processus était dans l'état élu, mais il attend une ressource qui n'est pas disponible.

**prêt** : le processus était bloqué mais il a obtenu la ressource qu'il attendait.



### Etats d'un processus

Les systèmes d'exploitation gèrent l'exécution « simultanée » de plusieurs processus en leur attribuant différents états :

**élu** : le processus est en train de s'exécuter (il utilise le cpu).

**bloqué** : le processus était dans l'état élu, mais il attend une ressource qui n'est pas disponible.

**prêt** : le processus était bloqué mais il a obtenu la ressource qu'il attendait.

**terminé** : le processus a terminé son exécution.

## Schéma : états des processus

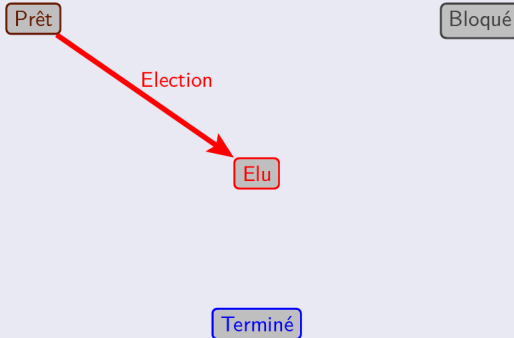
Prêt

Bloqué

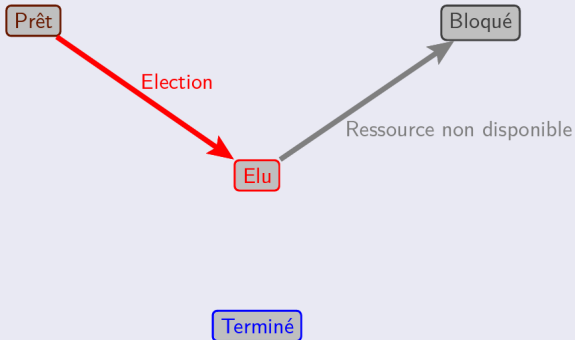
Exé

Terminé

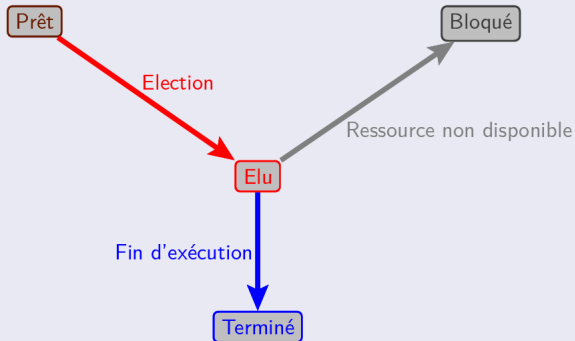
## Schéma : états des processus



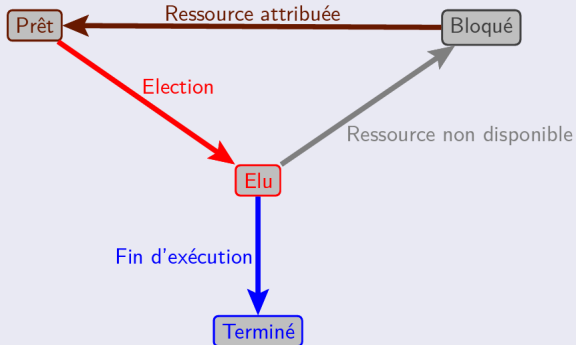
## Schéma : états des processus



## Schéma : états des processus



## Schéma : états des processus



## Quelques commandes

### Quelques commandes

Tout processus possède un numéro unique permettant de l'identifier : le **pid**.



### Quelques commandes

Tout processus possède un numéro unique permettant de l'identifier : le **pid**.  
Les processus (sauf le processus initial de pid 0) ont un aussi un **ppid** qui est le pid de leur processus parent.

### Quelques commandes

Tout processus possède un numéro unique permettant de l'identifier : le **pid**.  
Les processus (sauf le processus initial de pid 0) ont un aussi un **ppid** qui est le pid de leur processus parent.

Plusieurs commandes permettent de visualiser les processus :

### Quelques commandes

Tout processus possède un numéro unique permettant de l'identifier : le **pid**.  
Les processus (sauf le processus initial de pid 0) ont un aussi un **ppid** qui est le pid de leur processus parent.

Plusieurs commandes permettent de visualiser les processus :

**ps** pour avoir une vue instantanée des processus,

### Quelques commandes

Tout processus possède un numéro unique permettant de l'identifier : le **pid**.  
Les processus (sauf le processus initial de pid 0) ont un aussi un **ppid** qui est le pid de leur processus parent.

Plusieurs commandes permettent de visualiser les processus :

- ps** pour avoir une vue instantanée des processus,
- ps tree** pour avoir une vue hiérarchisée des processus,

### Quelques commandes

Tout processus possède un numéro unique permettant de l'identifier : le **pid**. Les processus (sauf le processus initial de pid 0) ont un aussi un **ppid** qui est le pid de leur processus parent.

Plusieurs commandes permettent de visualiser les processus :

- ps** pour avoir une vue instantanée des processus,
- ps tree** pour avoir une vue hiérarchisée des processus,
- top** pour avoir une vue dynamique des processus.

### Quelques commandes

Tout processus possède un numéro unique permettant de l'identifier : le **pid**. Les processus (sauf le processus initial de pid 0) ont un aussi un **ppid** qui est le pid de leur processus parent.

Plusieurs commandes permettent de visualiser les processus :

- ps** pour avoir une vue instantanée des processus,
- pstree** pour avoir une vue hiérarchisée des processus,
- top** pour avoir une vue dynamique des processus.

La commande **kill** permet de mettre fin à un processus.

### Définition

L' **ordonnanceur** (*scheduler* en anglais) est la partie du système d'exploitation qui choisit l'ordre d'exécution des processus. Parmi les algorithmes d'ordonnancement possibles, on peut citer :

### Remarques

## C13 Processus

### Définition

L' **ordonnanceur** (*scheduler* en anglais) est la partie du système d'exploitation qui choisit l'ordre d'exécution des processus. Parmi les algorithmes d'ordonnancement possibles, on peut citer :

le **tourniquet** : le temps cpu est affectée à chaque processus à tour de rôle

### Remarques



### Définition

L' **ordonnanceur** (*scheduler* en anglais) est la partie du système d'exploitation qui choisit l'ordre d'exécution des processus. Parmi les algorithmes d'ordonnancement possibles, on peut citer :

le **tourniquet** : le temps cpu est affectée à chaque processus à tour de rôle

le **premier entré, premier sorti**, (fifo, pour *first in first out*) : c'est le principe d'une file d'attente.

### Remarques

### Définition

L' **ordonnanceur** (*scheduler* en anglais) est la partie du système d'exploitation qui choisit l'ordre d'exécution des processus. Parmi les algorithmes d'ordonnancement possibles, on peut citer :

le **tourniquet** : le temps cpu est affectée à chaque processus à tour de rôle

le **premier entré, premier sorti**, (fifo, pour *first in first out*) : c'est le principe d'une file d'attente.

le **plus court d'abord** : il faut évaluer le temps d'execution des processus.

### Remarques

### Définition

L' **ordonnanceur** (*scheduler* en anglais) est la partie du système d'exploitation qui choisit l'ordre d'exécution des processus. Parmi les algorithmes d'ordonnancement possibles, on peut citer :

le **tourniquet** : le temps cpu est affectée à chaque processus à tour de rôle

le **premier entré, premier sorti**, (fifo, pour *first in first out*) : c'est le principe d'une file d'attente.

le **plus court d'abord** : il faut évaluer le temps d'exécution des processus.

La mise en place d'un **ordre de priorité** entre les processus.

### Remarques

### Définition

L' **ordonnanceur** (*scheduler* en anglais) est la partie du système d'exploitation qui choisit l'ordre d'exécution des processus. Parmi les algorithmes d'ordonnancement possibles, on peut citer :

le **tourniquet** : le temps cpu est affectée à chaque processus à tour de rôle

le **premier entré, premier sorti**, (fifo, pour *first in first out*) : c'est le principe d'une file d'attente.

le **plus court d'abord** : il faut évaluer le temps d'execution des processus.

La mise en place d'un **ordre de priorité** entre les processus.

### Remarques

L'exemple typique de la méthode fifo, est la gestion de la file d'attente d'une imprimante.

## C13 Processus

### Définition

L' **ordonnanceur** (*scheduler* en anglais) est la partie du système d'exploitation qui choisit l'ordre d'exécution des processus. Parmi les algorithmes d'ordonnancement possibles, on peut citer :

le **tourniquet** : le temps cpu est affectée à chaque processus à tour de rôle

le **premier entré, premier sorti**, (fifo, pour *first in first out*) : c'est le principe d'une file d'attente.

le **plus court d'abord** : il faut évaluer le temps d'execution des processus.

La mise en place d'un **ordre de priorité** entre les processus.

### Remarques

L'exemple typique de la méthode fifo, est la gestion de la file d'attente d'une imprimante.

Sur un système de type Linux, la commande **nice** permet d'attribuer un ordre de priorité à un processus de -20 (le plus prioritaire) à 19 (le moins prioritaire).

### Ressources

Les ressources peuvent être attribuées aux processus de façon :

### Ressources

Les ressources peuvent être attribuées aux processus de façon :

**partagée** c'est à dire que la ressource peut être utilisée de façon simultanée par plusieurs processus.

### Ressources

Les ressources peuvent être attribuées aux processus de façon :

**partagée** c'est à dire que la ressource peut être utilisée de façon simultanée par plusieurs processus. Par exemple, un fichier dans lequel on souhaite seulement lire des informations est une ressource partagée.



### Ressources

Les ressources peuvent être attribuées aux processus de façon :

**partagée** c'est à dire que la ressource peut être utilisée de façon simultanée par plusieurs processus. Par exemple, un fichier dans lequel on souhaite seulement lire des informations est une ressource partagée.

**exclusive** c'est à dire que la ressource est verrouillée et attribuée à un unique processus.

### Ressources

Les ressources peuvent être attribuées aux processus de façon :

**partagée** c'est à dire que la ressource peut être utilisé de façon simultanée par plusieurs processus. Par exemple, un fichier dans lequel on souhaite seulement lire des informations est une ressource partagée.

**exclusive** c'est à dire que la ressource est verrouillée et attribuée à un unique processus. Par exemple, une imprimante est une ressource attribuée de façon exclusive.

### Ressources

Les ressources peuvent être attribuées aux processus de façon :

**partagée** c'est à dire que la ressource peut être utilisé de façon simultanée par plusieurs processus. Par exemple, un fichier dans lequel on souhaite seulement lire des informations est une ressource partagée.

**exclusive** c'est à dire que la ressource est verrouillée et attribuée à un unique processus. Par exemple, une imprimante est une ressource attribuée de façon exclusive.

### Interblocage

On se trouve dans une situation d'**interblocage** (en anglais *deadlock*), lorsque des ressources exclusives requises par des processus sont utilisées par d'autres qui attendent eux mêmes des ressources utilisées par les premiers.

## Illustration d'une situation d'interblocage

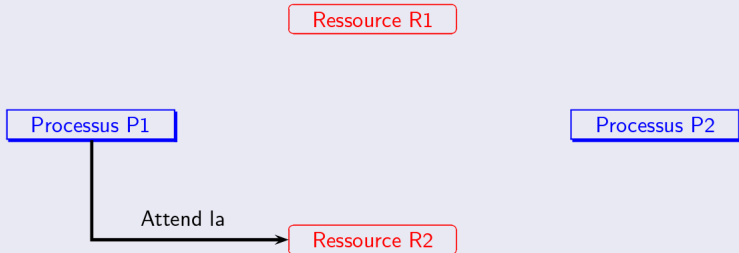
Processus P1

Ressource R1

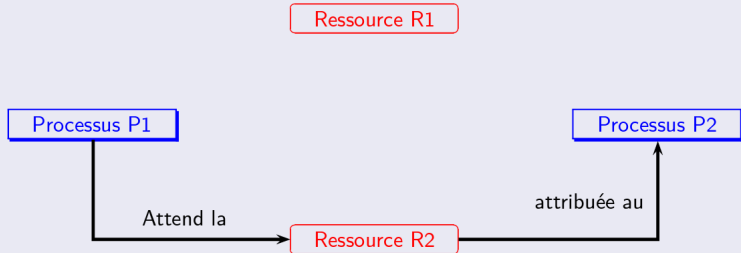
Processus P2

Ressource R2

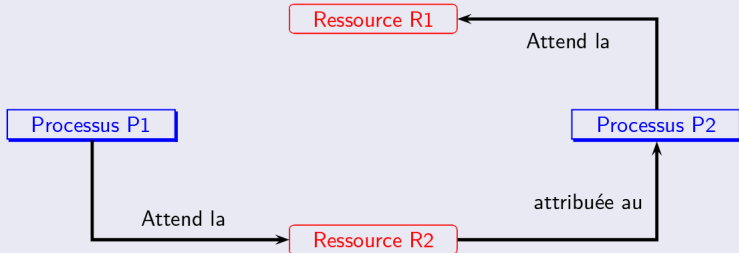
## Illustration d'une situation d'interblocage



## Illustration d'une situation d'interblocage



## Illustration d'une situation d'interblocage



## Illustration d'une situation d'interblocage

