

# TD2 – Processus UNIX

## Questions de cours

1. Dans la figure ci-dessous, trois états de processus apparaissent. En théorie, avec trois états, on pourrait avoir six transitions, deux en sortie de chaque état. Cependant, on ne voit que quatre transitions. Existe-t-il des circonstances dans lesquelles l'une ou l'autre des transitions manquantes pourraient se produire ?

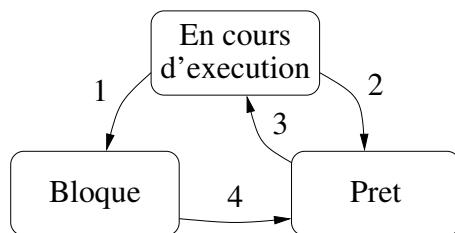


FIGURE 1 – Les états d'un processus

- (a) Le processus est bloqué en attente d'une donnée.
- (b) L'ordonnanceur choisit un autre processus.
- (c) L'ordonnanceur choisit ce processus.
- (d) La donnée devient disponible.

# Programmation système

## Exercice 1 :

Écrivez un programme générant un processus fils avec la primitive système `fork`.

- le processus fils doit afficher son numéro (*pid*) ainsi que le numéro du père à l'aide des primitives système `getpid` et `getppid`, puis sort (primitive `exit`) avec un code de retour égal au dernier chiffre du *pid*.
- le processus père, quant à lui, affiche le *pid* du fils, puis attend sa terminaison (primitive `wait`) et affiche son code de retour.

## Exercice 2 :

Écrivez un programme qui lance *n* processus fils dans une première étape puis, dans une deuxième étape, attend leur terminaison à l'aide de la primitive `wait`. À chaque fois qu'un processus se termine, le père affiche son numéro (*pid*) et son code de retour.

## Exercice 3 :

Écrivez un programme ayant la syntaxe suivante :

`matproc n m`

L'action de ce programme doit être de générer *n* processus, chacun d'entre-eux devant générer *n* processus à son tour, et ainsi de suite jusqu'à *m* niveaux.

Combien de processus sont générés au total ?

## Exercice 4 :

Écrivez un programme qui :

1. lance la commande `ls` sur un répertoire passé en paramètre,
2. redirige la sortie standard de `ls` sur `/dev/null`,
3. affiche le temps pris par la commande `ls` (primitive système `times`).