ISS - COURS 8 : Les waits

Dans ces schémas on distingue :
- les fonctions du C qui correspondent à l'implémentation du Bash et sont notées avec des parenthèses (ici : *fork(), exec(), waitpid(), wait(), exit()*)
- les commandes du Bash (ici: wait)

Dans la plupart des langages il existe des fonctions "waits" (reposant sur l'appel système ***sys_wait4()***) qui permettent d'attendre la fin d'exécution d'un processus :
- *la fonction C waitpid() :* permet d'attendre la fin du processus dont le pid est passé en paramètre, dans notre cas celui du processus de la dernière commande
- *la fonction C wait() :* permet d'attendre la fin d'un des fils du processus
- la commande Bash **`wait`** : permet d'attendre la fin de tous les processus fils

À la fin de son exécution, un processus rentre dans l'état zombie :
- il conserve une partie de ses données pour que son père puisse les récupérer
- il ne peut plus être élu (il n'a d'ailleurs plus de code à exécuter)
- il attend que son père le détruise avec une des fonctions "wait"

Si un processus père s'arrête avant son fils , il n'y aurait plus personne pour faire un wait sur l'orphelin qui resterait alors éternellement zombie :
- en s'arrêtant un père supprime tous ses fils zombies pour lesquels il n'a pas fait de wait
- en s'arrêtant un père fait adopter par le processus init (pid=1) tous ses fils encore vivants (Il n'y a donc jamais d'orphelin)
- c'est le processus init qui détruira les processus adoptés une fois zombie grâce à une boucle : while(true){wait()}