**Compte-rendu TP3 – DEBARD Jean et POIRIER Vincent**

**Exercice 1 : un peu de généalogie**

* Sortie obtenue :

Mon PID est 3317 et i=0

Mon PID est 3317 et i=1

Mon PID est 3317 et i=2

Mon PID est 3320 et i=2

Le processus 3320 a fini de s'executer

Mon PID est 3319 et i=1

Mon PID est 3318 et i=0

Mon PID est 3319 et i=2

Mon PID est 3318 et i=1

Mon PID est 3318 et i=2

Mon PID est 3322 et i=1

Mon PID est 3322 et i=2

Mon PID est 3321 et i=2

Le processus 3321 a fini de s'executer

Le processus 3319 a fini de s'executer

Mon PID est 3323 et i=2

Le processus 3323 a fini de s'executer

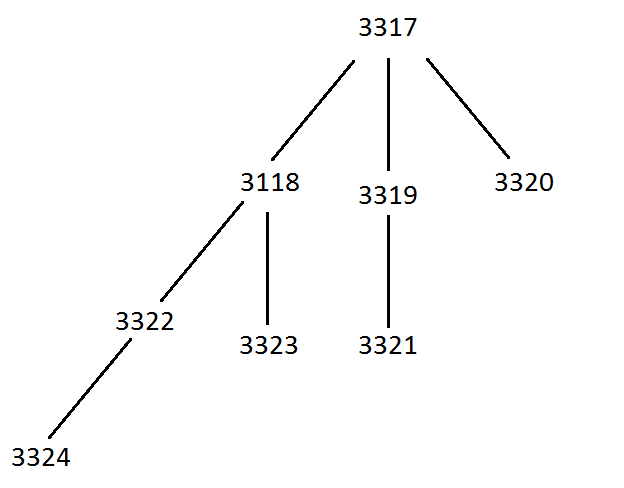
Mon PID est 3324 et i=2

Le processus 3324 a fini de s'executer

Le processus 3322 a fini de s'executer

Le processus 3318 a fini de s'executer

8 processus : 3317 – 3320 – 3319 – 3318 – 3322 – 3321 – 3323 – 3324

Généalogie :

## Exercice 2 : écriture d'un shell basique

#include "ligne\_commande.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

int main(int argc, char\*\* argv) {

pid\_t pid;

int etat;

while(1) {

printf("> ");

fflush(stdout);

char\*\* commande = lis\_ligne();

if(fin\_de\_fichier(commande)) {

printf("Fin du programme...\n");

return 0;

}

else {

if(!ligne\_vide(commande)) {

if(!strcmp(commande[0], "exit")) {

printf("Fin du programme...\n");

return 0;

}

pid = fork();

if(pid == 0) {

if(execvp(commande[0], commande) == -1) {

printf("Erreur, commande incorrecte.\n");

}

exit(0);

}

else {

wait(&etat);

}

}

}

}

}

## Exercice 3 : gestion de l'environnement dans le shell