

CR CTP PRS PIERRE LENGLE

On utilise CP pour renommer :

```
pierre@pierre:~/Bureau/PRS_LENGLE_CTP$ cp -r PRS_CTP_Codes PRS_CTP_Codes_LENGLE
pierre@pierre:~/Bureau/PRS_LENGLE_CTP$ ll
total 20
drwxr-xr-x  4 pierre pierre 4096 nov.  26 14:40 ./
drwxr-xr-x 22 pierre pierre 4096 nov.  26 14:39 ../
drwxr-xr-x  2 pierre pierre 4096 nov.  26 01:46 PRS_CTP_Codes/
drwxr-xr-x  2 pierre pierre 4096 nov.  26 14:40 PRS_CTP_Codes_LENGLE/
-rw-r--r--  1 pierre pierre 2672 nov.  26 01:47 PRS_CTP_Codes.tar.gz
pierre@pierre:~/Bureau/PRS_LENGLE_CTP$
```

On utilise gcc karaOK_v0.c -o karaOK

```
pierre@pierre:~/Bureau/PRS_LENGLE_CTP/PRS_CTP_Codes_LENGLE$ gcc -Wall karaOK_v0.c -o karaOK
karaOK_v0.c: In function 'main':
karaOK_v0.c:94:9: warning: implicit declaration of function 'CHECK' [-Wimplicit-function-declaration]
CHECK(pidFils[i] = fork(), "fork()");
^
karaOK_v0.c:114:5: warning: implicit declaration of function 'setTerm'; did you mean 'setenv'? [-Wimplicit-function-declaration]
setTerm(YELLOW);
^
setenv
karaOK_v0.c:114:13: error: 'YELLOW' undeclared (first use in this function); did you mean 'ELOOP'?
setTerm(YELLOW);
^
ELOOP
karaOK_v0.c:114:13: note: each undeclared identifier is reported only once for each function it appears in
karaOK_v0.c:118:5: warning: implicit declaration of function 'resetTerm' [-Wimplicit-function-declaration]
resetTerm();
^
karaOK_v0.c: In function 'welcome':
karaOK_v0.c:169:5: warning: implicit declaration of function 'clearScreen'; did you mean 'clearerr'? [-Wimplicit-function-declaration]
clearScreen();
^
clearerr
karaOK_v0.c:170:13: error: 'YELLOW' undeclared (first use in this function); did you mean 'ELOOP'?
setTerm(YELLOW);
^
ELOOP
```

Pour résoudre ce problème on ajoute la ligne suivante au fichier karaOK.c :

```
#include "common.h"
```

on recompile et on observe aucune erreur de compilation.

Question 4 :

On exécute le programme et on obtient le résultat suivant :

```
pierre@pierre:~/Bureau/PRS_LENGLE_CTP/PRS_CTP_Codes_LENGLE$ ./karaOK

+-----+
|  Je suis le processus  4942, fils de  4106  |
+-----+

Fils 1 :      lancement effectif du processus 4943, fils de 4942

  PID  PPID  SIZE  CLS  COMMAND          CMD
  4106  4097  2484  TS  bash             bash
  4942  4106   308  TS  karaOK           ./karaOK
  4943  4942   308  TS  karaOK           ./karaOK
  4944  4942   316  TS  sh               sh -c echo ;ps -o pid,ppid,size,cls,comm,cmd
  4945  4944  1012  TS  ps              ps -o pid,ppid,size,cls,comm,cmd
      Fils 1 :      fin du processus 4943

F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  4106  4097  0  80   0 -  6198 wait  pts/0        00:00:02 bash
0 S  1000  4942  4106  0  80   0 -  1128 wait  pts/0        00:00:00 karaOK
1 Z  1000  4943  4942  0  80   0 -    0 -      pts/0        00:00:00 karaOK <defunct>
0 S  1000  4947  4942  0  80   0 -  1157 wait  pts/0        00:00:00 sh
0 R  1000  4948  4947  0  80   0 -  7558 -      pts/0        00:00:00 ps

Fin du programme karaOK de pid = 4942
```

le fonction atoi() converti un string en un INT.
 La page manuel correspondante est la 3.
 il faut inclure stdlib.h avec #include<stdlib.h>
 sur ma machine il n'existe aucune entre manuel pour amoi() et alui().

Question 5 :

l'appel vas effectuer un appel systeme pour afficher la commande ps
 cette commande est suivi de -o qui permet des definir le format d'affichage de l'appel.
 On affichera donc le pid, le ppid, size , cls, comm et cmd.

Question 6 :

```
1 Z 1000 4943 4942 0 80 0 - 0 - pts/0 00:00:00 karaOK <defunct>
```

le processus est noté Z, on observe la mention <defunct>

dans le man on utilise /zombie pour cherche le mot cle zombie.

Processes marked <defunct> are dead processes (so-called "zombies") that remain because their parent has not destroyed them properly. These processes will be destroyed by init(8) if the parent process exits.

Question 8 :

```
pierre@pierre:~/Bureau/PRS LENGLE_CTP/PRS_CTP_Codes LENGLE$ ./karaOK 2> /home/pierre/cache
```

```
+-----+
| Je suis le processus 5931, fils de 4106 |
+-----+
```

PID	PPID	SIZE	CLS	COMMAND	CMD
4106	4097	3100	TS	bash	bash
5931	4106	308	TS	karaOK	./karaOK
5932	5931	308	TS	karaOK	./karaOK
5933	5931	316	TS	sh	sh -c echo ;ps -o pid,ppid,size,cls,comm,cmd
5934	5933	1012	TS	ps	ps -o pid,ppid,size,cls,comm,cmd

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	4106	4097	0	80	0	-	6352	wait	pts/0	00:00:02	bash
0	S	1000	5931	4106	0	80	0	-	1128	wait	pts/0	00:00:00	karaOK
1	Z	1000	5932	5931	0	80	0	-	0	-	pts/0	00:00:00	karaOK <defunct>
0	S	1000	5936	5931	0	80	0	-	1157	wait	pts/0	00:00:00	sh
0	R	1000	5937	5936	0	80	0	-	7558	-	pts/0	00:00:00	ps

Fin du programme karaOK de pid = 5931

```
pierre@pierre:~/Bureau/PRS LENGLE_CTP/PRS_CTP_Codes LENGLE$
```

On redirige les erreurs avec 2> (j'ai créer un fichier cache car je ne me rappel plus ou l'on doit normalement rediriger les erreurs)

Question 9 :

```
pierre@pierre:~/Bureau/PRS LENGLE_CTP/PRS_CTP_Codes LENGLE$ cp karaOK_v0.c karaOK_v1.c
pierre@pierre:~/Bureau/PRS LENGLE_CTP/PRS_CTP_Codes LENGLE$ ll
total 52
drwxr-xr-x 2 pierre pierre 4096 nov. 26 15:16 ./
drwxr-xr-x 3 pierre pierre 4096 nov. 26 14:53 ../
-rwxr-x--- 1 pierre pierre 232 nov. 26 14:40 ._common.h*
-rwxr-x--- 1 pierre pierre 1572 nov. 26 14:40 common.h*
-rwxr-xr-x 1 pierre pierre 13496 nov. 26 14:46 karaOK*
-rwxr-x--- 1 pierre pierre 227 nov. 26 14:40 ._karaOK_v0.c*
-rwxr-x--- 1 pierre pierre 6661 nov. 26 15:03 karaOK_v0.c*
-rwxr-x--- 1 pierre pierre 6661 nov. 26 15:16 karaOK_v1.c*
pierre@pierre:~/Bureau/PRS LENGLE_CTP/PRS_CTP_Codes LENGLE$
```

Question 10 :

```
+-----+
| Je suis le processus 6471, fils de 4106 |
+-----+

Fils 1 :      lancement effectif du processus 6472, fils de 6471
Fils 1 :      fin du processus 6472
Fils 1 :      fin du processus de pid : 6472

F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  4106  4097  0  80   0 - 6352 wait  pts/0      00:00:03 bash
0 S  1000  6471  4106  0  80   0 - 1128 wait  pts/0      00:00:00 karaOKv1
0 S  1000  6473  6471  0  80   0 - 1157 wait  pts/0      00:00:00 sh
0 R  1000  6474  6473  0  80   0 - 7558 -      pts/0      00:00:00 ps

Fin du programme karaOKv1 de pid = 6471
```

On observe l'absence de procesus zombie.

Question 11 :

On obtient le resultat suivant :

```
pierre@pierre:~/Bureau/PRS LENGLE_CTP/PRS_CTP_Codes LENGLE$ ./karaOKv1 4 2> /home/pierre/cache

+-----+
| Je suis le processus 6745, fils de 4106 |
+-----+
"F      "Mai,, cc'eesstt flaac im  "Doloie, t ei  "Rl+++e+ + c ,dhd o'arsnua tnyie olrtn o"
aud teb"
osno ldeoisl"
d'or"

F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  4106  4097  0  80   0 - 6352 wait  pts/0      00:00:03 bash
0 S  1000  6745  4106  0  80   0 - 1128 wait  pts/0      00:00:00 karaOKv1
0 S  1000  6750  6745  0  80   0 - 1157 wait  pts/0      00:00:00 sh
0 R  1000  6751  6750  0  80   0 - 7558 -      pts/0      00:00:00 ps

Fin du programme karaOKv1 de pid = 6745

pierre@pierre:~/Bureau/PRS LENGLE_CTP/PRS_CTP_Codes LENGLE$
```

on observe un melange de couleur et de caracteres, cela est du au fait que les processus partage l'affichage et donc chacun affiche des qu'il le peux ce qui créer une anarchie dans l'affichage

```

+-----+
|   Je suis le processus  7840, fils de  4106   |
+-----+
"Do, le dos il a bon dos"
"Ré, rayon de soleil d'or"

F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  4106  4097  0  80   0 -  6390 wait  pts/0      00:00:03 bash
0 S  1000  7840  4106  0  80   0 -  1674 wait  pts/0      00:00:00 karaOKv2
0 S  1000  7843  7840  0  80   0 -  1157 wait  pts/0      00:00:00 sh
0 R  1000  7844  7843  0  80   0 -  7558 -      pts/0      00:00:00 ps

Fin du programme karaOKv2 de pid = 7840

```

on utilise des semaphore nomme et on remarque le bon affichage.

Question 13

afin de compiler on ajouter l'option -pthread, cette option est obligatoire pour compiler l'utilisation de semaphore

Avec 4 fils on obtient le resultat suivant :

```

+-----+
|   Je suis le processus  8019, fils de  4106   |
+-----+
"Do, le dos il a bon dos"
"Ré, rayon de soleil d'or"
"Mi, c'est la moitié d'un tout"
"Fa, c'est facile à chanter"

F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  4106  4097  0  80   0 -  6390 wait  pts/0      00:00:03 bash
0 S  1000  8019  4106  0  80   0 -  1674 wait  pts/0      00:00:00 karaOKv2
0 S  1000  8025  8019  0  80   0 -  1157 wait  pts/0      00:00:00 sh
0 R  1000  8026  8025  0  80   0 -  7558 -      pts/0      00:00:00 ps

Fin du programme karaOKv2 de pid = 8019

```

```

+-----+
|   Je suis le processus  8066, fils de  4106   |
+-----+
"Fa, c'est facile à chanter"
"Do, le dos il a bon dos"
"Ré, rayon de soleil d'or"
"Mi, c'est la moitié d'un tout"

F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  4106  4097  0  80   0 -  6390 wait  pts/0      00:00:03 bash
0 S  1000  8066  4106  0  80   0 -  1674 wait  pts/0      00:00:00 karaOKv2
0 S  1000  8072  8066  0  80   0 -  1157 wait  pts/0      00:00:00 sh
0 R  1000  8073  8072  0  80   0 -  7558 -      pts/0      00:00:00 ps

Fin du programme karaOKv2 de pid = 8066

pierre@pierre:~/Bureau/PRS LENGLE_CTP/PRS_CTP_Codes LENGLE$

```

on observe un résultat différent. Cette difference est du lancement des programme fils qui se commence par un sleep de duree aleatoire.

Question 14 :

La ressource critique est créée dans le dossier `/dev/shm`.

Si le fichier n'est pas supprimé à la fin de l'exécution par le programme, on obtient une erreur de segmentation lors du nouveau lancement du programme. D'où l'importance du `unlink`.

Question 15 :

Avant la création des processus fils, on crée 7 sémaphores nommées de valeur initiale 0, et le premier sémaphore a 1. Puis lors de l'exécution d'un processus fils N , on incrémente le sémaphore $N+1$.

Question 16

```
+-----+
| Je suis le processus 8543, fils de 4106 |
+-----+
"Do, le dos il a bon dos"
"Ré, rayon de soleil d'or"
"Mi, c'est la moitié d'un tout"
"Fa, c'est facile à chanter"
"Sol, la terre où vous marchez"
"La, l'endroit où nous allons"
"Si, siffler comme un pinson"

F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY      TIME CMD
0 S  1000  4106  4097  0  80   0  -  6390 wait pts/0    00:00:04 bash
0 S  1000  8543  4106  0  80   0  -  1675 wait pts/0    00:00:00 kara0Kv3
0 S  1000  8553  8543  0  80   0  -  1157 wait pts/0    00:00:00 sh
0 R  1000  8554  8553  0  80   0  -  7558 -    pts/0    00:00:00 ps

Fin du programme kara0Kv3 de pid = 8543

pierre@pierre:~/Bureau/PBS_L'ENGLE_CTP/PBS_CTP_Codes_L'ENGLES
```

Après plusieurs lancements du programme, on observe toujours la même sortie avec le bon ordre des messages.