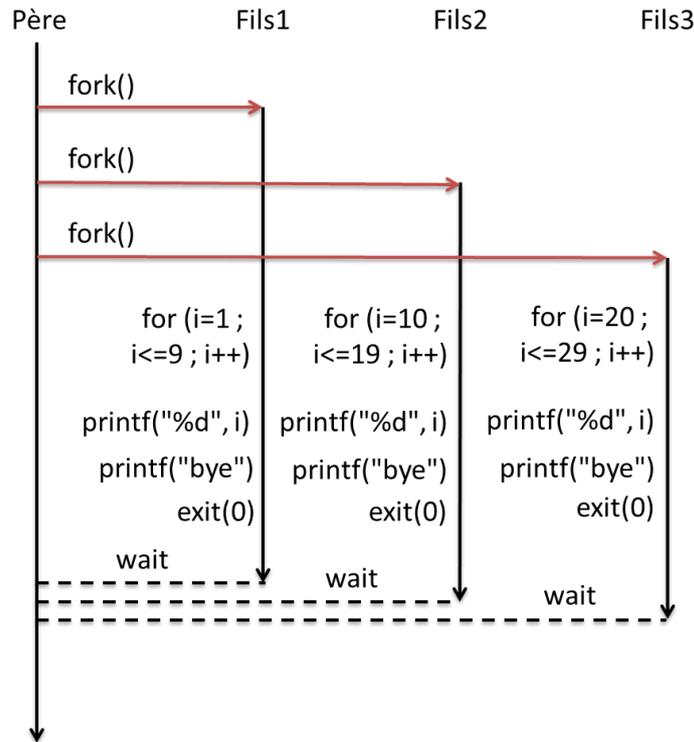




Le père devra afficher le PID de chacun des fils qu'il a créés et ensuite attendre la fin de l'exécution de ceux-ci, puis afficher un message pour dire que tout s'est bien déroulé.

1) Donner le schéma d'exécution d'un tel programme.



2) Annoter votre schéma avec les instructions systèmes permettant de réaliser les étapes clés du programme.

Cf 1)

3) Donner le pseudo-code (C ou algo) d'un tel programme.

```
#include <stdio.h>
#include <stdlib.h>

void fils(int nfiles) {
    int i;

    printf("[%d] FILS PID %d, PERE PID %d\n", getpid(), getpid(),
    getppid());

    for (i = 0 ; i<10 ; i++) {
        printf("[%d] %d\n", getpid(), i+nfiles*10);
    }

    printf("[%d] FILS : au revoir !\n", getpid());
    exit(0);
}

int main() {
```

```
int i;

printf("[%d] bonjour, je suis le père !\n",getpid());

for (i = 0 ; i<3 ; i++) {
    if (fork()==0) {
        fils(i);
    }
}

for (i=0 ; i<3 ; i++) {
    wait(NULL);
}

printf("[%d] PERE : au revoir !\n",getpid());

return 0;
}
```

- 4) Donner un nouveau schéma d'exécution annoté afin de modifier le comportement du programme afin qu'il affiche les nombres dans l'ordre croissant. A quoi cela correspond-t-il au niveau de la concurrence entre processus ?

*Plusieurs solutions :*

- *Par exemple, le père crée un fils lorsque le précédent a fini son travail.*
- *Ou encore les fils se synchronisent par des signaux (lancé par le père, les fils eux-mêmes ou un processus dédié à la synchronisation).*
- ...

### **Exercice 3 (Sémaphores : synchronisation et exclusion mutuelle) – 5 points**

Soit un programme exécutant 3 processus, chacun des trois processus  $P_i$  comportent trois tâches identifiées  $T_{i1}$ ,  $T_{i2}$ ,  $T_{i3}$  ( $i = 1, 2$ , ou  $3$ ).

On considère les processus  $P_1$ ,  $P_2$ ,  $P_3$  :

- Chacun des processus  $P_i$  est constitué des trois tâches  $T_{i1}$ ,  $T_{i2}$ ,  $T_{i3}$  (dans cet ordre) ;
- Ces tâches sont également régies par les relations de précédence (début( $T_{i2}$ )>fin( $T_{i3}$ ) et début( $T_{i2}$ ) > fin( $T_{i1}$ )) ;
- Les tâches  $T_{i2}$  ( $i = 1, 2$  ou  $3$ ) sont en exclusion mutuelle.

- 1) Rappeler les 2 primitives d'utilisation des sémaphores et préciser leurs rôles respectifs.

● P(S)

```

|| S.cpt ← S.cpt - 1
|| si S.cpt < 0 alors
||     bloquer l'appelant
||     le placer dans la file
|| FSi
    
```

● V(S)

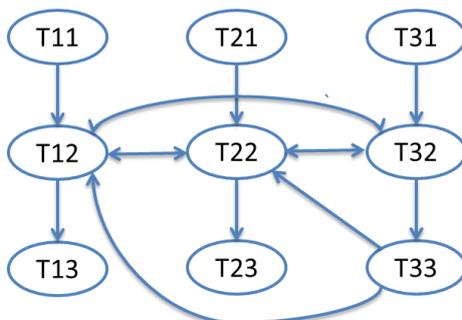
```

|| S.cpt ← S.cpt + 1
|| si S.cpt <= 0 alors
||     sortir processus q de S.f
||     reveiller(q)
|| FSi
    
```

*P(S) consiste à prendre un drapeau (donc attendre si celui-ci n'est pas levé).*

*V(S) consiste à rendre le drapeau (donc le lever).*

2) Dresser le diagramme de précédence complet illustrant ces mises en concurrences.



3) Détailler les pseudos-codes des trois processus ; vous préciserez également les valeurs d'initialisation des compteurs de ces sémaphores.

```

int main() {
S1.Init_sémaphore(0) ; // synchro début(T12)>fin(T33)
S2.Init_sémaphore(0) ; // synchro début(T22) > fin(T33)
S3.Init_sémaphore(1) ; // exclusion mutuelle T12, T22, T32
    
```

```

if (fork() == 0) {
    T11();
    P(S1);
    P(S3);
    T12();
    V(S3);
    T13();
    exit(0);
}
    
```

```

if (fork() == 0) {
    T21();
    P(S2);
    P(S3);
    T22();
    V(S3);
    T23();
    exit(0);
}
    
```

```

if (fork()==0) {
    T31();
    P(S3);
    T32();
    V(S3);
    T33();
    V(S1);
    V(S2);
    exit(0);
}
    
```

**Exercice 4 (Tubes) – 6 points**

Soit un programme comportant 1 processus père et 3 processus fils.

Chaque processus fils à sa spécialisation :

- Le premier est capable de faire des additions ;
- Le deuxième est capable de faire des multiplications ;
- Le troisième est capable de faire des soustractions.

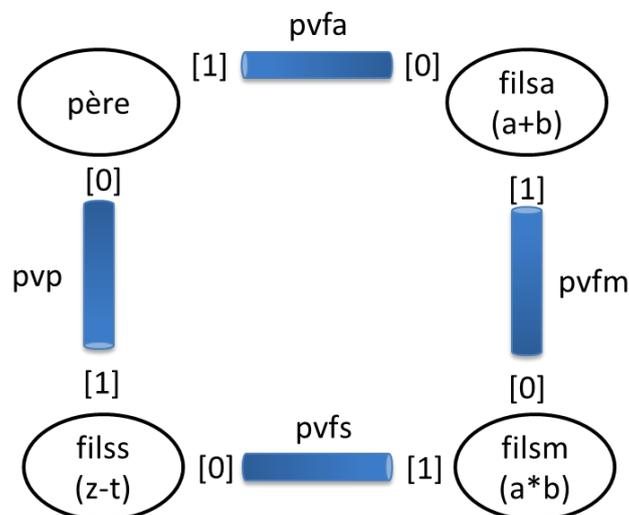
Le but du programme est de réaliser des opérations du type :  $X + Y * Z - T$ .

Déroulement des opérations :

- Le père enverra X, Y, Z et T au fils spécialisé dans l'addition, à l'aide d'un tube.
- Ensuite, le fils additionneur réalisera l'addition et donnera le résultat de l'addition ainsi que Z et T au fils spécialisé dans la multiplication, à l'aide d'un tube.
- Le fils multiplicateur réalisera la multiplication et donnera le résultat ainsi que T au fils spécialisé dans la soustraction, à l'aide d'un tube.
- Le dernier fils réalisera la soustraction et enverra le résultat au père, à l'aide d'un tube.
- Le père affichera le résultat.

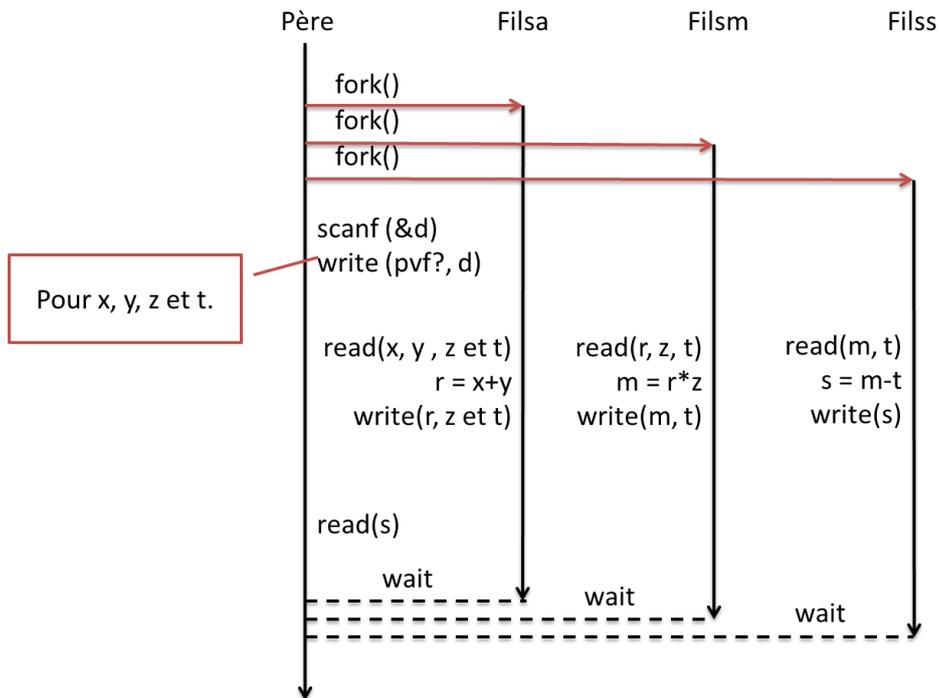
1) Donner le schéma de communication faisant apparaître les processus et les tubes associés.

*Précision pour la correction : on ne tient pas compte des priorités opératoires, en fait, on calcule  $((X+Y)*Z)-T$ .*



2) Donner le schéma d'exécution d'un tel programme.

3) Annoter votre schéma avec les instructions systèmes permettant de réaliser les étapes clés du programme.



- 4) Donner le pseudo-code du père.
- 5) Donner le pseudo-code du fils spécialisé dans l'addition.
- 6) Donner le pseudo-code du fils spécialisé dans la multiplication.
- 7) Donner le pseudo-code du fils spécialisé dans la soustraction.