Université de Reims
Champagne-Ardenne
U.F.R. de Sciences
Exactes et Naturelles

Licence
Informatique
Info 0403
2020/2021

Travaux Dirigés 5

Tubes

Exercice 1

1) Écrire un programme permettant de transférer le contenu d'un tableau d'entiers d'un père vers un fils en utilisant un tube. Le fils affichera le contenu du tableau reçu.

```
int main(int argc, char* argv[]){

   int PvF[2];
   int buf[5] = {0,0,0,0,0};

   pipe(&PvF[0]);

   if (fork()==0){
      close(PvF[1]);
      read(PvF[0],&buf,5 * sizeof(int));
      printf("Tab Recu %d %d %d %d %d,Proc  %d\n",buf[0],buf[1],buf[2],buf[3],buf[4],getpid());
      close(PvF[0]);
      exit(0);
   }

   close(PvF[0]);

   for (int i =0 ; i<5;i++){
      buf[i]=i;
   }

   write(PvF[1],&buf, 5 * sizeof(int));

   wait(NULL);

   close(PvF[1]);

         exit(0);
}
```

2) Modifier le programme précédent pour que le fils inverse l'ordre du tableau et le retourne au père qui l'affichera.

```
int main(int argc, char* argv[]){

   int pvf[2];
   int fvp[2];

   int buf[3] = {1,2,3};

   pipe(&pvf[0]);
   pipe(&fvp[0]);

   if (fork()==0){
      close(pvf[1]);
      close(fvp[0]);

      read(pvf[0], &buf, 3 * sizeof(int));

      buf[0] = 3;
      buf[1] = 2;
      buf[2] = 1;

      write(fvp[1], &buf, 3 * sizeof(int));

      close(pvf[0]);
      close(fvp[1]);

      exit(0);
   }

   close(pvf[0]);
   close(fvp[1]);


   write(pvf[1], &buf, 3 * sizeof(int));
   read(fvp[0], &buf, 3 * sizeof(int));

   printf("Reception de : [%d, %d, %d]. \t Processus:  %d\n",buf[0],buf[1],buf[2],getpid());

   close(pvf[1]);
   close(fvp[0]);

         exit(0);
}
```

Exercice 2 (Interblocages)

L'objectif de cet exercice est de transférer une chaine de caractères via un tube entre un père et un fils. Nous proposons comme solution le code suivant :

```
/* Code du père */
write(p[1], "Bonjour", sizeof(char)*7);
sleep(10);
spid = wait(0);
exit(0);
```

```
/* Code du fils */
r = read(p[0], &c, sizeof(char));
while(r!=0){
  printf("Caract`ere c n", c);
  r = read(p[0], &c, sizeof(char));
}
```

1) Expliquez pourquoi ce programme n'est pas satisfaisant.

On a un interblocage, les deux processus s'attendent l'un l'autre. Le processus fils ne sait pas quand s'arrêter car on lit caractère par caractère et non un nombre précis.

2) Proposez une m´méthode permettant de transférer des chaines de longueurs différentes.

Les deux processus doivent s'entendre sur la taille de la chaine. Le père envoi la taille avant la chaine.

Exercice 3 (Tubes nommés)

Les tubes nommés fonctionnent sensiblement de la même manière que les tubes non nommés excepté deux différences :

— Les programmes de lecture et d'´écriture dans le tube sont dans deux deux fichiers différents puisqu'ils correspondent `a deux processus différents.
— Le pipe est créé indépendamment des deux processus et avant leur emploi. Pour cela, on utilise la fonction : int mknod(const char *path, mode t mode, dev t dev);. Il est possible d'utiliser la commande UNIX mknod nom tube p.

1) Écrire un programme C permettant de créer un tube nommé de nom np. Est-il visible par une commande : ls ? Si oui, quelle est sa taille ?

```
int main(int argc, char* argv[]){
        mknod("np", 0777 | S_IFIFO, 0 );
        exit(0);
}
```

3) Écrire un programme C permettant d'´écrire 2000 caractères dans le tube et un autre programme permettant de lire ces 2000 caractères.

```
int main(int argc, char* argv[]){

   int fd = open("np", O_WRONLY);

   char buf[2000] = "C'est un trou de verdure où chante une rivière, Accrochant follement aux herbes des haillons D'argent ; où le soleil, de la montagne fière, Luit : c'est un petit val qui mousse de rayons. Un soldat jeune, bouche ouverte, tête nue, Et la nuque baignant dans le frais cresson bleu, Dort ; il est étendu dans l'herbe, sous la nue, Pâle dans son lit vert où la lumière pleut. Les pieds dans les glaïeuls, il dort. Souriant comme Sourirait un enfant malade, il fait un somme : Nature, berce-le chaudement : il a froid. Les parfums ne font pas frissonner sa narine ; Il dort dans le soleil, la main sur sa poitrine, Tranquille. Il a deux trous rouges au côté droit.";

   write(fd,&buf[0],2000 * sizeof(char));

   close(fd);
   exit(0);
}
```

```
int main(int argc, char* argv[]){

   int fd = open("np", O_RDONLY);

   char buf[2000] ;

   read(fd,&buf[0],2000 * sizeof(char));

   printf("%s\n",buf);

   close(fd);
   exit(0);
}
```

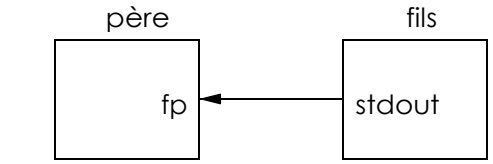
Exercice 4 (Tube vers un programme externe)

Nous utilisons les commandes suivantes :

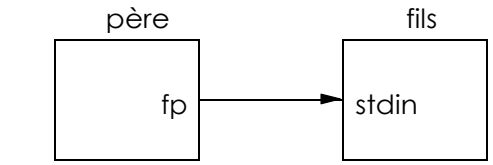
— FILE *popen(const char *cmd, const char *type); : ouverture d'un tube vers un processus fils.

— int pclose(FILE *stream); : fermeture du tube.

La fonction popen fonctionne de la manière suivante :



(1)



(2)

Le cas (1) est effectif lorsque type vaut "w" et le cas (2) lorsqu'il vaut "r".

1) Écrire un programme C permettant de sauvegarder dans un fichier la sortie de la commande ls `a l'aide des fonctions précédentes.

2) Écrire un programme valeur qui demande à l'utilisateur d'entrer un entier et l'affiche `a l'´écran. Il se termine lorsque l'utilisateur entre -1. Ensuite, ´écrire un autre programme exécutant valeur et qui lui envoie successivement différentes valeurs