

Licence informatique — Programmation Système

Devoir Surveillé

Durée : 1h30 — Notes de cours autorisées

Lisez l'intégralité du sujet avant de commencer à répondre aux questions. Certaines peuvent être traitées même si vous n'avez pas répondu aux précédentes.

1 Mode utilisateur, mode noyau

Rappelez ce que signifie l'exécution d'un processus "en mode utilisateur". Même question pour le "mode noyau". Expliquez à quelles occasions s'effectuent les transitions entre ces modes.

Pourquoi la présence de ces deux modes est-elle nécessaire au fonctionnement d'un système d'exploitation multi-utilisateurs ? Illustrez à l'aide d'un exemple.

2 Gestion des fichiers

Question 1 On dispose de deux programmes (ecrire et lire) dont voici les sources :

```
/* ecrire.c */

static char *message = "bonjour";

int main(int argc, char **argv)
{
    int fd, i;

    fd = open("fichier", O_WRONLY | O_CREAT, 0666);
    for(i=0; i<strlen(message); i++)
        write(fd, &message[i], 1);
    close(fd);
}
```

```
/* lire.c */

#define MAX 3

static char tampon[MAX];

int main(int argc, char **argv)
{
    int fd, i, n;

    fd = open("fichier", O_RDONLY);
    while(1) {
        n = read(fd, tampon, MAX);
        if(n == 0)
            break;
        for(i=0; i<n; i++)
            putchar(tampon[i]);
        putchar('\n');
    }
    close(fd);
}
```

On suppose que l'utilisateur lance d'abord le programme ecrire, puis le programme lire (dans le même répertoire). Indiquez le contenu du fichier "fichier" à l'issue du programme ecrire, puis indiquez précisément ce qu'affichera le programme lire à l'écran¹.

Question 2 Si on lance simultanément deux processus exécutant le programme ecrire, quel sera le contenu du fichier à l'issue des deux exécutions ? Expliquez précisément pourquoi.

Question 3 On veut modifier le programme ecrire pour qu'il n'écrive dans le fichier que les caractères de rang impair du message, mais à la bonne position. Si la variable message contient la chaîne "bonjour", le programme devra écrire 'b' à la position 0, 'n' à la position 2, etc. Pour ce faire, vous utiliserez l'appel système lseek. On appellera ecrire_impair cette nouvelle version du programme.

Donnez le code de ecrire_impair.c.

¹Rappel : la fonction putchar imprime un caractère sur la sortie standard.

Question 4 On suppose l'existence d'un programme analogue `ecrire_pair.c` qui n'écrit que les caractères de rang pair du message.

Le résultat de l'exécution de `ecrire_impair` suivi de celle de `ecrire_pair` sera-t-il le même que celui de l'exécution de `ecrire_pair` suivi de `ecrire_impair` en ce qui concerne le contenu final du fichier ?

3 Communication par tubes

Voici le source d'un programme `lanceur.c` destiné à créer un certain nombre de processus reliés par des tubes.

```
int main(int argc, char **argv[])
{
    int i = 0;
    int tube[2];
    int in = STDIN_FILENO;
    int out = STDOUT_FILENO;
    int nb = atoi(argv[1]);

    do {
        pipe(tube);
        if(fork() == 0) {
            close(tube[1]); in = tube[0];
        } else {
            close(tube[0]); out = tube[1];
            break;
        }
        i++;
    } while(i < nb-1);

    printf("Processus %d\n", i);
    return 0;
}
```

Question 1 Expliquez précisément le déroulement de ce programme et illustrez à l'aide d'un dessin. Quel sera l'affichage produit par une exécution de la commande `lanceur 3` ?

Question 2 On souhaite étendre ce programme pour lancer une chaîne de processus reliés par l'intermédiaire de tubes. Concrètement, on souhaite passer en paramètre de `lanceur` une liste de commandes, comme par exemple :

`lanceur ls cat wc`

et on souhaite que cela produise le même effet² que si l'on avait tapé la commande shell :

`ls | cat | wc`

Indiquez les modifications à apporter au programme `lanceur.c` pour obtenir ce comportement.

4 Tableaux dynamiques

Il s'agit d'implanter en mémoire des tableaux d'entiers dynamiques à 3 dimensions (`tab3D`) de telle façon à pouvoir les utiliser le plus *naturellement* possible, i.e. comme les tableaux déclarés statiquement dans le code (e.g. `tab[2][6][2]`).

1. Dessiner l'implantation mémoire d'un tableau dynamique 3D, définir le type `tab3D`;
2. écrire le code de `tab3D creer(size_t m, size_t l, size_t c)`;
3. écrire le code de `void tab3D detruire(tab3D t)`.

N.B. Une réponse simplifiée au cas 2D rapportera aussi des points.

²Sauf que l'on ne se souciera pas du problème relatif à l'ordre de terminaison entre les processus.