
Interrogation (*durée 50min*)

*Documents et notes de cours non autorisés.
Le barème est indicatif.*

1 Questions de cours (*5 pts*)

1. Expliquez, en illustrant avec un exemple simple, le fonctionnement des signaux UNIX.
2. Donnez les deux façons possibles pour la création d'un processus sous UNIX. Précisez le fonctionnement de chacune d'entre elles.
3. Quels sont les moyens mis à disposition du développeur permettant de synchroniser des processus ? (donnez simplement les mots-clés)
4. Expliquez le rôle de la fonction `dup()` (donnez un exemple si nécessaire).

2 Applications (*15 pts*)

2.1 Analyse de code (*10 pts*)

Soit le code source *C* (Source 1) correspondant au programme *inception* et donné par Léo à son réveil. Répondez aux questions suivantes :

1. Que fait ce programme ? Et qu'affiche t'il ?
2. Commentez chaque ligne importante du programme puis donnez une représentation des processus ainsi que des descripteurs ouverts et fermés (comme vu en TD).
3. Tous les tubes sont-ils correctement fermés ?
4. Donnez l'équivalent de ce programme en bash.

```

#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void dream();
void dream_within_dream();

char chemin[100];
int fd[2];
int fd2[2];

int main(int argc, char *argv[]){
    char c;
    if(argc > 2){
        fprintf(stderr, "Usage : %s [chemin]", argv[0]);
        exit(EXIT_FAILURE);
    }

    if(argc==2)
        strncpy(chemin, argv[1], 100);
    else
        strncpy(chemin, ".", 100);

    pipe(fd);

    if(!fork())
        dream();

    close(fd[1]);

    printf("Réalité, la taille du répertoire %s : \n", chemin);

    while(read(fd[0], &c, 1)>0)
        printf("%c",c);

    printf("\n");
    close(fd[0]);

    exit(EXIT_SUCCESS);
}

void dream(){
    pipe(fd2);
    close(fd[0]);

    if(!fork())
        dream_within_dream();

    close(fd2[1]);

    close(STDIN_FILENO);
    dup(fd2[0]);
    close(fd2[0]);

    close(STDOUT_FILENO);
    dup(fd[1]);
    close(fd[1]);

    execl("/usr/bin/tail", "tail", "-nl", NULL);
    fprintf(stderr, "Erreur execl dream, réveil avec un seau d'eau");
    exit(EXIT_FAILURE);
}

void dream_within_dream(){
    close(fd[1]);
    close(STDOUT_FILENO);

    close(fd2[0]);
    dup(fd2[1]);
    close(fd2[1]);

    execl("/usr/bin/du", "du", "-hc", chemin, NULL);
    fprintf(stderr, "Erreur execl dream_within_dream, réveil en se jetant par la fenêtre");
    exit(EXIT_FAILURE);
}

```

Source 1 – "Code source fourni par Léo à son réveil"

2.2 Correction de code (5 pts)

Un développeur chevronné vous a fourni le code source *C* suivant (Source 2). Il est persuadé que tout fonctionne comme prévu. Répondez aux questions de façon précise :

1. Tel qu'il est écrit, que fait ce programme ?
2. Précisez, en justifiant, les lignes du programme qui posent problème.
3. Proposez une version correcte du programme qui permet d'écrire la *date* du jour dans le fichier *toto* puis de renommer celui-ci en *titi* (remplir directement sur le sujet).

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

int main(){
    int a, a1;

    close(STDOUT_FILENO);
    open("toto", O_RDWR|O_CREAT|O_APPEND);

    execl("/bin/date", "date", NULL);

    a=fork();
    if(!a){
        execl("/bin/mv", "mv", "toto", "titi", NULL);
        fprintf(stderr, "Erreur Exec\n");
        exit(2);
    }
}
```

Source 2 – "Code source fourni par le développeur"