

Questions types DS M3101

Quelques exemples de questions qui ressemblent ou sont de même nature que celles que vous rencontrerez au DS de fin de semestre.

Les questions de cours sous forme de QCM et de questions à développer représentent la moitié des points, l'analyse et la production de code l'autre moitié.

Questions de cours (QCM et Question à développer)

- Quels outils sont disponibles pour synchroniser des threads ?
- Qu'est-ce qu'une section critique ?
- Quelles sections mémoires sont initialisées avant le lancement d'un processus ?
- Qu'est-ce que l'ordonnancement ?
- Donnez une définition de l'adressage virtuel ? Quel est son utilité ?
- Qu'est-ce qu'un signal ?

Analyse de code

À la fin de l'exécution du programme ci-dessous, quelle sera la valeur de la variable **G** ?

- 1
- 0
- 4

- 3
- 2
- Il est impossible de le savoir

```
int G = 0;

void* op(void* param){
    G++;
}

int main(int argc, const char *argv[])
{
    pid_t id;

    id = fork();
    id = fork();

    op(NULL);

    return 0;
}
```

Le programme suivant a pour but de faire boucler **indéfiniment** un message entre un processus et son fils. Relevez au moins deux erreurs qui empêcheront le programme de fonctionner correctement.

```
int main(int argc, const char *argv[])
{
    pid_t ident;

    int tubeAller[2], tubeRetour[2];
    char message[20];

    if (pipe(tubeAller) != 0 ||
        pipe(tubeRetour) != 0){
        return EXIT_FAILURE;
    }

    ident = fork();
    switch (ident)
    {
        case -1 :
            return EXIT_FAILURE;
        case 0 :
            close(tubeRetour[0]);
            close(tubeAller[1]);
            while (1){
                sleep(1);
                printf("Fils : je lis dans le tube\n");
                read(tubeAller[0], message, 5);
                printf("Fils : j'ecris 'ping' dans le tube\n");
                write(tubeRetour[1], "ping", 5);
                break;
            }
        default :
            close(tubeRetour[0]);
            close(tubeAller[1]);
            while (1){
                printf("Pere : je lis dans le tube\n");
                read(tubeRetour[0], message, 5);
                printf("Pere : j'ai lu '%s'\n", message);
                printf("Pere : j'ecris 'pong' dans le tube\n");
                write(tubeAller[0], "pong", 5);
            }
    }
}
```

Production de code

On souhaite programmer l'application qui contrôle qu'un conducteur de TGV est bien réveillé (sinon le train doit être arrêté). Le conducteur doit régulièrement actionner un levier pour indiquer

qu'il est réveillé. L'action du conducteur sur le levier déclenche l'envoi du signal **SIGINT** par le noyau à notre programme. Si le conducteur n'actionne pas le levier pendant 30 secondes, l'application doit appeler la procédure `arreterLeTrain()` (supposée donnée) avant de s'arrêter elle-même.

Un rappel doit être affiché toutes les 10 secondes sur la sortie standard :

```
Il reste 20 secondes...  
Il reste 10 secondes...
```

Indications : pour gérer le temps, on programmera un compte à rebours de 10 secondes et on utilisera un compteur initialisé à 3. On rappelle que la fonction `alarm` démarre un décompte et qu'un signal **SIGALRM** est envoyé au processus par le noyau à la fin de ce décompte.

Donnez le code C de ce programme sur votre copie, pas sur le sujet.