

Examen d'Exercices Unix

Appels système classiques : deviner des nombres pensés.

Ecrire un programme C composé d'un père et d'un fils. Le père génère des nombres aléatoires compris entre 2 et 20 que le fils doit faire deviner par l'utilisateur. L'usage du programme est :

`devine max`

où `devine` est le nom de l'exécutable et `max` le nombre maximum de nombres à deviner.

Le fils demande à l'utilisateur d'introduire ses propositions sur l'entrée standard et envoie cette proposition à son père via un pipe.

Le père lit sur le pipe le nombre envoyé par son fils et le compare au nombre généré aléatoirement actuel, puis répond au fils soit : « Vous avez proposé un diviseur du nombre à trouver », « Vous avez proposé un multiple du nombre à trouver », « Votre proposition n'est ni un multiple ni un diviseur » ou « Bravo, vous avez deviné ». Dans ce cas, le père affiche le nombre à deviner et le nombre d'essai qu'il a fallu pour y parvenir, passe au nombre suivant et ce jusqu'à ce qu'il ait proposé `max` nombres.

Le fils demande à l'utilisateur d'entrer des nombres compris entre 2 et 20 et les envoie à son père. Si l'utilisateur introduit autre chose, on lui demande une proposition correcte. Le fils s'arrête soit quand le père a proposé les `max` nombres, soit si l'utilisateur tape <return>, Ctrl-D ou Ctrl-C au clavier.

Avant de se terminer, le père affiche un message avec le nombre de nombres devinés et la moyenne d'essais qu'il a fallu pour chacun d'eux.

Exemple d'exécution où ce qui est écrit par le père est en gras, ce qui est écrit par le fils est en italique souligné et ce que l'utilisateur introduit est en caractère « normal »

```
devine 3
Devine un nouveau nombre
2
Votre proposition n'est ni un multiple ni un diviseur
3
Votre proposition n'est ni un multiple ni un diviseur
5
Votre proposition n'est ni un multiple ni un diviseur
7
Votre proposition n'est ni un multiple ni un diviseur
11
Votre proposition n'est ni un multiple ni un diviseur
13
Votre proposition n'est ni un multiple ni un diviseur
17
Vous avez devine le nombre 17 en 7 essais
Bravo, vous avez bien devine
Devine un nouveau nombre
2
Vous avez propose un diviseur du nombre a deviner
3
Vous avez propose un diviseur du nombre a deviner
6
Vous avez propose un diviseur du nombre a deviner
```

Examen d'Exercices Unix

12

Vous avez devine le nombre 12 en 4 essais

Bravo, vous avez bien devine

Devine un nouveau nombre

2

Vous avez propose un diviseur du nombre a deviner

3

Votre proposition n'est ni un multiple ni un diviseur

5

Vous avez propose un diviseur du nombre a deviner

10

Vous avez devine le nombre 10 en 4 essais

Bravo, vous avez bien devine

Vous avez devine 3 nombres avec en moyenne 5 essais par nombre

Vous trouverez ci après un canevas du code. Vous devez compléter celui-ci aux endroits prévus à cet effet.

Examen d'Exercices Unix

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

```
/* macros éventuelles */
```

```
/* déclarations globales */
```

```
/* routine de traitement du signal SIGINT */
```

```
int generer(int nb) {
    return (int)(rand() / (RAND_MAX + 1.0) * (nb - 1)) + 2;
}
```

```
int main(int argc, char ** argv) {
    /* déclarations communes au père et au fils */
```

Examen d'Exercices Unix

```
/* traitement des arguments */
if (argc != 2) {
    fprintf(stderr,
        "usage : %s max\n", argv[0]);
    exit(1);
}
max = strtoul(argv[1], &p, 10);
if (*p != '\0') {
    fprintf(stderr,
        "le nombre de nombres a deviner doit etre un entier positif\n");
    exit(2);
}

/* création du fils et des moyens de communication avec lui */


/* traitement du père */
if (pid) { /* père */
    /* déclarations propres au père */


/* installation du traitement du signal SIGINT */
/* à faire ici ou dans le fils ? */


/* gestion des moyens de communications */
```


Examen d'Exercices Unix

```
/* génération des questions */
tabNombres = (int *)malloc(max * sizeof(int));
srand(time(NULL));
for (i = 0; i < max; i++) {
    int nombre = generer(MAX);
    tabNombres[i] = nombre;
}
/* boucle de traitement */
for (i = 0; i < max; i++) {
    int nombre = tabNombres[i];
    /* envoi du message nouveau nombre au fils */
    char * message = "Devine un nouveau nombre\n";

    /* boucle d'essais */
    while (1) {
        /* déclaration s éventuelles */

        /* lire la proposition provenant du fils */

        /* verifier si le fils a termine */

        /* analyse de la proposition et construction du message */
    }
}
```

Examen d'Exercices Unix

```
        /* envoyer le message au fils */

    }
    /* verifier si le fils a termine */

} /* fin de la boucle de traitement du père */
/* affichage des statistiques */

/* prévenir le fils que la session est finie */

/* attendre la fin du fils */

    exit(0);
} /* fin du père */

/* traitement du fils */
/* déclarations propres au fils */

/* installation du traitement du signal SIGINT */
/* à faire ici ou dans le père ? */

/* gestion des moyens de communications */
```

Examen d'Exercices Unix

```
/* boucle de traitement */

while (1) {
    /* déclarations éventuelles */

    /* afficher le message du père */
    SYS((lu = read(pipePF[0], ligne, 255)));
    if (lu == 0)
        break;
    ligne[lu] = '\0';
    printf("%s", ligne);

    do { /* boucle de traitement du nombre courant */

        /* attendre la proposition de l'utilisateur */

        /* vérifier que c'est un entier entre 2 et MAX */
```

Examen d'Exercices Unix

```
/* envoyer la proposition au père */
```

```
/* lire le message envoyé par le pere */
```

```
/* afficher le message du père */
```

```
    } while (  
    /* détecter si le fils a terminé */
```

```
    } /* fin de la boucle de traitement du fils */  
    /* terminer proprement */
```

```
}
```