Mardi 7 septembre 2004
Durée : 3 heures
Documents autorisés

Licence Informatique
Programmation Système
2003 - 2004

Examen

Les directives d'inclusions de fichiers en-tête n'apparaissent pas dans les programmes de l'énoncé. De plus, les valeurs de retour des primitives ne sont pas systématiquement testées. Sauf mention contraire, on pourra faire de même pour les autres programmes que l'on écrira.

1 Le langage *C*

Exercice 1 *On considère le programme suivant :*

```
1   int main(void)
2   {
3       char tab[6]={65,66,67,68,69,0};
4       long i;
5       char *ptr;
6
7       for (i=0; tab[i]; i++) printf("%d , %c\n", tab[i], tab[i]);
8       printf("%s\n",tab);
9       exit(EXIT_SUCCESS);
10  }
```

a. *Donner ce qui est affiché pendant l'exécution.*

b. *Réécrire la ligne 7 en utilisant la variable* `ptr` *au lieu de* `i`.

Exercice 2 *Ecrire une fonction qui prend une chaîne de caractères et renvoie la valeur vraie si c'est un palindrome et faux sinon.*

Exercice 3 *Donner (sans justification) ce qui est affiché par le programme suivant.*

```
1   int main(void)
2   {
3       long x=7,y=4,z=2;
4
5       printf("%d\n",x-y-z);
6       printf("%d\n",x&&y-z);
7       printf("%d\n",x+y%z );
8       printf("%d\n",x&y+z );
9       exit(EXIT_SUCCESS);
10  }
```

2 Programmation système

Exercice 4

a. *Donner une commande* `bash` *permettant de compter les fichiers sources C (autrement dit les fichiers qui se terminent par* `.c` *ou* `.h`*).*

b. *Ecrire un programme qui exécute cette commande en utilisant les fonctions* `pipe`, `dup` *et* `execlp`.

Exercice 5 *On considère le programme suivant :*

```
1   int main(void)
2   {
3       if ( fork() && fork() ) fork();
4       exit(EXIT_SUCCESS);
5   }
```

a. *Donner l'arbre généalogique des processus créés.*

b. *Quel arbre obtient-on si l'on remplace l'opérateur* `&&` *par* `*` *?*

Exercice 6 *On considère le programme suivant :*

```
1   int main(void)
2   {
3       long i=0;
4
5       fork(); fork(); fork();
6       i++;
7       printf("%d\n",i);
8       exit(EXIT_SUCCESS);
9   }
```

a. *Combien de lignes sont affichées ?*

b. *Combien de processus sont créés ?*

c. *Quelles sont les valeurs affichées ?*

Exercice 7 *On souhaite faire la concaténation de deux fichiers texte.*

a. *Ecrire un programme qui prend deux noms de fichiers texte en argument et ajoute les lignes du premier fichier à la suite du second fichier.*

b. *Ecrire une commande* `bash` *équivalente au programme précédent.*

Exercice 8 *On souhaite calculer les entiers de la suite de Fibonacci.*

a. *Ecrire un programme* `suivant` *qui lit deux entiers p et q sur son entrée standard puis affiche q et $p+q$ sur sa sortie standard.*

b. *Utiliser le programme* `suivant` *pour écrire une commande* `bash` *qui affiche le quatrième nombre de Fibonacci.*

c. *Utiliser le programme* `suivant` *pour écrire un programme qui prend un entier n en argument et affiche le $n_{ème}$ nombre de Fibonacci.*

Exercice 9 *On propose une autre façon de calculer les entiers de la suite de Fibonacci.*

a. *Ecrire un programme qui crée deux processus communiquant dans les deux sens par tubes : chaque processus reçoit deux entiers p et q de l'autre processus et lui renvoie q et $p+q$. Les deux entiers de départ sont pris en argument par le programme.*

b. *Modifier le programme précédent de façon à ce que le processus père affiche la valeur de p lorsqu'il reçoit le signal* `SIGINT` *et termine lui et son fils.*

Exercice 10 *Dire (en justifiant) ce qui est affiché pendant l'exécution du programme suivant.*

```
1   int main(void)
2   {
3       int tube[2];
4
5       signal(SIGPIPE,SIG_IGN);
6       pipe(tube); close(tube[0]);
7       if (write(tube[1],"A",1)==-1) perror("erreur");
8       printf("fin\n");
9       exit(EXIT_SUCCESS);
10  }
```

Exercice 11 *Donner les différents affichages pouvant être produits par le programme suivant.*

```
1   void interruption(int signum) { printf("UN\n"); }
2
3   int main(void)
4   {
5       signal(SIGCHLD, &interruption);
6       if ( fork() ) { printf("DEUX\n"); sleep(2); }
7       else printf("TROIS\n");
8       exit(EXIT_SUCCESS);
9   }
```