



+90/1/2+

L3 Informatique

Année 2024-2025



SY5 – Systèmes d'exploitation

QCM du 23 octobre 2024

Durée : 30 minutes

Énoncé constitué de 17 questions

Compléter le cadre d'identification ci-contre (numéro étudiant à coder en cochant une case par ligne).

Pour sélectionner une réponse, la case correspondante doit être *entièrement noircie*. Utiliser un stylo noir ou bleu foncé.

Pour annuler la sélection d'une réponse, vous pouvez soit effacer la coche, soit la recouvrir de correcteur blanc *sans redessiner la case*.

Dans l'ensemble du QCM, *toto* désigne un fichier ordinaire de contenu "abcdefghijkl" et *buf* un tableau de type `char[32]` rempli d'octets nuls (tous deux réinitialisés au début de chaque question). Les erreurs ne sont pas gérées ; on suppose que tous les appels système réussissent.

Question 1 Combien d'appels à `write` le code suivant provoque-t-il, si la sortie standard est un terminal ?

```
printf("abc\n"); printf("def");  
printf("ghi\n"); printf("jkl"); exit(0);
```

- ☐ 0 ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ plus

Question 2 Quel est l'affichage obtenu après exécution du code suivant ?

```
fd = open("toto", O_RDONLY); read(fd, buf, 6);  
lseek(fd, -3, SEEK_CUR); read(fd, buf, 6);  
printf("%s", buf);
```

- ☐ abcdefghijkl ☐ abcdefdefghi ☐ abcdef
☐ abcdefjkl ☐ ghijkl ☐ rien
☒ defghi ☐ jkldef ☐ autre chose

Question 3 Quel est l'affichage obtenu après exécution du code suivant ?

```
fd = open("toto", O_RDONLY);  
do { nb = read(fd, buf, 8); } while (nb > 0);  
printf("%s", buf);
```

- ☒ ijklfgh ☐ ijkl ☐ efghijkl
☐ abcdefghijkl ☐ abcdijkl ☐ rien
☐ abcdefgh ☐ ijklijkl ☐ autre chose

Question 4 Combien d'appels à `write` le code suivant provoque-t-il, si la sortie standard est redirigée sur un fichier ordinaire ?

```
printf("abc\n"); printf("def");  
printf("ghi\n"); printf("jkl"); _exit(0);
```

- ☒ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ plus

Question 5 Que contient *toto* après exécution de `a.out >| toto` ?

```
fprintf(stderr, "a"); printf("b");  
fd = dup(1); dup2(2, 1); dup2(fd, 2);  
fprintf(stderr, "c"); printf("d");
```

- ☐ d ☐ b ☐ ac ☒ c ☐ abd ☐ bd
☒ bc ☐ abcd ☐ rien ☐ autre chose

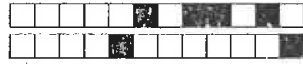
Question 6 Cocher tous les tests permettant de déterminer si un `mode_t m` est celui d'un fichier ordinaire.

- ☐ `(m & S_IFMT) == S_IFREG`
☒ `(m & S_IFMT) == S_IFREG`
☐ `(m & S_IFREG) == S_IFREG`
☐ `(m & S_IFREG) == S_IFREG`
☐ `m != S_IFDIR`
☐ `m == S_IFREG`

Question 7 Quelle est la plus petite valeur que peut afficher ce programme s'il est exécuté dans un répertoire contenant deux sous-répertoires vides ?

```
cpt = 0; d1 = opendir(".");  
while((e1 = readdir(d1)))  
    if ((d2 = opendir(e1->d_name))) {  
        while((e2 = readdir(d2))) cpt += 1;  
        closedir(d2);  
    }  
printf("%d", cpt);
```

- ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6
☐ 7 ☐ 8 ☐ 9 ☐ 10 ☒ 11 ☐ plus



Pour chacune des 4 questions suivantes, cocher exactement les éléments nécessaires (valeur de flags et présence d'un 3^e argument) lors de l'appel `open(fic, flags, ...)` pour pouvoir simuler la redirection indiquée. On supposera que l'option `noclobber` du shell est activée, i.e. que les fichiers existants sont protégés contre l'écrasement par `>`, qui peut être forcé par `>|`.

Question 8 Pour simuler une redirection `>> fic` :

- ☐ O_RDWR ☒ O_CREAT ☐ O_TRUNC
☒ O_WRONLY ☐ O_RDONLY ☒ O_EXCL
☒ O_APPEND ☒ le 3^e argument est nécessaire

Question 9 Pour simuler une redirection `> fic` :

- ☒ O_CREAT ☐ O_RDONLY ☒ O_WRONLY
☒ O_TRUNC ☒ O_EXCL ☐ O_RDWR
☐ O_APPEND ☒ le 3^e argument est nécessaire

Question 10 Pour simuler une redirection `< fic` :

- ☐ O_RDWR ☐ O_EXCL ☐ O_CREAT
☒ O_RDONLY ☐ O_WRONLY ☐ O_TRUNC
☐ O_APPEND ☐ le 3^e argument est nécessaire

Question 11 Pour simuler une redirection `>| fic` :

- ☐ O_TRUNC ☒ O_WRONLY ☐ O_APPEND
☐ O_RDONLY ☐ O_EXCL ☒ O_CREAT
☐ O_RDWR ☒ le 3^e argument est nécessaire

Après définition de deux descripteurs `fd1` et `fd2`, on exécute les instructions suivantes :

```
nb = read(fd1, buf, 3); write(STDOUT_FILENO, buf, nb); write(fd1, "***", 3);
```

```
nb = read(fd2, buf, 3); write(STDOUT_FILENO, buf, nb); write(fd2, "...", 3);
```

Indiquer dans chaque cas ci-dessous l'affichage obtenu, et le contenu final du fichier `toto` (il y a donc a priori 2 cases à cocher pour chaque question).

Question 12 `fd1 = open("toto", O_RDWR | O_TRUNC); fd2 = dup(fd1);`

- | | | | |
|---|---------------------------------|--|--|
| <input type="checkbox"/> abcdefghijkl***... | <input type="checkbox"/> abcghi | <input checked="" type="checkbox"/> ***... | <input type="checkbox"/> abc***ghijkl... |
| <input type="checkbox"/> abc | <input type="checkbox"/> ... | <input type="checkbox"/> abc...ghijkl*** | <input type="checkbox"/> abc...ghijkl |
| <input type="checkbox"/> abc***ghi... | <input type="checkbox"/> abcabc | <input type="checkbox"/> *** | <input checked="" type="checkbox"/> rien |

Question 13 `fd1 = open("toto", O_RDWR); fd2 = dup(fd1);`

- | | | | |
|---|--|--|---------------------------------|
| <input type="checkbox"/> abcdefghijkl***... | <input type="checkbox"/> *** | <input type="checkbox"/> ... | <input type="checkbox"/> ***... |
| <input type="checkbox"/> abc***ghijkl... | <input checked="" type="checkbox"/> abc***ghi... | <input checked="" type="checkbox"/> abcghi | <input type="checkbox"/> abc |
| <input type="checkbox"/> abcabc | <input type="checkbox"/> abc...ghijkl*** | <input type="checkbox"/> abc...ghijkl | <input type="checkbox"/> rien |

Question 14 `fd1 = open("toto", O_RDWR); fd2 = open("toto", O_RDWR | O_APPEND);`

- | | | | |
|---|---|---------------------------------|---|
| <input type="checkbox"/> abcabc | <input type="checkbox"/> abc...ghijkl | <input type="checkbox"/> ***... | <input checked="" type="checkbox"/> abc |
| <input type="checkbox"/> abc***ghi... | <input type="checkbox"/> abc...ghijkl*** | <input type="checkbox"/> abcghi | <input type="checkbox"/> *** |
| <input checked="" type="checkbox"/> abc***ghijkl... | <input type="checkbox"/> abcdefghijkl***... | <input type="checkbox"/> ... | <input type="checkbox"/> rien |

Question 15 `fd1 = open("toto", O_RDWR); fd2 = open("toto", O_RDWR);`

- | | | | |
|---------------------------------|---|--|--|
| <input type="checkbox"/> ... | <input type="checkbox"/> abc***ghi... | <input checked="" type="checkbox"/> abc...ghijkl | <input type="checkbox"/> abc***ghijkl... |
| <input type="checkbox"/> *** | <input type="checkbox"/> abc...ghijkl*** | <input type="checkbox"/> abc | <input checked="" type="checkbox"/> abcabc |
| <input type="checkbox"/> abcghi | <input type="checkbox"/> abcdefghijkl***... | <input type="checkbox"/> ***... | <input type="checkbox"/> rien |

Question 16 `fd1 = open("toto", O_RDWR | O_APPEND); fd2 = dup(fd1);`

- | | | | |
|--|---------------------------------------|--|--|
| <input type="checkbox"/> abc...ghijkl*** | <input type="checkbox"/> *** | <input type="checkbox"/> abc | <input type="checkbox"/> abcghi |
| <input type="checkbox"/> abc***ghi... | <input type="checkbox"/> abc...ghijkl | <input type="checkbox"/> abc***ghijkl... | <input type="checkbox"/> ... |
| <input type="checkbox"/> ***... | <input type="checkbox"/> abcabc | <input checked="" type="checkbox"/> abcdefghijkl***... | <input checked="" type="checkbox"/> rien |

Question 17 `fd1 = open("toto", O_RDWR | O_APPEND); fd2 = open("toto", O_RDWR | O_APPEND);`

- | | | | |
|--|--|---------------------------------------|--|
| <input type="checkbox"/> ... | <input type="checkbox"/> ***... | <input type="checkbox"/> abc***ghi... | <input type="checkbox"/> *** |
| <input checked="" type="checkbox"/> abcdefghijkl***... | <input type="checkbox"/> abc***ghijkl... | <input type="checkbox"/> abcghi | <input type="checkbox"/> abcabc |
| <input type="checkbox"/> abc | <input type="checkbox"/> abc...ghijkl*** | <input type="checkbox"/> abc...ghijkl | <input checked="" type="checkbox"/> rien |