

<b>Commencé le</b>	dimanche 2 avril 2023, 08:52
<b>État</b>	Terminé
<b>Terminé le</b>	dimanche 2 avril 2023, 08:54
<b>Temps mis</b>	1 min 46 s
<b>Points</b>	24,00/8,00
<b>Note</b>	<b>30,00</b> sur 10,00 ( <b>300%</b> )
<i>Dans l'évaluation avec indication de certitude, la note ci-dessus est affichée relativement au maximum pour toutes les réponses correctes, avec C = 1. ?</i>	
<b>Résultats tout le test (8 questions)</b>	
<b>Évaluation moyenne du degré de certitude</b>	3,00
<b>Précision</b>	100,0%
<b>Bonus degré de certitude</b>	0,0%
<b>Précision + bonus</b>	100,0%
<b>Ventilé par certitude</b>	
<b>C = 3</b>	Réponses : 8. Précision : <b>100%</b> . (Plage optimale 80% à 100%). Vous étiez <b>OK</b> en sélectionnant ce degré de certitude.
<b>C = 2</b>	Pas de réponse
<b>C = 1</b>	Pas de réponse

**Question 1**

Terminé

Évaluation du degré de certitude 3,00

Pondération 1,00

Posix utilise des descripteurs de fichiers pour faire référence aux fichiers dans les fonction read, write, close (et ce qui est retourné par open). Ce descripteur de fichier est:

Veuillez choisir une réponse.

- ☐ 1. Un pointeur sur une structure de données FILE
- ☐ 2. Un tableau de caractères
- ☒ 3. Un nombre entier
- ☐ 4. Un pointeur sur une chaîne de caractères contenant le nom du fichier (chemin et nom de fichier)

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

**Question 2**

Terminé

Évaluation du degré de certitude 3,00

Pondération 1,00

Quelle exécution de commande est équivalente au code Posix suivant:

```
struct dirent *dentry;
DIR *f = opendir("foo");
if (f == NULL) {
    perror("Erreur ouverture\n");
    exit(EXIT_FAILURE);
}
while((dentry = readdir(f)) != NULL) {
    printf("%s\n", dentry->d_name);
}
closedir(f);
```

Veuillez choisir une réponse.

- ☒ 1. ls foo
- ☐ 2. ls -R foo
- ☐ 3. ls
- ☐ 4. ls -l foo

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

**Question 3**

Terminé

Évaluation du degré de certitude 3,00

Pondération 1,00

Il existe plusieurs modes de gestion des entrées sorties pour un système d'exploitation:

Veuillez choisir au moins une réponse.

- ☒ 1. Mode par canal DMA (Direct Memory Access)
- ☒ 2. Mode programmé par interruption
- ☐ 3. Mode par vol de priorité au processus
- ☒ 4. Mode programmé simple (boucle d'attente active)

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

**Question 4**

Terminé

Évaluation du degré de certitude 3,00

Pondération 1,00

Quand on ouvre un dossier avec la fonction Posix opendir, on récupère un élément de quel type ?

Veuillez choisir au moins une réponse.

- ☐ 1. Identique à un descripteur de fichier car un dossier sous Unix est en fait un fichier
- ☒ 2. NULL si le dossier n'est pas accessible
- ☐ 3. Un pointeur sur une chaîne de caractère correspondant au chemin + nom du dossier
- ☐ 4. -1 si le dossier n'est pas accessible
- ☒ 5. Un pointeur sur une structure DIR

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

**Question 5**

Terminé

Évaluation du degré de certitude 3,00

Pondération 1,00

L'ensemble des informations sur un fichier (droits d'accès, liens, propriétaire, groupe, taille, date de modification, ...) sont accessibles en Posix grâce à:

Veuillez choisir une réponse.

- ☐ 1. Des variables stockant chacune de ces informations
- ☐ 2. Des macros permettant d'avoir accès à chaque des informations
- ☐ 3. Ces informations ne sont pas accessibles en Posix
- ☒ 4. Une structure de données *stat* qui regroupe l'ensemble des informations

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

**Question 6**

Terminé

Évaluation du degré de certitude 3,00

Pondération 1,00

Quelle exécution de commande est équivalente au code Posix suivant (foo et bar sont des fichiers):

```
char buffer[MAX];
int n;
int fd1 = open("bar", O_WRONLY|O_TRUNC|O_CREAT, 0600);
int fd2 = open("foo", O_RDONLY|O_EXCL);

while((n = read(fd2, buffer, MAX)) != 0)
    write(fd1, buffer, n);
close(fd1);
close(fd2);
```

Veuillez choisir une réponse.

- ☒ 1. cp foo bar
- ☐ 2. cp bar foo
- ☐ 3. cat foo bar
- ☐ 4. mv foo bar
- ☐ 5. cp -R foo bar

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

**Question 7**

Terminé

Évaluation du degré de certitude 3,00

Pondération 1,00

Les commandes Unix comme *ls*, *cd*, *cp*, *mv*, *rm*, ... sont programmées en C à l'aide des primitives Posix de gestion des entrées-sorties?

Veuillez choisir une réponse.

- ☐ 1. Non
- ☒ 2. Oui

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

**Question 8**

Terminé


Évaluation du degré de certitude 3,00

Pondération 1,00

En Posix, à l'aide d'un descripteur de fichier vous pouvez accéder à:

Veuillez choisir au moins une réponse.

- ☒ 1. La sortie standard d'un processus
- ☒ 2. Aux données issues du clavier
- ☒ 3. Un fichier sauvegardé sur un support de stockage quelque soit le formatage
- ☐ 4. A des données issues de la carte réseau

Degré de certitude  : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)