0/1

1/1

0/1

1/1

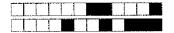
0/1

Note: 11/20 (score total : 6/11)



+49/1/24+

	LIFSE Contrôle	
	Contrôle du mardi $28/02/23$ - 15 minutes	Numéro d'étudiant :
	Nom: BROILLET	
	Prénom: Vingile	
	No. étu. : 12103804	
	Utilisez un stylo noir (pas au crayon de bois), et répondez uniquement dans les cadres prévus à cet effet.	
	Aucun document n'est autorisé. Les téléphones, ordinateurs, et toutes communication avec les autres étudiants sont interdits. Une seule réponse par question.	
ς	Question 1 Le chemin/./ désigne	
	🔀 le répertoire parent	
	le répertoire courant	
	□ le répertoire enfant Question 2 En ligne de commandes, quel est le nom de la variable d'environnement qui contient la liste de tous les répertoires où vont être cherchés les exécutables ? □ HOME □ USER □ PATH □ PWD Question 3 D'après la page de manuel, l'en-tête de l'appel système read() est ssize_t read(int fd, void *buf, size_t count). On précise que le fichier /dev/random existe et peut être lu pour obtenir des octets aléatoires. On considère l'extrait de code suivant :	
s.		
1	<pre>int f = open("/dev/random", O_RDONLY);</pre>	
2 3 4	<pre>char t; int nbrd = read(f, &t, 8); close(f);</pre>	
Ċ	Ce morceau de code	
	va provoquer une erreur à la compilation	
(va s'exécuter normalement, et charger une octet (aléatoire) dans t isque de provoquer une erreur de segmentation à l'exécution Question 4 Dans le contexte des systèmes d'exploitation, PPID est l'acronyme de	
Ç		
	Processing Process IDentifier	
	Parent Process IDentifier	
	Parent Process Instruction Decoder	
Ç	puestion 5 Dans le contexte des fichiers, le typage MIME précise	
	k le type des fichiers sur le web	
	les devoirs des utilisateurs sur les fichiers	
	les droits des utilisateurs sur les fichiers	



1 Exercice

1/1

1/1

1/1

0/1

1/1

0/1

echo '\$return'

On considère le code suivant, qui permet de réaliser une commande mycat qui ressemble à cat (en incluant les bons fichiers, ce programme pourrait être compilé sans problème):

```
#define LEN 16
         int main(int argc, char *argv[]) {
      2
      3
           char buf [LEN];
           int fd = open(argv[1], 0_RDONLY);
      4
           int nbrd = read(fd, buf, LEN);
      5
           while(nbrd > 0) {
      6
      7
             int nrem = nbrd;
      8
             int nbwr = 0;
             while(nrem > 0) {
     10
               int t = write(STDOUT_FILENO, buf+nbwr, nrem);
     11
              nrem -= t;
     12
              nbwr += t;
     13
     14
             nbrd = read(fd, buf, LEN);
     15
           }
     16
           close(fd);
     17
           return 0;
     18
         }
               Si on appelle mycat avec ./mycat titi.txt toto.txt alors
Question 6
 🥵 dans le programme argv[1] désigne la chaîne "titi.txt"
 le programme se termine sur un échec
 dans le programme argv[1] désigne la chaîne "toto.txt"
Question 7
               Dans ce programme, fd est le descripteur du fichier
  🎇 sur lequel on va lire des octets avec read()
 sur lequel on va lire des octets avec write()
 utilisé pour écrire à l'écran avec avec write()
               Les appels à read() des lignes 5 et 14 permettent de lire
Question 8
  exactement LEN octets
  au moins LEN octets
  au plus LEN octets
               buf+nbwr à la ligne 10 est, avant l'appel à write(), l'adresse mémoire
  jusqu'à laquelle les octets lus sur STDOUT_FILENO doivent être écrits dans buf
  X à partir de laquelle les octets à écrire sur STDOUT_FILENO doivent être chargés
 a partir de laquelle les octets lus sur STDOUT_FILENO doivent être rangés
                 La valeur de nour après l'exécution de la ligne 12 est
Question 10
  🔣 le nombre d'octets présents dans le buffer buf déjà écrits
  le nombre d'octets présents dans le buffer buf restant à écrire
  le nombre d'octets qui vont être écrits sur STDOUT_FILENO par l'appel
                 On lance le programme avec ./mycat tutu.txt dans le shell ; quelle commande
Question 11
permet d'afficher l'entier retourné par le programme?
  echo $!
  X echo "$?"
```