

Unix et Programmation Shell

aucun document autorisé

4 juillet 2022

Q 1. Cochez le système intrus :

GNU/Linux	android	ZURIX
BSD	MacOS	UNIX

Q 2. Quels sont les auteurs d'un système de cryptographie à clef publique vu en cours ?

Adleman	Rivest	Ader
Shamir	Clostermann	Richtofen

Q 3. Cochez les outils du C-programmeur

valgrind	gcc	gimp
gdb	tac	make

Q 4. On considère le système RSA de module 2021. Cochez les exposants de chiffrement admissibles :

1	2	3	4	5
---	---	---	---	---

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void oops( int d, int p )
5 {   int table[p];
6     printf ( "%d\n", d );
7     oops( d+1, p );
8 }
9 int main( int argc, char * argv[] ){
10     int p = atoi( argv[1] );
11     oops( 0, p );
12     return 0;
13 }
```

Q 5. Quelle ligne du code `stack.c` est responsable du message d'erreur :

```

~> gcc stack.c
~> ./a.out
Erreur de segmentation (core dumped)
```

Q 6. Quelle zone mémoire est concernée par l'erreur :

```

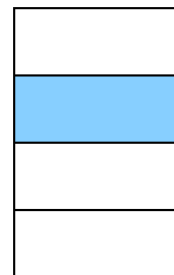
~> gcc stack
~> ./a.out 1024
...
2010
2011
Erreur de segmentation (core dumped)
```

Q 7. A votre avis, quel sera le dernier nombre affiché par la commande :

```

~> gcc stack
~> ./a.out $( bc <<< '2^12' )
```

1024	512	64
------	-----	----



parametres

```

void
swap( int *x, int* y )
{int z;
  z = *x;
  *x = *y;
  *y = z;
}
```

Q 8. Décrire les cases du cadre de pile de la fonction `swap`, en précisant le contenu de la case sombre.

```

1 #include <stdio.h>
2 int main( int argc, char* argv[] )
3 { printf( argv[1], argv[2], argv[3] );
4   return 0;
5 }
```

prog.c

Q 9. Quel argument 1 faut-il passer à la commande `prog.exe` pour obtenir l'affichage "hello world" ?

```

~> gcc prog.c -o prog.exe
~> ./prog.exe ..... hello world
hello world
```