



SI3 - Examen ProgSys - Durée 20 minutes - Aucun document autorisé

<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0
<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1
<input checked="" type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2
<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input checked="" type="checkbox"/>	3
<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4
<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input checked="" type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input checked="" type="checkbox"/>	5	<input type="checkbox"/>	5
<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input checked="" type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6
<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7
<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8
<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9

← Veuillez noircir les cases correspondantes à votre Numéro Etudiant (NE) (1 case/ligne et 1 case/colonne). Pour NE=21056798, il faut noircir le 2 dans la 1ère colonne, le 1 dans la 2ème colonne etc ...

Ecrivez votre Nom et votre N° étudiant (NE)

BEROUK.....
21605053.....

A LIRE OBLIGATOIREMENT AVANT DE COMMENCER ...



Barème : Toutes les questions comportent au moins une bonne réponse. Toute bonne réponse rapporte des points et toute mauvaise réponse en fait perdre. Les questions avec une * peuvent avoir une ou plusieurs bonnes réponses.

Q. 1 * Un Makefile sert principalement:

- ☒ À décrire comment compiler un projet
- ☒ À gagner du temps pour recompiler un projet après modification de quelques fichiers sources
- ☒ À faire de la compilation séparée
- ☒ À décrire comment charger les bibliothèques pour exécuter un projet

Q. 2 * Que me permet la commande suivante: `ldd prog.exe`

- ☐ Vérifier les bibliothèques statiques utilisées par le programme
- ☒ Vérifier les bibliothèques dynamiques utilisées par le programme
- ☐ Créer une bibliothèque dynamique nommée `prog.exe`
- ☐ Debugger le programme

Q. 3 Un debugger permet de revenir en arrière dans l'exécution:

- ☒ Oui ☒ Non

Q. 4 * Un système d'exploitation est composé:

- ☒ De librairies
- ☒ D'un ensemble de programmes utilitaires
- ☒ D'un noyau

Q. 5 Le mécanisme de chargement de bibliothèques dynamiques sous Linux et Windows est complètement différent?

- ☐ Oui ☒ Non

Q. 6 Une bibliothèque partagée `.dll` peut être utilisée pour tout exécutable sous Linux?

- ☒ Non ☐ Oui

Q. 7 * Un système d'exploitation gère:

- ☒ les entrées-sorties
- ☒ les ressources de calcul de la machine
- ☒ les ressources mémoire de la machine

Q. 8 * Quels outils permettent de chercher les bugs d'un programme?

- ☐ ldd ☒ ddd ☒ gdb ☐ ar

Q. 9 * Posix est une norme:

- ☒ D'interface direct du noyau
- ☐ D'interface de communication entre machines
- ☒ D'interface de la bibliothèque C

Q. 10 * Quelles sont les options de compilation de gcc nécessaires lors des différentes étapes de création d'une bibliothèque dynamique:

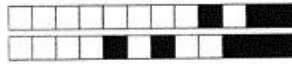
- ☐ -Wall ☒ -fpic ☐ -g ☒ -shared

Q. 11 Pour un exécutable donné, il y a moyen de connaître les bibliothèques statiques utilisées?

- ☒ Non ☐ Oui

Q. 12 * Il est possible de faire les combinaisons suivantes:

- ☒ Bibliothèque statique, édition de liens statique
- ☒ Bibliothèque dynamique, édition de liens dynamique
- ☐ Bibliothèque statique, édition de liens dynamique
- ☒ Bibliothèque dynamique, édition de liens statique



Q. 13 Soit deux fichiers objet *file1.o* *file2.o* et soit une bibliothèque statique créée avec ces deux fichiers *.o*. Un programme *prog1.exe* utilisant la bibliothèque statique et le *prog2.exe* créé en incluant les deux *.o* auront obligatoirement une taille:

- ☐ *prog1.exe* > *prog2.exe*
☒ *prog1.exe* <= *prog2.exe*
☐ *prog1.exe* = *prog2.exe*

0/1

Q. 14 Soit les variables *int a[]* et *int b*, avec *ddd*, si je demande l'affichage **a@(b)*:

- ☐ J'obtiens une erreur
☒ J'affiche les "b" valeurs à partir de l'adresse pointée par *a*
☐ J'affiche les "b" adresses à partir de l'adresse pointée par *a*
☐ J'affiche l'adresse du pointeur *a*

1/1

Q. 15 * Quelles commandes permettent de créer une bibliothèque:

- ☒ *gcc* ☐ *nm* ☐ *ldd* ☒ *ar*

1/2

Q. 16 Pour un exécutable donné, il y a moyen de connaître toutes les fonctions définies y compris celles issues des bibliothèques statiques utilisées?

- ☐ Non
☒ Oui, grâce à la commande *nm*
☐ Oui, grâce à la commande *ldd*

0/1

Soit le code suivant:

```
int var1;  
char var2[] = "buf1";  
main() {  
    int var3;  
    static int var4;  
    static char var5[] = "buf2";  
    char * var6;  
    var6 = malloc(512);  
}
```

Q. 17 * Quelles sont les variables qui seront dans *bss*?

- ☒ *var1*
☐ *var2*
☐ *var3*
☒ *var4*
☐ *var5*
☐ *var6*

2/2

Q. 18 * Soit la fonction suivante. Que provoquera cette fonction?

```
int *f() {  
    int x = 42;  
    return &x;  
}
```

- ☐ Une erreur à l'édition de liens
☒ Un problème à l'exécution (comportement indéfini)
☒ Un avertissement à la compilation
☐ Une erreur à la compilation

0/2

Q. 19 Quand je lie une bibliothèque statique à un programme C

- ☒ Tous les *.o* de la bibliothèque qui ont au moins une fonction utilisée par mon programme sont ajoutés à l'exécutable
☐ Seules les fonctions utilisées par mon programme sont ajoutées à l'exécutable
☐ Toutes les fonctions de la bibliothèque sont ajoutées à mon exécutable

0/1

Q. 20 * Quelle est l'utilité de la commande *ranlib*?

- ☒ Ajouter un index à une bibliothèque statique
☐ Créer une bibliothèque dynamique
☒ Faire l'édition de liens d'une bibliothèque statique

-0.5/1

Q. 21 * En C/Posix, le chargement de bibliothèque dynamique avec édition de liens dynamique est utilisé pour:

- ☐ Pour rendre le programme plus rapide
☒ Pour la mise en oeuvre de plugins dans des programmes
☒ Pour changer dynamiquement l'implémentation de fonctions
☐ Pour permettre l'exécution du programme sous plusieurs OS (Windows, Linux, ...)

1/2

Q. 22 * *ddd*, basé sur *gdb*, permet:

- ☒ D'exécuter un programme pas à pas
☒ De trouver facilement la ligne de code sur laquelle un programme fait Segmentation Fault
☐ De modifier le fil d'exécution du programme
☐ De trouver automatiquement les bugs d'un programme
☒ D'afficher les valeurs de variables pendant l'exécution

2/3