

L2 informatique - Université Paris Est à Marne-La-Vallée

Architecture des systèmes informatiques : Examen seconde session 2018-2019

Durée de l'épreuve : 2 heures

Les notes de cours, de TD et de TP ne sont pas autorisées. Les ordinateurs, calculatrices ainsi que les téléphones sont interdits et éteints durant l'épreuve. Il est possible à tout moment d'admettre la réponse (algorithme, fonction, programme, etc.) à une question non traitée à condition de le préciser explicitement. Il est très fortement préconisé de lire exhaustivement l'énoncé avant de commencer à répondre aux exercices.

Les justifications et explications sont toujours plus importantes que les résultats attendus aux questions. Ainsi, la qualité de la rédaction de vos réponses aura un impact très important sur la notation. Fournir un résultat correct sans aucune explication ne donne aucun point. Fournir une explication valide avec un résultat erroné peut donner un grand nombre de points sur les questions.

Le barème est juste donné à titre indicatif, il donne un indice sur la longueur et la difficulté des exercices.

Exercice 1 : Ouais, j'ai lu le cours 10 minutes... (4 points)

- 1) Combien il y a-t-il de transistors dans un ordinateur moderne ? Donnez un ordre de grandeur et décrivez les endroits de la machine où ces derniers sont situés.
- 2) Qu'est ce qu'un processus ? Quelles informations peut-on trouver dans un processus ?
- 3) Comment sont organisés les droits sur les fichiers en Unix ?
- 4) Quels sont les objectifs de l'ordonnanceur du système Unix ?

Exercice 2 : Lire dans les pensées (4 points)

Pour chacune des commandes qui suivent, tentez de déterminer les intentions précises de son auteur. Il s'agit ici de deviner quelle information l'auteur de la commande veut obtenir. Il est inutile de décrire et décomposer la commande, il faut en deviner la finalité.

- 1) `ps -ef | grep "root" | wc -l`
- 2) `find / -name "stdio.h" 2> /dev/null`
- 3) `ls -al | grep "-rwxrwxrwx"`
- 4) `find -name ".py" | xargs chmod +x`

Exercice 3 : changements de base (4 points)

Traduire les chiffres suivants de la base 10 vers la base 7 :

66, 48, 114, 400

Les calculs étant plus importants que les résultats. Veuillez faire apparaître toutes les étapes de calcul vous permettant d'opérer le changement de base.

Exercice 4 : Table de vérité (4 points)

Écrire une fonction C qui affiche la table de vérité de l'expression logique **a ou ((non b) et c)**.

Voici une suggestion d'affichage en console

```
a | b | c | expr
0 | 0 | 0 | 0
0 | 0 | 1 | 1
etc...
```

Exercice 5 : Traçage d'exécution (4 points)

Un grand écrivain propose le fichier suivant qu'il sauvegarde sous le nom **zarby.py**.

```
#!/usr/bin/python3
```

```
def zarby(n, m):
    print(" " * m + str((n, m)))
    if (m == 0):
        if (n == 0):
            return 1;
        return 1 + zarby(n-1, n-1)
    return 1 + zarby(n, m-1)
```

```
print(zarby(5, 2))
```

Cette personne essaie d'exécuter ce nouveau fichier fraîchement créé mais obtient le message suivant :

```
nborie@bayer:$ ./zarb.py
```

```
bash: ./zarb.py: Permission non accordée
```

- 1) Quelle est l'erreur produite par cet informaticien de niveau 1 ? Comment peut-on exécuter son fichier ?
- 2) Quel est l'affichage produit en console lorsque l'on exécute son fichier ?

Exercice 6 : Décodage (4 points)

Le flux suivant binaire vient d'être intercepté. Il semblerait que ce flux soit en fait des caractères en code ascii. Tentez de décoder ce flux. Voici un morceau de la table ascii pour vous aider.

a	b	c	d	e	f	g	h	i	j	k	l	m
97	98	99	100	101	102	103	104	105	106	107	108	109
n	o	p	q	r	s	t	u	v	w	x	y	z
110	111	112	113	114	115	116	117	118	119	120	121	122

Flux intercepté:

```
0110 0111 0111 0000 0111 0100
0110 0011 0110 1011 0110 1011
0110 0001 0111 0010
```

Quelques Rappels :

ps -ef : affiche tous les processus actifs sur la machine avec leurs informations détaillés, un par ligne

grep motif : recherche les lignes de l'entrée standard contenant le motif en argument

find rep *conditions* : recherche dans rep et ses sous répertoires les éléments (fichier ou répertoire) vérifiant les *conditions*