

# 3R-IN3 – Système d’exploitation – TP évalué

Durée 2h

Conditions de l’épreuve : voir encadré

Xavier Hilaire  
x.hilaire@esiee.fr

5 juin 2024

## Consignes :

- Travail individuel, sur les machines de l’ESIEE à l’exclusion de toute autre chose, sous Linux.
- Documents autorisés, matériel électronique personnel interdit.
- Vous devrez déposer sur <https://mvproxy.esiee.fr>, au plus tard à 10h05 délai de rigueur, une archive ZIP contenant
  - Vos scripts shell
  - Un fichier LibreOffice (format ODT) contenant les captures d’écran de votre terminal qui correspondent aux tests demandés pour chaque exercice. Ce fichier n’a pas besoin de contenir autre chose. Pour faire une capture d’écran : `sleep 3; xwd | xwdtopnm | pnmtopng > /tmp/cap.png` puis cliquez à l’intérieur du terminal.
- Les apprentis en tiers-temps disposent du temps additionnel qui leur est accordé et pourront déposer au-delà de 10h05

## Exercice 1 – informations sur les CPU

Le fichier `/proc/cpuinfo` contient diverses informations sur les processeurs de votre machine.

1. A l’aide de la commande `cat`, affichez ce fichier en redirigeant l’affichage vers un fichier `cpuinfo.txt`. Vous devriez obtenir une sortie listant, en plusieurs blocs, des informations sur le(s) processeur(s) de votre machine. Il est inutile de mentionner le résultat de cette étape dans votre rapport.
2. Utilisez `grep` pour extraire de ce fichier les lignes contenant les mots `processor`, `model name`, et `cpu MHz`, que vous sauvegarderez dans un fichier `filtered_cpinfo.txt`. Le résultat obtenu ressemblera à :

```
processor : 0
model name : Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz
cpu MHz : 855.149
processor : 1
model name : Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz
cpu MHz : 800.000
processor : 2
model name : Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz
```

```
cpu MHz : 800.000
processor : 3
model name : Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz
cpu MHz : 2881.098
```

Il est inutile de mentionner le résultat de cette étape dans votre rapport.

3. En vous appuyant sur les étapes précédentes, écrivez un script shell nommé `lscpu.sh`, qui effectuera les actions suivantes et sortira un listing réduit des CPUs sur la sortie standard :
  - (a) Lire les fichiers de `/proc/cpuinfo`. (point 1 ci-dessus)
  - (b) Filtrer les informations obtenus de `/proc/cpuinfo`. (point 2 ci-dessus)
  - (c) Afficher le modèle et la fréquence de chaque processeur (sans décimale).
  - (d) Compter le nombre de processeurs (si l'option `-c` est fournie).
  - (e) Enlever tout espace superflu dans la sortie (si besoin).

Les seuls binaires autorisés sont : `cat`, `cut`, `echo`, `sed`, `grep`, `tr`. Vous pouvez utiliser toutes les instructions et constructions d'un shell POSIX (`/bin/sh`) ou Bash (`/bin/bash`). Dans votre rapport, fournissez une capture de votre terminal à titre de test.

Exemples de sorties :

```
$> ./lscpu.sh -c
Processeur 0: Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz at 3492 MHz
Processeur 1: Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz at 3588 MHz
Processeur 2: Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz at 3554 MHz
Processeur 3: Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz at 3645 MHz
Nombre de processeurs: 4
$> ./lscpu.sh
Processeur 0: Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz at 3492 MHz
Processeur 1: Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz at 3588 MHz
Processeur 2: Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz at 3554 MHz
Processeur 3: Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz at 3645 MHz
$>
```

## Exercice 2 – commande `ls` personnalisée

La commande `file` permet de connaître la nature de des fichiers qui lui sont passés en arguments :

```
$ file /etc/passwd /bin/ls
/etc/passwd: ASCII text
/bin/ls:      ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, int
$
```

Écrivez un script shell (ou bash) `mysls.sh [-t] fichier...`, qui accepte en entrée un nombre variable d'arguments, et pour chacun d'eux, fera les choses suivantes :

- Si c'est un fichier, il affiche directement ce fichier suivi des informations fournies par `file` à condition qu'il appartienne à l'utilisateur qui exécute le script.
- Si c'est un répertoire, il fait la même chose sur chacun des fichiers qu'il comporte, à l'exclusion de toute autre chose (il ne s'occupe pas des sous-répertoires)
- Si ce n'est ni l'un ni l'autre, alors c'est une erreur, et il devra afficher le nom du fichier suivi d'un message d'erreur sur la sortie standard.

De plus, si l'option `-t` est spécifiée, alors la sortie devra être triée selon le type de chaque fichier.

Exemple de sorties possibles :

```

$ ./mysls.sh /tmp /etc /etc/passwd ~/bin/args1
/tmp/cap.png: PNG image data, 1294 x 1167, 8-bit/color RGB, non-interlaced
/tmp/Presentation_IT_JPO_1_6_2024_courte.pptx: Microsoft PowerPoint 2007+
/tmp/resadjust.log: Unicode text, UTF-8 text
/tmp/tmp0t302i18: PNG image data, 48 x 48, 8-bit/color RGBA, non-interlaced
/tmp/tmp443u1wrt: PNG image data, 48 x 48, 8-bit/color RGBA, non-interlaced
/tmp/tmp7tw794su.png: PNG image data, 48 x 48, 8-bit/color RGBA, non-interlaced
/tmp/tmpphpszf8u.png: PNG image data, 48 x 48, 8-bit/color RGBA, non-interlaced
/home/hilairex/bin/args1: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically li
$ ./mysls.sh -t /tmp /etc /etc/passwd ~/bin/args1
/home/hilairex/bin/args1: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically li
/tmp/Presentation_IT_JPO_1_6_2024_courte.pptx: Microsoft PowerPoint 2007+
/tmp/cap.png: PNG image data, 1294 x 1167, 8-bit/color RGB, non-interlaced
/tmp/tmp0t302i18: PNG image data, 48 x 48, 8-bit/color RGBA, non-interlaced
/tmp/tmp443u1wrt: PNG image data, 48 x 48, 8-bit/color RGBA, non-interlaced
/tmp/tmp7tw794su.png: PNG image data, 48 x 48, 8-bit/color RGBA, non-interlaced
/tmp/tmpphpszf8u.png: PNG image data, 48 x 48, 8-bit/color RGBA, non-interlaced
/tmp/resadjust.log: Unicode text, UTF-8 text
$

```

Validez votre script au moins sur `/tmp`, où vous pourrez créer vos fichiers, mais où devraient s'en trouver d'autres qui ne vous appartiennent pas.

### Exercice 3 – recherche de liens matériels

Comme cela a été expliqué en cours, lorsque l'on crée un lien matériel d'un fichier existant, le fichier et le lien possèdent le même numéro d'inode. On peut voir ce dernier soit avec la commande `ls -li`, soit avec `stat -c '%i'` :

```

$ cp /etc/passwd .
$ ln passwd toto
$ ls -li passwd toto
1074680073 passwd
1074680073 toto
$ stat -c '%i' passwd toto
1074680073
1074680073
$

```

Ecrivez un script shell (ou bash) `liens.sh` acceptant des noms de répertoires ou de fichiers en arguments, et qui fait les choses suivantes :

- Si l'argument est un répertoire, il le parcourt récursivement pour en extraire les fichiers
- Sinon c'est que l'argument est un fichier, donc considère ce fichier directement
- Il affiche alors, par blocs, les fichiers qui sont réalité des liens matériels d'un même fichier.

Exemple d'utilisation :

```

$ curl -k https://perso.esiee.fr/~hilairex/3R-IN3/fichiers-tp.tgz | tar xvpzf - -C $HOME
$ cd ~/fichiers-tp
$ ln html/node134.html lien134.html
$ ln html/node134.html repertoire/rep2/deuxieme.html
$ ln html/node134.html repertoire/troisieme
$ ln repertoire/rep2/fichier.ext $HOME

```

Ces commandes créent trois liens matériels de `node134.html`, toujours à l'intérieur de l'arborescence de `~/fichiers-tp`, et un seul de `fichier.ext`, dans votre HOME utilisateur, donc hors de l'arborescence. Sorties obtenues :

```
$ ./liens.sh ~/fichiers-tp/ ~/fichier.ext
Identifient le même fichier:
| /home/hilairex/fichiers-tp/html/node134.html
| /home/hilairex/fichiers-tp/lien134.html
| /home/hilairex/fichiers-tp/repertoire/rep2/deuxieme.html
| /home/hilairex/fichiers-tp/repertoire/troisieme
Identifient le même fichier:
| /home/hilairex/fichier.ext
| /home/hilairex/fichiers-tp/repertoire/rep2/fichier.ext
$
$ ./liens.sh ~/fichiers-tp/
Identifient le même fichier:
| /home/hilairex/fichiers-tp/html/node134.html
| /home/hilairex/fichiers-tp/lien134.html
| /home/hilairex/fichiers-tp/repertoire/rep2/deuxieme.html
| /home/hilairex/fichiers-tp/repertoire/troisieme
$
```

Vous devriez obtenir idéalement les mêmes sorties, ou quelque chose d'approchant.

Joignez une capture d'écran de votre terminal sur ces tests pour valider votre script.

Les seuls binaires autorisés sont : `test`, `find`, `sort`, et éventuellement : `sed`, `uniq`, et `grep`. Il est inutile de gérer les erreurs liées à une mauvaise utilisation (fichier illisible, argument qui n'est ni un répertoire ni un fichier, etc.).