

INF1070 Utilisation et administration des systèmes informatiques

Examen Final — Automne 2018

Samedi 15 décembre — Durée 3 heures

Jean Privat & Alexandre Blondin Massé

Aucun document ni appareil électronique n'est autorisé. Répondez et rendez seulement le formulaire (dernière page). L'annexe est détachable. Les questions faisant apparaître le symbole ♣ peuvent présenter plusieurs bonnes réponses. Les autres ont une unique bonne réponse. Des points négatifs pourront être affectés à de très mauvaises réponses.

1 Expressions régulières

Question 1 Laquelle des expressions régulières étendues (ERE) suivantes reconnaît les adresses IPv6 comme 2001:0db8:0000:85a3:0000:0000:ac1f:8001?

- | | | | |
|----------------------------|---|----------------------------|--|
| <input type="checkbox"/> A | <code>([0-9][a-f]{4}:){7}[0-9][a-f]{4}</code> | <input type="checkbox"/> D | <code>([0-9a-f]{4}:){7}[0-9a-f]{4}</code> |
| <input type="checkbox"/> B | <code>([0-9a-f:]{4}){8}</code> | <input type="checkbox"/> E | <code>([0-9a-f]{4}:){8}</code> |
| <input type="checkbox"/> C | <code>([0-9][a-f]{4}:){8}</code> | <input type="checkbox"/> F | <code>([0-2][0-9][0-9]\.){3}[0-2][0-9][0-9]</code> |

Question 2 ♣ Parmi les expressions régulières étendues (ERE) suivantes, lesquelles trouvent une correspondance dans le mot « prestidigitateur » ?

- | | | | | | | | |
|----------------------------|--------------------------|----------------------------|---------------------------------|----------------------------|---------------------|----------------------------|--------------------------|
| <input type="checkbox"/> A | <code>(.)...\1</code> | <input type="checkbox"/> C | <code>[^aeiou]{3}</code> | <input type="checkbox"/> E | <code>(.)\1</code> | <input type="checkbox"/> G | <code>(.)(.)*\1\2</code> |
| <input type="checkbox"/> B | <code>^[^restidi]</code> | <input type="checkbox"/> D | <code>(i[tr] ea)[ug]{2,}</code> | <input type="checkbox"/> F | <code>.*..\$</code> | <input type="checkbox"/> H | <code>[aeiou]{2}</code> |

Question 3 ♣ Parmi les expressions régulières étendues (ERE) suivantes, lesquelles trouvent une correspondance **seulement** avec les noms de fichiers qui se terminent par l'une des extensions `.txt`, `.text` ou `.tex` ?

- | | | | | | |
|----------------------------|--------------------------------|----------------------------|------------------------------------|----------------------------|-----------------------------------|
| <input type="checkbox"/> A | <code>\.te?xt?\$</code> | <input type="checkbox"/> C | <code>{.tex,.text,.tex}\$</code> | <input type="checkbox"/> E | <code>\.te?xt\$ \.text?\$</code> |
| <input type="checkbox"/> B | <code>[.]t(xt ext ex)\$</code> | <input type="checkbox"/> D | <code>^.*\.(tex text txt)\$</code> | <input type="checkbox"/> F | <code>\.txt \.text \.tex\$</code> |

Question 4 ♣ Parmi les mots suivants, lesquels ont une correspondance avec les deux expressions régulières étendues (ERE) « `^[sinus]+(di?f?or?)*(m[es])?$` » et « `^[a-1]*[p-z]*[^issues]*$` » ?

- | | | | | | | | | | | | |
|----------------------------|-----|----------------------------|------|----------------------------|------|----------------------------|-----|----------------------------|------|----------------------------|---------|
| <input type="checkbox"/> A | sud | <input type="checkbox"/> B | info | <input type="checkbox"/> C | nord | <input type="checkbox"/> D | nid | <input type="checkbox"/> E | sors | <input type="checkbox"/> F | informe |
|----------------------------|-----|----------------------------|------|----------------------------|------|----------------------------|-----|----------------------------|------|----------------------------|---------|

2 Processus

Question 5 ♣ Parmi les énoncés suivants, indiquez tous ceux qui sont vrais concernant les processus en Unix.

- | | |
|----------------------------|---|
| <input type="checkbox"/> A | Les processus sont isolés et ne peuvent pas collaborer ensemble |
| <input type="checkbox"/> B | Tout processus est rattaché à un terminal |
| <input type="checkbox"/> C | Tout processus est associé à un utilisateur |
| <input type="checkbox"/> D | Un démon est un processus qui ne peut pas être interrompu |
| <input type="checkbox"/> E | Tout processus a au moins un processus enfant |
| <input type="checkbox"/> F | Une tâche shell (<i>job</i>) peut regrouper plusieurs processus |
| <input type="checkbox"/> G | Pour un programme donné, il existe au plus un seul processus à la fois |
| <input type="checkbox"/> H | Deux processus qui coexistent ont nécessairement des identifiants (PID) distincts |

Question 6 Alice a lancé depuis un terminal le programme `./long_calcul`, mais il n'est toujours pas terminé. Que devrait-elle faire pour récupérer le contrôle de son terminal sans perdre le travail en cours ?

- | | | | |
|----------------------------|---|----------------------------|--|
| <input type="checkbox"/> A | Appuyer sur Ctrl-C puis taper <code>bg</code> | <input type="checkbox"/> D | Appuyer sur Ctrl-Z puis taper <code>bg</code> |
| <input type="checkbox"/> B | Appuyer sur Ctrl-C puis taper <code>fg</code> | <input type="checkbox"/> E | Taper <code>%bg</code> suivi de la touche Entrée |
| <input type="checkbox"/> C | Maintenir Ctrl et appuyer sur <code>fg</code> | <input type="checkbox"/> F | Appuyer sur Ctrl-Z puis taper <code>fg</code> |

2.1 Les commandes ps et jobs

Dans un terminal, j'entre la commande « `ps -e -o uid,pid,ppid,c,sz,stime,TTY,time,cmd` » et j'obtiens le résultat suivant (certaines lignes non pertinentes ont été supprimées par économie d'espace) :

```

UID      PID PPID  C   SZ  STIME TTY      TIME CMD
root      1    0  0 56441 16:38 ?        00:00:02 /sbin/init splash
root     270    1  0 41086 16:38 ?        00:00:01 /lib/systemd/systemd-journald
root     295    1  0 11766 16:38 ?        00:00:00 /lib/systemd/systemd-udev
avahi    920    1  0 11855 16:38 ?        00:00:00 avahi-daemon: running
root     934    1  0 250305 16:38 ?       00:00:01 /usr/lib/snapd/snapd
avahi    946   920  0 11769 16:38 ?        00:00:00 avahi-daemon: chroot helper
root    1000    1  0 76552 16:38 ?        00:00:00 /usr/sbin/cups-browsed
gdm     1151    1  0 19247 16:38 ?        00:00:00 /lib/systemd/systemd --user
gdm     1153 1151  0 28079 16:38 ?        00:00:00 (sd-pam)
root    1476    1  0 72442 16:38 ?        00:00:00 /usr/lib/x86_64-linux-gnu/boltd
root    1477    1  0 110386 16:38 ?       00:00:14 /usr/lib/packagekit/packagekitd
whoopsie 1581    1  0 97146 16:38 ?        00:00:00 /usr/bin/whoopsie -f
alice   1589    1  0 19245 16:38 ?        00:00:00 /lib/systemd/systemd --user
geoclue 1834    1  0 116749 16:38 ?       00:00:00 /usr/lib/geoclue-2.0/geoclue -t 5
alice   4323 1589  7 193804 16:59 ?        00:00:50 /usr/[...]/gnome-terminal-server
alice   4421 4323  0  6075 16:59 pts/0    00:00:00 bash
alice   4496 4421  0  5304 16:59 pts/0    00:00:00 tmux
alice   4498 1589  6  8174 16:59 ?        00:00:41 tmux
alice   4534 4498  0  6251 16:59 pts/2    00:00:00 -bash
alice   4568 4498  0  6251 16:59 pts/3    00:00:00 -bash
alice   4602 4498  0  6380 16:59 pts/4    00:00:00 -bash
alice   4981 4498  0  6252 17:07 pts/1    00:00:00 -bash
alice   5019 4534 85  1903 17:07 pts/2    00:02:39 cat /dev/urandom
alice   5020 4534 22  1874 17:07 pts/2    00:00:41 tr -cd ABC
bob     5030 4981  0  1877 17:07 pts/1    00:00:00 tail -f error.log
bob     5031 4568  0  3399 17:08 pts/3    00:00:00 nc -l 8888
alice   5032 4498  0  6251 17:08 pts/5    00:00:00 -bash
alice   5235 4602  0 12182 17:10 pts/4    00:00:00 xeyes
alice   5236 4602  8 202123 17:10 pts/4    00:00:00 gnome-calculator
alice   5248 4602  0  9342 17:10 pts/4    00:00:00 ps -e -o uid,pid,ppid,[...]
alice   5249 4602  0  1871 17:10 pts/4    00:00:00 tee ps.out

```

suivi de la commande « `jobs` » :

```

[1]-  Running                  xeyes &
[2]+  Running                  gnome-calculator &

```

Question 7 ♣ Parmi les énoncés suivants, indiquez tous ceux qui sont vrais.

- A Le terminal courant est identifié par `pts/0`
- B L'identifiant du processus parent de `xeyes` est 5235
- C Le processus qui a le plus consommé de temps CPU a le PID 5019
- D Je suis l'utilisateur `alice`
- E Le PID du processus le plus ancien est 934
- F Le processus de PID 5235 a été lancé en arrière-plan

Question 8 ♣ Parmi les commandes `kill` suivantes, lesquelles vont terminer l'exécution d'un processus avec succès ?

- A `kill -9 1581`
- B `kill %1`
- C `kill -s SIGTERM 4602`
- D `kill %2`
- E `kill 1`
- F `kill 5019`

3 Shell avancé

Question 9 ♣ Parmi les commandes suivantes, lesquelles affichent, entre autres, une ligne contenant « toto » ?

- | | |
|---|--|
| <input type="checkbox"/> A (echo toto; echo tata) tail -n 1 | <input type="checkbox"/> E echo toto > tata; cat tata |
| <input type="checkbox"/> B echo toto echo tata | <input type="checkbox"/> F (echo otot >&2; echo toto) rev |
| <input type="checkbox"/> C echo tata > tr o a | <input type="checkbox"/> G echo toto (echo tata; tail -n 1) |
| <input type="checkbox"/> D echo toto 2>/dev/null | <input type="checkbox"/> H echo to (ta="\$(cat -)"; echo "\$ta\$ta") |

Question 10 Alice et Bob souhaitent communiquer. Pour s'amuser, ils s'entendent pour chiffrer leurs messages de façon très simple avec un algorithme non cryptologiquement sécuritaire. Alice procède de la manière suivante pour envoyer un message à Bob : `cat message | tr a-z u-zf-ta-e | rev > message.enc`

Quelle commande, parmi les suivantes, permet à Bob de déchiffrer le message envoyé par Alice ?

- | | |
|---|---|
| <input type="checkbox"/> A tr u-zf-ta-e a-z < message.enc rev | <input type="checkbox"/> D tac message.enc tr a-z u-zf-ta-e |
| <input type="checkbox"/> B tac message.enc tr u-zf-ta-e a-z | <input type="checkbox"/> E cat message.enc rev untr a-z u-zf-ta-e |
| <input type="checkbox"/> C cat message.enc tr a-z u-zf-ta-e rev | <input type="checkbox"/> F rev tr u-zf-ta-e a-z < message.enc |

Question 11 ♣ Considérez les cinq lignes de commandes suivantes :

```
echo 1; sleep 2; sleep 3      # C1
sleep 1 & sleep 2 & sleep 3  # C2
sleep 1 && sleep 2 && sleep 3 # C3
sleep 1 || sleep 2 || sleep 3 # C4
sleep 1 | sleep 2 | sleep 4  # C5
```

Parmi les énoncés suivants, lesquels sont vrais en prenant en compte la durée d'exécution totale de chaque ligne de commande ?

- | | | |
|--|---|---|
| <input type="checkbox"/> A C5 dure 7 secondes | <input type="checkbox"/> C C1 dure 5 secondes | <input type="checkbox"/> E C2 dure 6 secondes |
| <input type="checkbox"/> B C1 est plus lent que C5 | <input type="checkbox"/> D C4 dure 1 secondes | <input type="checkbox"/> F C3 dure 3 secondes |

4 Scripts

Question 12 Considérez le script exécutable suivant nommé `oui_ou_non` :

```
#!/bin/sh
if [ "$#" -ne 1 ]; then
    exit 1
fi
if [ "$1" = oui ]; then
    echo oui
    ./oui_ou_non non
elif [ "$1" = non ]; then
    echo non
    exit 2
fi
```

Qu'est-ce qui sera affiché sur la sortie standard si on lance la commande « `./oui_ou_non oui; echo $?` »

- | | |
|--|--|
| <input type="checkbox"/> A Seulement « oui » | <input type="checkbox"/> F Seulement « non » |
| <input type="checkbox"/> B « non », suivi de « oui », suivi de « 1 » | <input type="checkbox"/> G « oui », suivi de « non », suivi de « 2 » |
| <input type="checkbox"/> C « non », suivi de « oui », suivi de « 2 » | <input type="checkbox"/> H « oui », suivi de « non », suivi de « 0 » |
| <input type="checkbox"/> D « non », suivi de « oui », suivi de « 0 » | <input type="checkbox"/> I « oui », suivi de « non », suivi de « 1 » |
| <input type="checkbox"/> E « oui » suivi de « non » | <input type="checkbox"/> J « non » suivi de « oui » |

Question 13 ♣ Soit un répertoire `/opt/inf1070` accessible à l'utilisateur.

```
$ ls -l /opt/inf1070/
lrwxrwxrwx 1 inf1070 inf1070      5 dec 15 14:00 exam -> final
-rwxr-xr-x 1 inf1070 inf1070 3754 dec 15 14:00 final
-rwxr-xr-x 1 inf1070 inf1070 3541 oct 27 14:00 intra
$ pwd
/home/etudiant/
$ echo $PATH
/usr/local/bin:/usr/bin:/bin
```

Quelles sont les différentes façons d'exécuter le script shell `final` à partir du répertoire courant ?

- | | | |
|---|---|---|
| <input type="checkbox"/> A <code>final</code> | <input type="checkbox"/> C <code>sh /opt/inf1070/final</code> | <input type="checkbox"/> E <code>./opt/inf1070/final</code> |
| <input type="checkbox"/> B <code>/opt/inf1070/exam</code> | <input type="checkbox"/> D <code>/bin/sh final</code> | <input type="checkbox"/> F <code>./final</code> |

Question 14 Considérez le script exécutable suivant nommé `subst` :

```
#!/bin/sh
a=4
b="echo $a"
a=5
echo "$b"
```

Qu'est-ce qui sera affiché sur la sortie standard si on lance la commande « `./subst` »

- | | | | |
|--|--|--|--|
| <input type="checkbox"/> A <code>echo 4</code> | <input type="checkbox"/> C <code>echo \$a</code> | <input type="checkbox"/> E <code>5</code> | <input type="checkbox"/> G <code>\$b</code> |
| <input type="checkbox"/> B <code>4</code> | <input type="checkbox"/> D <code>"\$b"</code> | <input type="checkbox"/> F <code>echo 5</code> | <input type="checkbox"/> H <code>"echo \$a"</code> |

4.1 Le script `groupes.sh`

Soit le script shell exécutable `groupes.sh` suivant :

```
#!/bin/sh
echo "Bonjour $USER"
echo "Vos groupes:"
id -Gn | grep -Eow $1
echo "Au revoir $(id -un)"
```

Voici un exemple d'utilisation du script :

```
$ id -Gn
privat adm cdrom floppy sudo audio dip video plugdev input netdev lpadmin docker
$ ./groupes.sh '\w+'
Bonjour privat
Vos groupes:
sudo
audio
video
Au revoir privat
```

Question 15 ♣ Lorsque j'exécute la commande « `USER=blondin ./groupes.sh '.u\w+'` », quelles sont les différences avec l'affichage de l'exemple ?

- | | |
|---|---|
| <input type="checkbox"/> A La ligne "sudo" n'est plus affichée | <input type="checkbox"/> D La ligne "video" n'est plus affichée |
| <input type="checkbox"/> B Une nouvelle ligne "plugdev" est affichée | <input type="checkbox"/> E La ligne "Au revoir privat" a changé |
| <input type="checkbox"/> C Un message d'erreur est affiché sur la sortie d'erreur | <input type="checkbox"/> F La ligne "Bonjour privat" a changé |

Question 16 ♣ Même question pour la commande « `id=blondin ./groupes.sh 'audio video'` ».

- | | |
|---|---|
| <input type="checkbox"/> A La ligne "Bonjour privat" a changé | <input type="checkbox"/> D Une nouvelle ligne "plugdev" est affichée |
| <input type="checkbox"/> B La ligne "Au revoir privat" a changé | <input type="checkbox"/> E La ligne "sudo" n'est plus affichée |
| <input type="checkbox"/> C La ligne "video" n'est plus affichée | <input type="checkbox"/> F Un message d'erreur est affiché sur la sortie d'erreur |

5 Réseaux

Question 17 ♣ Soit l'URL `https://www.info.uqam.ca/~privat/inf1070/` de la page web du cours. Lesquelles des affirmations suivantes sont vraies ?

- | |
|--|
| <input type="checkbox"/> A <code>uqam.ca</code> est un TLD (<i>top level domain</i>) |
| <input type="checkbox"/> B Le navigateur web doit déterminer l'adresse MAC du serveur pour accéder à la page web |
| <input type="checkbox"/> C <code>www.info.uqam.ca</code> est un nom d'hôte (<i>hostname</i>) |
| <input type="checkbox"/> D Le protocole utilisé est HTTP sur TLS |
| <input type="checkbox"/> E <code>privat</code> est l'utilisateur du serveur qui a démarré le processus du démon web (<code>httpd</code>) |

Question 18 Que signifie le I de IP ?

- | | | | |
|--|--|--|---|
| <input type="checkbox"/> A Implicite | <input type="checkbox"/> C Information | <input type="checkbox"/> E Internet | <input type="checkbox"/> G Intranet |
| <input type="checkbox"/> B 1 (en chiffre romain) | <input type="checkbox"/> D Interface | <input type="checkbox"/> F Intermediaire | <input type="checkbox"/> H Infrastructure |

Question 19 ♣ Lesquelles des affirmations suivantes sur la cybersécurité sont vraies ?

- | |
|--|
| <input type="checkbox"/> A Pour héberger un serveur web, il faut acheter un certificat cryptographique |
| <input type="checkbox"/> B Les pages web en HTTPS garantissent la confidentialité du contenu des messages |
| <input type="checkbox"/> C <code>ssh</code> est une version sécurisée de <code>sh</code> (le shell POSIX) |
| <input type="checkbox"/> D Le chiffrement symétrique n'est pratiquement plus utilisé (pas assez sécuritaire) |
| <input type="checkbox"/> E Les pages web en HTTPS garantissent l'anonymat de l'adresse IP du client |

Question 20 ♣ À l'instar de la commande `nc`, la commande `socat`, que nous n'avons pas vue en classe, permet d'ouvrir et de se connecter aux sockets (voir l'annexe pour consulter le manuel expliquant son utilisation).

Sur la machine locale `ying` (d'adresse IP `132.208.138.246`), on exécute les commandes suivantes

```
echo toto | nc -l 8881 &
echo toto | nc -l 8882 &
echo tata | nc -l 8883 &
socat tcp-listen:8884 exec:cat &
socat tcp-listen:8885 exec:"nc yang 8886" &
```

Sur la machine distante `yang` (d'adresse IP `132.208.132.48`), on se connecte par SSH puis, on exécute les commandes suivantes

```
echo tata | nc -l 8881 &
echo otot | nc -l 8882 &
echo toto | nc -l 8883 &
socat tcp-listen:8884 exec:rev &
socat tcp-listen:8885 exec:"tr a o" &
socat tcp-listen:8886 exec:"nc localhost 8883" &
```

Parmi les commandes suivantes, lesquelles lorsqu'exécutées sur la machine locale `ying` affichent le message `toto` sur leurs sorties standard ?

- | | |
|---|--|
| <input type="checkbox"/> A <code>nc yang 8882</code> | <input type="checkbox"/> D <code>echo toto nc ying 8883</code> |
| <input type="checkbox"/> B <code>nc localhost 8885</code> | <input type="checkbox"/> E <code>echo otot nc 132.208.132.48 8884</code> |
| <input type="checkbox"/> C <code>nc localhost 8881</code> | |

Annexe détachable

cat Concatène des fichiers et les affiche sur la sortie standard. Si aucun fichier n'est fourni, ou si le fichier est « - », cat lit l'entrée standard.

echo Affiche une ligne de texte.

grep Recherche les lignes correspondant à un motif donné. Par défaut, grep affiche les lignes qui contiennent une correspondance au motif.

-E, --extended-regexp Interpréter le motif comme une expression régulière étendue (ERE).

-w, --word-regexp Sélectionner seulement les lignes contenant les correspondances qui forment des mots entiers. La sous-chaîne correspondante doit donc soit se trouver au début de la ligne, soit être précédée d'un caractère ne pouvant entrer dans la constitution d'un mot. De même, elle doit soit se trouver à la fin de la ligne, soit être suivie par un caractère ne pouvant entrer dans la constitution d'un mot. Les caractères composant les mots sont les lettres, les chiffres et le souligné « _ ».

-o, --only-matching Afficher seulement les parties (non vides) correspondantes d'une ligne correspondant, avec chaque partie correspondante sur une ligne séparée.

id Affiche les UIDs et GIDs effectifs et réels.

-G, --groups N'afficher que les groupes.

-n, --name Afficher les noms de groupe ou d'utilisateur à la place du numéro d'ID. Nécessite -u, -g, ou -G.

-u, --user N'afficher que l'UID.

jobs Affiche les tâches du shell courantes et leur état.

kill Envoie un signal à des processus. Les processus sont identifiés par leurs PID ou par des spécifications de tâches (préfixées par %). Si on ne précise pas de signal, TERM est envoyé, qui demande au processus de se terminer.

-9 Envoie le signal KILL (qui ne peut pas être intercepté par le processus).

-s Indique le signal à envoyer. Celui-ci peut être spécifié par son nom ou par son numéro.

ls Affiche le contenu de répertoires.

-l Utilise un format d'affichage long.

nc Établit des connexions TCP et UDP avec un hôte distant. Une fois la connexion établie, l'entrée standard est envoyée à l'hôte distant et les données reçues de l'hôte distant sont écrites sur la sortie standard.

-l Spécifie que nc doit écouter une connexion entrante. Un numéro de port local doit être fourni en argument. nc attend alors qu'un hôte distant se connecte.

Autrement une connexion à un hôte distant est établie. Un hôte distant et numéro de port distant doivent être fournis en argument. nc échoue si la connexion ne peut être établie.

pwd Affiche le chemin complet du répertoire courant.

ps Affiche un instantané des processus en cours.

-e Sélectionner tous les processus

-o Format défini par l'utilisateur. Le format est un argument unique sous la forme liste vide ou séparée par des

virgules, qui offre une façon de spécifier des colonnes de sortie individuelles. Des mots clés reconnus sont les suivants :

c (C) : Utilisation du processeur. Actuellement, c'est la valeur entière du pourcentage d'utilisation sur la durée de vie du processus.

cmd (CMD) : commande avec tous ses arguments sous forme de chaîne. Des modifications aux arguments peuvent être affichées. La sortie dans cette colonne peut contenir des espaces. Parfois, les arguments de processus ne seront pas disponibles ; lorsque cela se produit, ps affiche à la place le nom de l'exécutable entre parenthèses.

pid (PID) : Un nombre représentant l'ID de processus

ppid (PPID) : ID du processus parent

stime (STIME) : Heure du démarrage du processus

sz (SZ) : Taille en pages physiques de l'image mémoire du processus. Cela inclut le texte, les données et la pile

time (TIME) : Temps de calcul cumulé (utilisation du processeur) au format "[JJ-]HH:MM:SS".

tty (TTY) : Terminal qui contrôle le processus

uid (UID) : Identifiant de l'utilisateur

rev Inverse l'ordre des caractères de chaque lignes des fichiers ou de l'entrée standard.

sleep Attend une durée déterminée (en secondes).

socket Établit un relais multifonction (SOcket CAT). Socket est un utilitaire en ligne de commande qui établit deux flux d'octets bidirectionnels et transfère des données entre eux. Comme les flux peuvent être construits à partir d'un grand ensemble de types de puits et de sources de données et que de nombreuses options d'adresse peuvent être appliquées aux flux, socket peut être utilisé à différentes fins.

tcp-listen:port : Écoute sur port et accepte une connexion TCP. Notez que l'ouverture de cette adresse bloque jusqu'à ce qu'un client se connecte.

exec:commande : commande est une commande simple avec des arguments séparés par des espaces simples. Après avoir exécuté la commande avec succès, socket écrit les données sur l'entrée standard du processus et lit à partir de sa sortie standard.

tac Concatène et affiche des fichiers à l'envers (la dernière ligne en premier).

tail Affiche la dernière partie de fichiers.

-n, --lines=K afficher les K dernières lignes, au lieu des 10 dernières.

tr Convertit ou élimine des caractères de l'entrée standard. Par défaut, tr convertit le caractère du premier ensemble de caractères vers le caractère correspondant dans le second ensemble de caractères. Les ensembles sont indiqués sous forme de chaînes de caractères. La plupart se représentent eux-mêmes ; autrement, des séquences interprétées sont :

CAR1-CAR2 : tous les caractères de CAR1 à CAR2, dans l'ordre croissant

\\ : contre-oblique (« backslash »)



0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9

← Codez les 8 chiffres de votre code permanent ci-contre, et inscrivez-le à nouveau ci-dessous avec vos nom et prénom.

Code permanent :

 Prénom :

 Nom :

Les réponses aux questions sont à donner exclusivement sur cette feuille : les réponses données sur les feuilles précédentes ne seront pas prises en compte.

Important : noircissez complètement l'intérieur de chaque case (pas de croix, pas de cercles).

- Question 1 : A B C D E F
- Question 2 : A B C D E F G H
- Question 3 : A B C D E F
- Question 4 : A B C D E F
- Question 5 : A B C D E F G H
- Question 6 : A B C D E F
- Question 7 : A B C D E F
- Question 8 : A B C D E F
- Question 9 : A B C D E F G H
- Question 10 : A B C D E F
- Question 11 : A B C D E F
- Question 12 : A B C D E F G H I J
- Question 13 : A B C D E F
- Question 14 : A B C D E F G H
- Question 15 : A B C D E F
- Question 16 : A B C D E F
- Question 17 : A B C D E
- Question 18 : A B C D E F G H
- Question 19 : A B C D E
- Question 20 : A B C D E