

CSC3102 — Contrôle Final 2

Année 2025 – 2026 (corrigé)

Consignes :

- répondez aux questions en écrivant :
 - soit dans les cadres fournis au fil du sujet,
 - soit aux endroits marqués des numéros de questions dans les *listings* en fin de sujet ;
- si vous manquez de place, vous pouvez écrire sur la dernière page en l'indiquant clairement ;
- polycopié, notes personnelles manuscrites et exemples de scripts sont autorisés ;
- la durée est fixée à 1h30 ;
- le barème est donné à titre indicatif.

Petite Souris & Cie.

La Petite Souris est impressionnée par l'efficacité économique de la Fée des Dents, son homologue américaine. Elle a donc décidé de développer son activité d'acquisition de dents de lait à domicile.

Elle a commencé par recruter deux collectrices qui travaillaient auparavant pour la Fée des Dents. Celles-ci ont donc rejoint sa nouvelle entreprise, « Petite Souris & Cie. » Elle pourra ainsi collecter les dents plus efficacement, tout en restant tranquillement dans son terrier.

Vous trouverez à la fin du sujet les ébauches du script **petite-souris.sh**, qui simule les opérations de direction de la Petite Souris ; et du script **collectrice.sh**, qui simule le travail d'une collectrice. Vous y trouverez aussi le script déjà implémenté **prendre-sou.sh**, qui sera utilisé à la question (5).

[4 pt] Travail des collectrices : validation de la présence de la dent

Complétons le script **collectrice.sh**, qui simule le travail d'une collectrice au service de la Petite Souris. Ce script reçoit un unique argument : le chemin vers un fichier qui contient la liste des oreillers à visiter (préparé par la Petite Souris aux questions (10) à (13), un chemin d'oreiller par ligne).

Le script consiste en une boucle pour traiter chaque oreiller de la liste. Cette boucle doit :

1. vérifier que la dent n'a pas été volée par la Fée des Dents (question (4)) ;
2. prendre un sou de la tirelire (questions (5) et (6)) ;
3. collecter la dent (question (7)) ;
4. rapporter la dent (question (8)).

(1) [1,5 pt] Au début du script, vérifiez que celui-ci a reçu un seul argument. Si ce n'est pas le cas, affichez un message d'erreur et terminez le script en erreur.

(2) [1 pt] Complétez la structure de la boucle (son corps sera le sujet des questions suivantes) pour lire ligne par ligne le fichier dont le chemin a été reçu en argument. La ligne lue doit être stockée dans une variable nommée `oreiller`, qui contiendra donc le chemin vers le répertoire de l'oreiller à visiter.

Cependant, il est possible qu'il n'y ait pas de dent sous l'oreiller ! En effet, la Fée des Dents est jalouse que la Petite Souris ait embauché ses collectrices ; en représailles, elle lui vole les dents. La présence d'une dent est matérialisée par l'existence d'un fichier normal nommé `dent` dans le dossier de l'oreiller. S'il n'y a pas de dent, la collectrice doit le signaler à la Petite Souris.

(3) [0,5 pt] Complétez l'en-tête de la condition pour tester l'**absence** du fichier dent : celui-ci s'appelle dent, et son chemin est déjà construit dans la variable `fichier_dent`.

(4) [1 pt] Dans la branche de la condition où le fichier dent n'existe pas, envoyez un signal de votre choix au processus parent du processus de la collectrice. Ce signal sera géré à la question (17).

[5 pt] Travail des collectrices : collection de la dent

Traitons maintenant le cas attendu où il y a bien une dent sous l'oreiller. Avant de partir collecter la dent, la collectrice doit obtenir un sou à déposer sous l'oreiller à la place de celle-ci. Cette opération est déjà écrite dans le script **prendre-sou.sh** (voir en fin de sujet). Remarquez que celui-ci indique par son code de retour, s'il a été possible de prendre un sou :

- code de retour d'erreur : plus aucun sou disponible dans la tirelire ;
- code de retour de succès : la collectrice a pu prendre un sou.

(5) [1 pt] À l'endroit approprié dans **collectrice.sh**, appelez le script **prendre-sou.sh** et conservez son code de retour dans une variable.

(6) [1,5 pt] Complétez la condition pour vérifier si le code de retour de l'appel à **prendre-sou.sh** signale une erreur. Puis complétez son corps. Dans le cas d'une erreur, la collectrice doit :

1. écrire un message sur le tapis roulant (mis en place à la question (14)), c'est-à-dire dans le tube nommé `tapis-roulant.tube`, qui a la forme suivante : `<PID> retour plus d'argent !`, où `<PID>` est le PID du processus simulant la collectrice ;
2. terminer avec un code d'erreur.

À ce stade, la collectrice a confirmé la présence d'une dent, et a un sou pour l'échanger avec la dent. Elle peut donc se déplacer jusqu'à l'oreiller, remplacer la dent par le sou, et la ramener au terrier.

(7) [1,5 pt] Réalisez l'échange de la dent avec le sou :

1. changez de répertoire courant pour placer le processus de la collectrice dans le dossier oreiller ;
2. lisez le contenu du fichier dent (une seule ligne) pour le conserver pour la question suivante (8) ;
3. écrivez le mot « sou » dans le fichier dent.

(8) [0,5 pt] Ramenez la dent au terrier et déposez-la sur le tapis roulant :

1. changez de répertoire pour placer le processus de la collectrice dans son répertoire de connexion ;
2. écrivez le message suivant dans le tube du tapis roulant (nommé `tapis-roulant.tube`, comme à la question (6)) : `<PID> dent <DENT>`, où `<PID>` est le PID du processus simulant la collectrice, et `<DENT>` est la dent collectée, c'est-à-dire le contenu du fichier dent lu à la question (7).

(9) [0,5 pt] Enfin, la collectrice pourra aussi finir par avoir visité tous ses oreillers. Écrivez le message suivant dans le tube du tapis roulant (nommé `tapis-roulant.tube`, comme à la question (6)) : `<PID> retour fin de la collecte`, où `<PID>` est le PID du processus simulant la collectrice.

[4,5 pt] Préparation du plan de collecte

Les collectrices sont prêtes à travailler, mais il leur manque un plan de collecte !

La zone de collecte de dents de la Petite Souris est représentée par une arborescence de dossiers. Dans le dossier Europe se trouvent les dossiers des pays France, Belgique, Luxembourg et Suisse. Dans chaque dossier de pays, les villes sont représentées par des dossiers `ville-<N>`, où `<N>` est un numéro identifiant la ville. Enfin, dans chaque dossier de ville se trouvent des dossiers oreillers nommés `<M>.oreiller`, où `<M>` est un numéro identifiant un oreiller sous lequel devrait se trouver une dent.

Afin de préparer la collecte, la Petite Souris constitue une liste des oreillers visitables. Les réponses à cette partie sont à écrire dans le script **petite-souris.sh**.

(10) [1 pt] Écrivez une commande qui trouve, de manière récursive dans le dossier Europe, tous les dossiers oreillers. La sortie de cette commande doit être écrite dans un fichier nommé `oreillers.lst`.

Il faut ensuite répartir équitablement les oreillers entre les deux collectrices. La première étape consiste à obtenir le nombre total d'oreillers, en utilisant la bonne invocation de la commande `wc`. **Attention**, la sortie de `wc` donne d'abord le nombre compté, et ensuite le nom du fichier traité ; ces deux champs sont séparés par une espace. Vous devez donc aussi utiliser la commande `cut` pour extraire uniquement le nombre de lignes.

(11) [1,5 pt] Écrivez **une suite de commandes** qui récupère le nombre de lignes dans `oreillers.lst`, et le stocke dans une variable.

(12) [0,5 pt] Calculez la moitié du nombre de lignes dans une autre variable, en admettant qu'il y en a un nombre pair.

(13) [0,5 pt] On rappelle que la commande `head -nN fichier` affiche les `N` premières lignes de fichier, et que la commande `tail -nN fichier` en affiche les `N` dernières. Écrivez dans le fichier `oreillers-collectrice1.lst` les chemins vers les oreillers que la collectrice 1 va visiter, et de même dans `oreillers-collectrice2.lst` pour la collectrice 2.

Enfin, les collectrices vont rapporter les dents, et informer la Petite Souris, en déposant des messages sur un tapis roulant. La Petite Souris doit préparer ce tapis roulant : il est matérialisé par un tube nommé `tapis-roulant.tube`.

(14) [1 pt] Créez un tube nommé `tapis-roulant.tube`, et maintenez-le ouvert en lecture / écriture grâce à une redirection avancée.

[6,5 pt] Lancement de la collecte

Les collectrices étant prêtes à travailler, et leurs plans de collecte étant établis, il est temps pour la Petite Souris de les envoyer au travail.

(15) [0,5 pt] Lancez deux processus en arrière-plan qui exécutent chacun le script **collectrice.sh** (on suppose le script déjà exécutable). Ils doivent recevoir en argument les noms des fichiers contenant leurs listes respectives d'oreillers à visiter.

(16) [2 pt] En prenant en compte le comportement des collectrices et de la Petite Souris, pensez-vous que le lancement en arrière-plan des deux collectrices provoque des problèmes de concurrence ?

Justifiez votre réponse dans tous les cas, avec un exemple de scénario problématique si nécessaire. S'il y a un problème, protégez la ou les sections critiques identifiées dans les scripts à l'aide des scripts **P.sh** et **V.sh**, vus en TP et considérés comme disponibles dans le répertoire courant. Prenez soin de protéger le plus finement possible. Faites précéder vos ajouts de la balise **(16)** afin de les identifier.

Problème de concurrence sur tirelire entre les collectrices qui pourraient écraser le décrétement de l'autre.

Même si la Petite Souris ne fait plus la collecte elle-même, elle suit le travail de ses collectrices par plusieurs biais, que vous avez déjà mis en place aux questions **(4)**, **(6)**, **(8)** et **(9)**.

(17) [1,5 pt] Une collectrice peut signaler à la Petite Souris qu'un vol de dent a eu lieu (mis en place à la question **(4)**). Écrivez à l'endroit adéquat (non indiqué dans les *listings*) le gestionnaire de ce signal. Celui-ci doit seulement afficher un message indiquant qu'un vol a eu lieu.

(18) [1 pt] Au début du corps de la boucle de suivi, lisez un message depuis le tube du tapis roulant (mis en place à la question **(14)**). Le message suit le format général <PID> <TYPE> <RESTE DU MESSAGE> (vu dans les questions **(6)**, **(8)**, et **(9)**), où :

- <PID> est le PID du processus de la collectrice qui a écrit le message ;
- <TYPE> est le type de message : soit la valeur retour, soit la valeur dent ;
- <RESTE DU MESSAGE> est tout le reste du message.

Conservez <PID>, <TYPE> et <RESTE DU MESSAGE> dans des variables distinctes.

(19) [0,5 pt] Enfin, traitez les messages selon le cas :

- si le type est retour, incrémentez la variable \$terminees puis affichez une ligne indiquant le PID de la collectrice et la raison de son retour (c'est-à-dire le reste du message) ;
- si le type est dent, affichez une ligne indiquant le PID de la collectrice et la dent ramenée (c'est-à-dire le reste du message).

(20) [1 pt] La Petite Souris souhaite agir contre le problème du vol de dents par la Fée des Dents (représentée par l'utilisateur fée-des-dents sur le système). Selon elle, le problème est que l'utilisateur fée-des-dents, qui n'est pas dans le même groupe que les collectrices, peut supprimer des fichiers dans les dossiers des oreillers. La Petite Souris ordonne donc à ses collectrices de changer les droits sur les dossiers oreillers qu'elles visitent pour corriger ce problème de sécurité. Écrivez ci-dessous la commande qu'une collectrice doit exécuter pour protéger un dossier oreiller lorsqu'elle le visite.

```
chmod o-w "$PWD"
```

collectrice.sh

```
#!/usr/bin/bash

#(1)
if [ $# -ne 1 ]; then
    echo "Usage: $0 LISTE_OREILLERS_FP"
    exit 2
fi

liste_oreillers="$1"

#(2)
while read oreiller; do

    fichier_dent="$oreiller/dent"

    #(3)
    if ! [ -f "$fichier_dent" ]; then

        #(4) le fichier $fichier_dent n'existe pas
        kill -USR1 $$PPID

    else

        #(16)
        ./P.sh tirelire.lock
        #(5)
        ./prendre-sou.sh
        ret=$?
        #(16)
        ./V.sh tirelire.lock
        #(6) erreur en prenant un sou ?
        if [ "$ret" -ne 0 ]; then
            echo "$$ retour plus d'argent !" >tapis-roulant.tube
            exit 1
        fi

        #(7)
        cd "$oreiller"
        dent="$(cat dent)"
        echo "sou" >dent

        #(8)
        cd "$HOME"
        echo "$$ dent $dent" >tapis-roulant.tube

    fi
done <"$liste_oreillers"

#(9)
echo "$$ retour fin de collecte" >tapis-roulant.tube

exit 0
```

petite-souris.sh

```
#!/usr/bin/bash

terminees=0

#(10)
trap "echo \"Au voleur ! au voleur ! à l'assassin !\"" USR1

#(11)
find Europe -type d -name "*.oreiller" >oreillers.lst

#(12)
nb="$(wc -l oreillers.lst | cut -d" " -f1)"

#(13)
nb_oreillers="$(expr "$nb" / 2)"

head -n "$nb_oreillers" oreillers.lst >oreillers-collectrice1.lst
tail -n "$nb_oreillers" oreillers.lst >oreillers-collectrice2.lst

#(14)
mkfifo tapis-roulant.tube
exec 3<>tapis-roulant.tube

#(15)
./collectrice.sh oreillers-collectrice1.lst &
./collectrice.sh oreillers-collectrice2.lst &

while [ "$terminees" -ne 2 ]; do
    #(18)
    read collectrice type msg <&3

    #(19)
    case "$type" in
        "retour")
            terminees="$(expr "$terminees" + 1)"
            echo "La collectrice $collectrice rentre : $msg"
            ;;
        "dent")
            echo "La collectrice $collectrice a ramené une dent : $msg"
            ;;
        esac
done

rm tapis-roulant.tube
```

prendre-sou.sh

```
#!/usr/bin/bash
```

```
read capital <tirelire
```

```
if [ "$capital" -le 0 ]; then
```

```
    exit 1
```

```
fi
```

```
capital="$(expr "$capital" - 1)"
```

```
echo "$capital" >tirelire
```

```
exit 0
```