



Outils de développement logiciel

TP n° 3 : écrire des scripts shell

Alain Lebreton

2023-2024

Objectif

Concevoir des scripts **shell** bash.

Pré-requis : chapitre 7

Durée estimée : 2 séances

1 Un script peu parlant

Exercice n° 1

1. Déplacez-vous dans votre dossier "odl" et créez le sous-dossier "tp03", puis placez-vous dans ce dernier. "tp03" sera votre dossier de travail pour cette séance et vous y effectuerez toutes vos actions.
2. Copiez dans le dossier de travail le fichier "Peukaloinen.sh" que vous trouverez dans le dossier "ressources-odl-fise/tp03/" et donnez-lui des droits en exécution, puis lancez-le. Que se passe-t-il ?
3. Recommencez en lançant cette fois-ci la commande : "source ./Peukaloinen.sh". Que se passe-t-il ?
4. Réécrivez le script de manière à ce qu'il devienne compréhensible pour toute personne pratiquant l'anglais (noms de variables, de fonctions, commentaires, etc.).

2 Chiffrement par décalage simple

Le chiffre de César est une méthode de chiffrement par décalage très simple qui fut utilisée par Jules César dans ses correspondances secrètes. Le texte chiffré s'obtient en remplaçant par une permutation circulaire de l'alphabet, chaque lettre du texte original par une lettre à distance fixe (3 dans le cas du chiffre de César).

Exercice n° 2

1. Dans votre dossier de travail, créez le sous-dossier "encryption", puis placez-vous dedans.
2. Proposez le script "encrypt.sh" qui utilise le chiffrement de César sur un fichier texte passé en paramètre au script (on ne se préoccupera pas du chiffrement des nombres et des caractères accentués). Le script sera appelé de la manière suivante (on pourra à bon escient utiliser la commande Unix tr) :

```
$ encrypt.sh file_to_encrypt encrypted_file
```

Exercice n° 3

Toujours dans le dossier "encryption", proposez le script "decrypt.sh" qui réalise le déchiffrement d'un fichier chiffré à l'aide du script précédent. L'appel se fera de la manière suivante :

```
$ decrypt.sh encrypted_file decrypted_file
```

3 Un incrustateur de filigrane

ImageMagick est une bibliothèque de commandes permettant de réaliser des opérations diverses sur les images (conversion, opérations géométriques, etc.). Elle dispose notamment de la commande `composite` qui permet entre autres d'insérer un filigrane sur une image.

Exercice n° 4

1. Dans votre dossier de travail, créez le sous-dossier "watermark", puis placez-vous dedans.
2. Proposez le script "wm.sh" qui met en oeuvre la commande `composite` afin d'insérer un filigrane sur l'image "im.xxx" passée en paramètre et de produire en sortie l'image "im-wm.xxx". Le script prendra en compte le texte du filigrane, sa taille et son opacité (0-100%), et il sera appelé de la manière suivante :

```
$ wm.sh <image> <watermark_text> <opacity> <size_of_watermark>
```

4 Statistiques d'utilisation

Il s'agit de construire un programme de visualisation, sous la forme d'un histogramme, du taux horaire d'utilisation d'un ordinateur. L'histogramme sera affiché par l'intermédiaire du navigateur Firefox sous la forme d'une image (telle que celle ci-dessous) dans une page HTML.

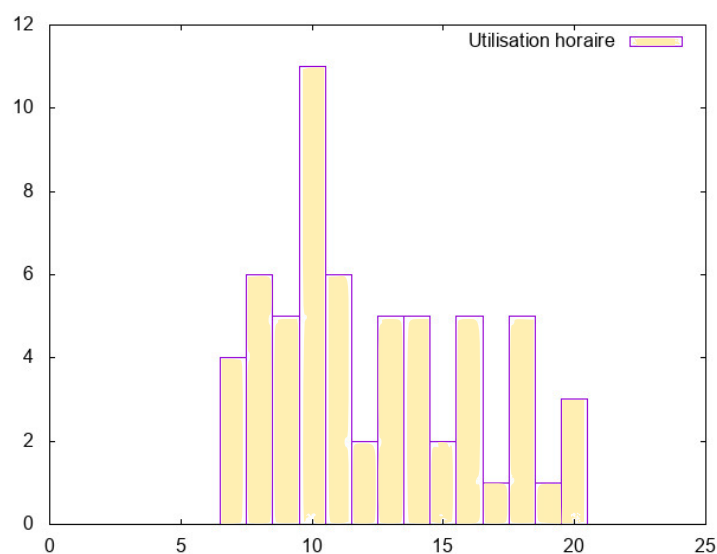


Figure 1: Histogramme d'utilisation d'un ordinateur

Préalable

La commande `last` permet d'afficher sur la sortie standard les informations sur les dernières connexions à la machine courante :

```
jsaigne pts/0      10.213.11.61      Mon Oct  2 16:10    still logged in
jcepte  pts/0      10.214.12.62      Mon Oct  2 13:44 - 16:06 (02:22)
alamb   pts/0      193.50.79.252     Mon Oct  2 11:03 - 13:18 (02:15)
jcepte  pts/1      10.214.12.62      Mon Oct  2 08:39 - 08:43 (00:04)
jsaigne pts/1      193.57.121.61     Mon Oct  2 08:22 - 08:23 (00:00)
alamb   pts/0      193.50.79.252     Mon Oct  2 07:32 - 10:17 (02:44)
rosair  pts/4      86.237.235.208    Sun Oct  1 17:45 - 18:25 (00:39)
bduteil pts/4      176.159.152.27    Sun Oct  1 17:42 - 17:42 (00:00)
rumeur  pts/5      193.50.79.252     Sun Oct  1 17:06 - 20:54 (03:47)
rumeur  pts/0      193.50.79.252     Sun Oct  1 17:01 - 21:29 (04:28)
ulrich  pts/4      86.241.19.68      Sun Oct  1 16:39 - 17:39 (00:59)
```

Afin d'accéder aux heures de connexion, il suffit de récupérer les valeurs après la 50e colonne, ou encore celles du 7e champ.

Exercice n° 5

1. Dans votre dossier de travail, créez le sous-dossier "histogram" puis placez-vous dedans.

2. Proposez le script "histo.sh" permettant d'afficher dans le navigateur Firefox le taux horaire d'utilisation de l'ordinateur sous la forme d'un histogramme. Ce script enchaînera les opérations suivantes :

- extraire les heures de connexions à partir de la commande `last` ;
- calculer le nombre de connexions pour chaque heure et produire les données à destination de Gnuplot sous la forme de lignes "***x y***", où ***x*** correspond à l'heure et ***y*** au nombre de connexions (ces données seront stockées dans un fichier) ;
- mettre en oeuvre `gnuplot` qui construit une image au format JPEG à partir des données et utilise un fichier de commandes qui sera construit automatiquement (aidez-vous du fichier "commands.gp" qui vous est fourni) ;
- construire automatiquement une page HTML qui contient au moins la balise `` pour l'affichage de l'image ;
- lancer Firefox en mode hors ligne avec le fichier HTML comme argument (Firefox accepte l'option `-new-window` qui permet d'ouvrir une nouvelle fenêtre si Firefox est déjà lancé) ;
- supprimer tous les fichiers intermédiaires créés, excepté le fichier HTML et l'image.

Vous aurez peut-être besoin des commandes `cut`, `grep`, `tr`, `sort`, `uniq` et `printf`, ainsi que de `tubes`.

5 Un générateur de squelette (facultatif)

Exercice n° 6

Dans votre dossier de travail, initiez un script "create_project.sh" qui vous permettra à terme de faciliter la gestion de vos TP et projets en langage C. Le script affiche au démarrage le menu suivant (ce menu sera affiché en permanence en haut du terminal) :

```
Générateur de squelette d'un projet C
=====
N|n : créer un nouveau projet
O|o : ouvrir un projet
S|s : supprimer le projet
C|c : compiler le projet
A|a : archiver le projet
Q|q : quitter
```

Chaque opération du menu sera réalisée par une fonction. Dans un premier temps, vous vous limiterez aux opérations suivantes :

1. "Créer un nouveau projet" : génère à partir du nom de dossier indiqué par l'utilisateur (de la forme "tpXX" pour les TP ODL), l'arborescence d'une distribution logicielle telle qu'elle est décrite à la section 12.2 du polycopié de cours. Les fichiers d'identification et de documentation pourront être créés avec un minimum d'informations telles que celles qui vous concernent en tant qu'étudiant de l'ENSICAEN (vous pouvez aussi mettre en place un système de fichiers "motifs" qui seraient faciles à modifier par un utilisateur ne connaissant que peu le langage **shell**. Les fichiers sources ne seront pas créés, excepté le fichier "Makefile" qui pourra l'être, même s'il est fourni vide pour le moment.
2. "Ouvrir un projet" : se place à la racine du projet courant (celui qui vient d'être créé) ou celui qui est indiqué par l'utilisateur si c'est la première opération choisie par lui lorsqu'il exécute le script.
3. "Supprimer le projet" : supprime l'ensemble des dossiers et fichiers du projet courant.
4. "Archiver le projet" : produit une archive compressée du projet courant sous le nom : "rapport-tpXX-jj-mm-aaaa-prenom_nom.tar.gz".

6 Édition du rapport

Pour ce TP, vous remettrez un rapport en anglais au format Markdown dans lequel vous inclurez le code de vos scripts que vous expliquerez en détail.

Quelques mots sur le langage Markdown

Le langage Markdown est un langage de balisage très léger et à la syntaxe très simple. Il permet une mise en forme plus agréable qu'un fichier texte tout en restant "lisible" par rapport à un fichier HTML ou LaTeX. Il a tendance à devenir la norme dans l'écriture des fichiers de présentation des applications "open source" (les fichiers README), et de nombreux visualiseurs sont à présent disponibles sur tous les systèmes. Le fichier "exemple_markdown.md" disponible dans le dossier "ressources-odl-fise/tp03" devrait suffire pour vous aider à réaliser vos différents rapports ¹. Vous pouvez le visualiser en utilisant une application adaptée, ou encore à l'aide de l'excellent convertisseur "Pandoc" et du navigateur textuel "Lynx" :

¹Ce document, de même que le polycopié de cours, sont d'ailleurs totalement rédigés en Markdown !

```
$ pandoc -f markdown_github exemple_markdown.md | lynx -stdin
```

7 Livrable

Avant la date limite, déposez sur la plateforme Moodle le fichier "rapport-tp03-nom1-nom2.md" qui contiendra votre compte-rendu avec le code de vos scripts.

8 Résumé

Dans ce TP, vous avez réalisé quelques scripts en ***shell bash***.

Nous espérons qu'à partir de maintenant, vous penserez à les utiliser afin d'automatiser vos différentes tâches. D'ailleurs, pourquoi ne pas vous faire un script permettant de générer l'arborescence complète ainsi que les fichiers de base qui sont attendus en première année.