

Mise en place

Dans ce TP vous :

- devrez travailler dans un dossier nommé **asr/binaire** dans votre répertoire principal;
- allez visualiser et manipuler des informations représentées en binaire. Ces informations, stockées sur le disque dur de votre machine, sont accessibles dans des fichiers.

Il existe deux commandes utiles pour manipuler le contenu des fichiers comme une simple représentation binaire :

- **od** est une commande permettant d'afficher les données contenues dans des fichiers sous forme de nombres représentés en octal ou sous d'autres formes;
- **hexedit** est une commande permettant de voir **et** de modifier interactivement le contenu d'un fichier sous forme hexadécimale ou ASCII.

Q 1. Créez le répertoire de travail et déplacez-y vous.

Exercice 1 : Prise en main des outils

Q 1. Déterminez grâce à la commande **stat** la taille du fichier `/etc/debian_version`.

Q 2. Lisez la documentation de **od** (au moins la page du manuel).

Q 3. Par défaut, c'est-à-dire sans arguments autre que le nom du fichier à étudier, et sachant que **sizeof(short)** vaut 2 :

- sous quelle forme est représenté l'offset?
- sous quelle forme sont représentés les nombres?
- combien d'octets représentent chaque nombre?
- combien **od** affiche-t-il d'octets par ligne?

Q 4. Le premier nombre affiché sur chaque ligne est l'*offset*. *Offset* signifie « **décalage** » en français. De quel décalage s'agit-il?

Q 5. Déterminez lignes de commandes permettant :

- de voir le contenu de `/etc/debian_version` sous forme de nombres **hexadécimaux** représentant **un octet chacun**
- de voir le contenu de `/etc/debian_version` sous forme de nombres **décimaux** représentant **un octet chacun**
- de voir le contenu de `/etc/debian_version` **un octet par ligne**

Q 6. Combien d'octets contient le fichier `/etc/debian_version`?

Q 7. Est-ce cohérent avec la taille que vous avez déterminée lors de la première question?

Q 8. Après avoir lu la documentation de **hexedit** décrivez comment :

- quitter **hexedit**?
- aller directement à la fin du fichier?
- basculer le curseur de la représentation hexadécimale à la représentation ASCII?
- sauvegarder les modifications faites?
- sauvegarder puis quitter **hexedit** en une seule action?

Exercice 2 : Modifier le contenu d'un fichier binaire

Q 1. Copier le fichier `/home/public/m1101/tp/binaire/Secret.class` dans votre dossier de travail¹.

Ce fichier est un programme écrit en Java dont on ne vous donne que la partie exécutable. Dans ce fichier est stocké un mot secret. Vous pouvez essayer de découvrir ce mot en exécutant le programme avec le mot que vous voulez tester. Par exemple pour tester le mot **toto**, vous devez saisir la ligne de commande suivante :

```
java Secret toto
```

1. Pour mémoire ce doit être `~/asr/binaire` qui doit d'ailleurs être votre dossier courant

Si le mot que vous avez testé est le bon alors le programme affiche le mot **Bravo**, sinon il affiche le mot **Perdu**.

Q 2. Essayez effectivement le mot **toto** puis quelques autres mots de votre choix.

Q 3. En analysant le fichier `Secret.class` avec `hexedit` ou `od`, essayez de déterminer quel est ce mot secret.

Q 4. Modifiez le fichier `Secret.class` de façon à ce que le mot secret, qui déclenche l’affichage de **Bravo**, soit désormais le mot `public`.

Q 5. Vérifiez que votre modification est effective.

Exercice 3 : Bonus - Comprendre le stockage des octets dans les fichiers et en mémoire

Q 1. Copier le fichier `/home/public/m1101/tp/binaire/donnees.bin` dans votre dossier de travail.

Du fait de leur architecture, les ordinateurs que vous utilisez stockent les données en respectant la convention *little endian*. En mémoire et pour des nombres représentés sur plus d’un octet, les octets de poids faibles sont ainsi stockés en premier.

Par ailleurs la taille des nombres entiers courts (`short int`), définie par le langage C et donc utilisée par les commandes écrites avec ce langage (comme `od` ou `hexedit`), est de 2 octets.

Par défaut quand `od` lit un `short int` d’un fichier vers la mémoire, il lit donc en premier l’octet de poids faible **puis** l’octet de poids fort. Ensuite, quand il affiche un tel nombre et comme le spécifie l’option par défaut `-t oS`, il affiche donc un entier, représenté sous forme octale, codé sur 2 octets et dont l’octet de poids faible représente le premier octet du fichier et l’octet de poids fort le second octet du fichier.

Q 2. Notez les 4 premiers nombres (c’est-à-dire une représentation des 8 premiers octets) affichées par `od donnees.bin`.

Q 3. Convertissez **correctement** chaque **octet** lu en hexadécimal.

Q 4. Trouvez la ligne de commande `od` qui affiche le résultat escompté exact permettant de vérifier votre conversion.