

Pour ce TP tous les fichiers que vous allez créer devront être placés dans un dossier nommé `asr/tp03` de votre répertoire principal.

### Exercice 1 : Prise en main des outils

Dans cette séance vous allez visualiser et manipuler des informations représentées en binaire. Ces informations, stockées sur le disque dur de votre machine, sont donc accessibles via 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.** La commande `stat` permet d'obtenir les caractéristiques d'un fichier, dont sa taille. Déterminez grâce à elle la taille du fichier `/etc/debian_version`.

**Q 2.** Après avoir lu la page du manuel de `od` et la page `info` de `od`, donnez une ligne de commande permettant :

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

**Q 3.** Selon vous, à quoi correspond le premier nombre affiché par `od` sur chaque ligne ?

**Q 4.** Que représente l'*offset* dont il est fait allusion dans la page du manuel de `od` ?

**Q 5.** Par défaut, c'est-à-dire sans arguments autre que le nom du fichier à étudier,

- combien `od` affiche-t-il d'octets par ligne ?
- sous quel forme est représenté l'offset ?

**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 page du manuel 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 : Visualiser le contenu binaire d'un fichier

Pour faire cet exercice vous avez besoin de récupérer le fichier `donnees.bin` disponible avec l'énoncé sur le cours en ligne et de le placer dans votre dossier `asr/tp03`.

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.

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 1.** Notez les 4 premiers nombres (c'est-à-dire une représentation des 8 premiers octets) affichées par `od` pour le fichier `donnees.bin`.

**Q 2.** Convertissez correctement chaque octet lu en hexadécimal.

**Q 3.** Trouvez la ligne de commande `od` qui affiche le résultat escompté exact permettant de vérifier votre conversion.

### **Exercice 3 : Modifier le contenu d'un fichier**

Pour faire cet exercice vous avez besoin de récupérer le fichier `Secret.class` disponible avec l'énoncé sur le cours en ligne et de le placer dans votre dossier `asr/tp03`. Ce fichier est également 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
```

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

**Q 1.** Essayez effectivement le mot `toto` puis quelques autres mots de votre choix.

**Q 2.** En analysant le fichier `Secret.class` avec `hexedit` ou `od`, essayez de déterminer quel est ce mot secret.

**Q 3.** 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 4.** Vérifiez que votre modification est effective.

### **Exercice 4 : Jouons avec les encodages**

**Q 1.** Dans un terminal, placez-vous dans le dossier `asr/tp03` si ça n'est pas déjà le cas.

**Q 2.** Dans un terminal, entrez la commande `export LANG=C.UTF-8`, puis saisissez, avec `emacs` et dans un fichier nommé `texte.utf8`, la ligne suivante « ABCé€ ».

**Q 3.** Déterminez l'option de `wc` permettant de compter les octets d'un fichier.

**Q 4.** Déterminez l'option de `wc` permettant de compter les caractères d'un fichier.

**Q 5.** Déterminez la taille du fichier `texte.utf8` en octets puis en caractères. Sont-elles identiques?

**Q 6.** Utilisez la commande `od` pour afficher le contenu binaire de ce fichier (sous la forme qui vous est la plus agréable) et déterminer les octets codant le caractère « é ».

**Q 7.** Lisez la page `iconv(1)` puis créez deux fichiers `texte.utf16` et `texte.utf32` à partir du contenu du fichier `texte.utf8`

**Q 8.** En consultant `man iconv`, décrivez ce que réalise la ligne de commande suivante :

```
iconv texte.utf8 -t UTF-16 -o texte.utf16
```

**Q 9.** Détruisez le fichier `texte.utf8`.

**Q 10.** Quelle ligne de commande permet la transformation inverse (obtenir le fichier `texte.utf8` à partir du fichier `texte.utf16`)?

**Q 11.** Créez de la même façon le même contenu encodé en UTF-32 (le fichier s'appellera `texte.utf32`).

**Q 12.** Étudiez le binaire du fichier `texte.utf16` puis indiquez selon vous quelle est l'endiannes du fichier.