



# TP - Introduction à l'environnement Unix/Linux

F2B002E - Harmonisation Mastères

Automne 2016

## 1 Découverte de l'interface graphique

La distribution de Linux utilisée est une Ubuntu. L'interface graphique est basée sur le standard Gnome. Cette interface graphique est hautement configurable, par les utilisateurs, mais également par les éditeurs de distribution Linux. Mais si l'aspect visuel peut changer d'une version de Linux à l'autre, le fonctionnement général restera le même (certains spécialistes avancent le terme d'*interface intuitive...*).<sup>1</sup>

Une fois que vous avez ouvert votre session graphique, vous avez en haut de l'écran une barre de menus et en bas une barre de tâches. Techniquement, qu'elles soient en haut, en bas, ou sur les côtés, ces barres sont identiques. Elles ont juste pour rôle de contenir d'autres éléments : typiquement en haut des menus déroulants et des icônes de raccourcis, en bas la liste des tâches. (Vous pourrez vous amuser à ajouter d'autres éléments, en supprimer, en déplacer, mais vous ferez ça à un autre moment...)

Entre ces deux barres, au centre de l'écran, vous trouverez votre *bureau*, là où s'afficheront les fenêtres de vos applications.

### 1.1 Votre espace disque dans le système complet

Cliquez sur le menu *Raccourcis*, puis choisissez votre *Dossier Personnel* (i.e. tous les fichiers de votre espace disque). Vous devez voir apparaître une nouvelle fenêtre : l'explorateur de fichiers Nautilus<sup>2</sup>.

Cet explorateur vous permet de visualiser *vos fichiers* et *vos dossiers* (que l'on appelle des *répertoires* en informatique scientifique).

La barre du haut de Nautilus comporte :

- au centre : l'emplacement, là où se trouvent les fichiers listés dans la fenêtre principale ;
- à gauche de l'emplacement : deux flèches pour évoluer dans la hiérarchie des emplacements ;
- à droite de l'emplacement : trois boutons pour paramétrer le mode d'affichage (en liste, en damier, ou d'autres options) ;
- et dans le coin tout à droite : l'accès au menu de Nautilus.

Créons un premier fichier.

Dans la fenêtre principale, visez une zone où il n'y a rien, et cliquez avec le bouton droit. Un menu contextuel apparaît. Faites **Nouveau document**.<sup>3</sup> Par exemple, appelez votre nouveau document **essai1**. (Vous pouvez également passer par le menu principal de Nautilus, en cliquant sur le bouton en haut à droite.) Faites **Nouveau document** > **Document vide**. Constatez le résultat dans l'explorateur.

Sélectionnez le fichier dans l'explorateur par un clic de souris avec le bouton droit. Sélectionnez **Propriétés**. Répondez aux questions suivantes :

- Qui est le propriétaire de ce fichier (donnez le «nom informatique» exact) ?
- Ce fichier appartient aussi à un groupe d'utilisateurs, quel est ce groupe ?
- Quelles sont les permissions (on dit aussi «droits») sur ce fichier ?
- Quelle est l'application par défaut que Nautilus va lancer pour **Ouvrir** ce fichier ? (essayez, vous verrez bien)
- Regardez les **Propriétés** de ce fichier. Est-ce que l'on vous parle de *type* ?

En fait, la notion de *type de fichier* n'existe pas au niveau du système de fichier Unix (à la différence de Mac OS) ni au niveau du noyau et encore moins au niveau de l'extension du fichier (comme sous Windows) mais uniquement au niveau de l'application bureau Gnome : Gnome essaie de deviner quelle

1. L'arrivée de l'interface *Unity* en 2010 sur Ubuntu, puis de *Gnome Shell* en 2011 sur Debian, a provoqué une certaine polémique. Ces types d'interface proposent un tout nouveau mode d'interaction à l'écran en rupture avec les interfaces traditionnelles. Il apparaît que ces interfaces sont plutôt intéressantes avec un écran tactile, et pour un usage ludique. Pour un usage professionnel sur un desktop, beaucoup préfèrent des interfaces plus traditionnelles. C'est pourquoi les administrateurs des machines de l'école vous ont configuré votre environnement par défaut avec l'interface *gnome-flashback* sans effets graphiques. Et à mon avis, ils ont bien raison. Vous pourrez essayer les autres types d'interface, mais plus tard...

2. *Nautilus* est parfois appelé tout simplement *Fichiers*.

3. Notez qu'avec la version actuelle de Nautilus, ce menu n'apparaît pas si le répertoire `~/Modèles/` est vide...

application doit être lancée pour chaque fichier (un mécanisme dérivé de MIME<sup>4</sup>), mais la notion de *type* s'arrête bien là.

De la même manière, Nautilus vous propose un onglet **tag** pour associer des mots clés à votre document, et par la suite faire des catégories et des recherches... Tout cela est purement interne au logiciel Nautilus!

**Permissions Unix :** Sous Unix, chaque fichier est déclaré comme appartenant à un utilisateur *propriétaire*, et à un *groupe* d'utilisateurs. Il y a une dernière catégorie : les *autres* utilisateurs, c'est à dire qui ne sont ni le propriétaire du fichier, ni membre du groupe.

Chacune de ces catégories d'utilisateurs disposent de droits sur ce fichier spécifiant les actions qu'ils sont autorisées ou non à faire dessus. Les actions possibles sont : *la lecture, l'écriture, l'exécution*.

Ceci est souvent représenté de manière textuelle par les fameux trois triplets : **rwX rwx rwx** (Qui signifient : **R**ead **W**rite **eX**ecute pour *le propriétaire, le groupe, les autres*.)

Seul le propriétaire peut changer ces permissions. Le propriétaire peut changer le groupe parmi l'un des groupes auquel il appartient. Seul l'administrateur peut changer le propriétaire d'un fichier, ajouter ou supprimer des utilisateurs à un groupe.

Notons que les administrateurs Unix ont souvent l'habitude de créer autant de groupes que d'utilisateurs : pour chaque nouvel utilisateur il existe un groupe qui porte le même nom, avec un seul membre (l'utilisateur lui-même), et qui est le groupe par défaut de l'utilisateur en question.

## 1.2 Fichiers et répertoires, les vôtres et les autres

Explorons l'arborescence du système de fichiers...

Dans la fenêtre principale de Nautilus, faites **<Ctrl-L>** (ou via le menu principal de Nautilus, puis **Saisir un emplacement**). Dans la barre du haut, le champ **Emplacement** vous offre la possibilité de changer d'emplacement en tapant au clavier et non plus à la souris (ouf!).

- Quel était le nom du répertoire qui s'est affiché naturellement lorsque vous avez lancé l'explorateur ? Ce répertoire est votre répertoire d'accueil (*home directory*).
- Regardez les **Propriétés** d'un fichier ou d'un répertoire ; quel est le symbole séparateur dans les noms des répertoires affichés dans l'item **Emplacement** ?

Placez-vous à la racine du système de fichiers et tentez de créer un nouveau répertoire (clic droit).

- Quel est le problème ?
- Y a-t-il une solution ?

Promenez-vous dans l'arborescence et retrouvez votre répertoire d'accueil. Dans la barre d'**Emplacement**, tapez le nom d'un autre répertoire (par exemple **/usr/bin**).

- Que vient-il de se passer ?

## 1.3 Les fichiers cachés

Retournez rapidement dans votre répertoire d'accueil avec l'icône **Dossier personnel**. Modifiez les paramètres d'affichage de Nautilus (à droite, dans la barre du haut, le bouton qui représente une flèche qui descend), et cochez **Afficher les fichiers cachés**.

Vous venez de faire apparaître de nouveaux fichiers dans votre répertoire d'accueil. Tous ces fichiers ont un nom qui commence par un caractère «point». Ce sont les «fichiers cachés» sous Unix. On voit qu'ils ne sont pas très bien cachés, c'est en fait une simple convention alors que sous Windows c'est un attribut du fichier. Vous pouvez constater qu'il n'y a pas que des fichiers, il y a aussi des répertoires cachés.

Les fichiers et répertoires cachés contiennent généralement des informations d'environnement. Les répertoires dont le nom commence par **.gnome** contiennent par exemple des informations sur votre environnement graphique, ils pourront contenir la mémorisation de votre écran en cours lorsque vous vous déconnecterez.

Les fichiers et répertoires cachés sont souvent un peu équivalents à ce qu'on appelle la «base de registres» sous Windows. Beaucoup d'applications mémorisent leurs paramètres et les préférences de l'utilisateur dans ce type de fichiers. Ils constituent en quelque sorte votre base de registres personnelle.

Un fichier caché est particulièrement important pour vous à Télécom Bretagne, il s'agit du fichier **.profile.PERSO**. Ce fichier contient les paramètres de votre environnement de travail pour les fenêtres émulant des terminaux et vous utiliserez beaucoup ce type d'outil. Mais nous y reviendrons plus tard.

---

4. Cela peut vous sembler magique ? ça l'est un peu puisque l'information se cache derrière **man magic** !

## 1.4 Les fonctions copier/couper/coller

Comme sous Windows il est facile de sélectionner du texte à l'aide de la souris et ensuite de le recopier ailleurs ou de le couper. On sélectionne à l'aide du **bouton gauche** de la souris de la manière suivante :

- Lancez un navigateur web (sur la barre de menu du haut, l'icône représentant la Terre, ou dans le menu **Applications > Internet**). Chargez une page web quelconque (la page par défaut peut faire l'affaire).
- Dans cette page, pointer le début du texte à sélectionner, appuyer sur le bouton gauche et «tirer» la sélection jusqu'à la fin de la zone.  
On peut aussi cliquer deux fois sur un mot pour le sélectionner ou trois fois pour sélectionner la ligne.
- Avec Nautilus, double-cliquez sur votre fichier `essai1`. Nautilus lance l'application Gedit pour l'éditer.
- La fonction «coller» s'obtient en se déplaçant à la destination (ici, la fenêtre Gedit) et en cliquant sur le **bouton milieu** de la souris.

Si vous avez déjà écrit des pages web, vous savez que les données que vous avez sélectionnées à la souris ne se limitent pas à ce simple texte mais qu'il est décoré de nombreuses balises HTML. En fait, lors de cette opération de copier-coller les deux applications négocient un format commun, en l'occurrence l'éditeur texte Gedit demande du texte ASCII simple. Ceci explique que certaines applications peuvent proposer du **Collage sans mise en forme** et autre **Collage spécial**...

Selon les environnements il est aussi possible d'utiliser les jeux de touches classiques suivants (attention, pas toujours effectifs) :

- copier : <Ctrl-C>
- coller : <Ctrl-V>
- couper : <Ctrl-X>

Généralement ces fonctions sont aussi accessibles via le menu **Édition** lorsque celui ci est visible dans la barre de menu des applications.

## 1.5 Utilisation de la corbeille

Dans le navigateur Nautilus revenez dans votre répertoire d'accueil si vous n'y êtes pas déjà. Pointez sur le fichier `essai1` et appuyez sur le bouton droit de la souris. Sélectionnez **Mettre à la corbeille**.

Toujours dans l'explorateur Nautilus, avec l'option **Afficher les fichiers cachés**, déplacez-vous maintenant dans le répertoire `.local/share/Trash`. Fouillez dans les sous-répertoires. Que constatez-vous ?

Comment supprimer réellement ce fichier ?

**Problème de quotas.** Même si vous ne voyez pas les disques physiques, ceux-ci existent bel et bien sur une machine serveur dans le réseau. Leur taille n'est pas extensible à l'infini. Tout le monde doit pouvoir travailler. Vous êtes donc limité en espace disque par des quotas. Nous verrons plus loin comment visualiser quels sont les vôtres. Pour l'instant nous constatons que les fichiers qui semblent effacés par déplacement dans la corbeille ne sont pas physiquement effacés de votre espace disque. Si vous n'effacez pas physiquement vos fichiers, c'est à dire avec un «*Vider la Corbeille*», votre corbeille va remplir petit à petit vos quotas et vous ne pourrez plus travailler.<sup>5</sup> Vous ne pourrez même plus vous connecter car ce processus nécessite justement de créer quelques fichiers !

## 1.6 Les sauvegardes de `.snapshot`

La *corbeille* offre un service rassurant pour les étourdis, celui de pouvoir *recupérer* un fichier qui a été mis à la corbeille. Mais comprenez bien que cette corbeille est une application, un bout de programme qui s'exécute dans votre session graphique (et donc, pas dans une session texte), qui offre ce service en déplaçant les fichiers dans un répertoire caché. Ce n'est pas du tout une fonctionnalité du système d'exploitation Unix : lui, ce qu'il permet, c'est de supprimer un fichier ; et quand c'est supprimé, c'est supprimé ! ... sauf si l'on a un système de sauvegarde...

À l'école, un système de sauvegarde à plusieurs niveaux a été mis en place. Les fichiers utilisateurs sont dans une baie de stockage qui utilise des disques RAID (pour palier aux défaillances matérielles). Les système de gestion de fichiers utilisé est *ReiserFS* qui offre un service de journalisation des données. Ces données sont sauvegardées sur bande régulièrement (le soir, typiquement).

Bref, si vous avez perdu vos données, il devrait être possible de les récupérer, mais ça peut être plus ou moins compliqué suivant les cas. Et la manière la plus simple, c'est le `.snapshot` de *ReiserFS*.

Avec Nautilus revenez dans votre répertoire d'accueil. Repérez le répertoire caché `.snapshot`. Il est possible que ce répertoire ne soit pas présent pour le moment ; il y a un mécanisme d'*auto-montage* qui ne va le charger

---

5. Vous pouvez consulter l'état de vos quotas dans le menu **Applications > Accessoires > Synthèse des quotas**

que si l'on a besoin d'y accéder. Dans ce cas, tapez son nom dans la barre d'Emplacement de Nautilus. Vous y trouvez divers sous répertoires, typiquement : `hourly.0/ hourly.1/ ... nightly.0/ nightly.1/`

Promenez-vous là dedans. Chacun de ces sous-répertoires est une *vue* de vos fichiers tels qu'ils étaient il y a une heure, deux heures, la nuit dernière, etc. Bref, si vous avez fait une petite bêtise et supprimé un fichier par mégarde, le plus simple est d'aller le rechercher ici (plutôt que d'aller trouver les administrateurs systèmes de l'école pour leur demander de ressortir les bandes de sauvegarde et de lancer une restauration...)

## 1.7 Le répertoire Bureau (Desktop)

- Avec Nautilus, allez dans le sous-répertoire **Bureau** de votre répertoire de travail. Maintenant, cliquez avec le bouton droit sur votre fond d'écran et faites **Nouveau document > Document vide**. Que constatez-vous ?
- Mettez ce fichier à la corbeille. Que constatez-vous ?

## 1.8 Quitter sa session

Lorsque l'on souhaite quitter sa session, il est préférable d'adopter l'attitude suivante :

- Allez à l'imprimante ramasser tout ce que vous avez imprimé et que vous n'êtes pas encore allé chercher...
- Faites le tour de vos applications et fermez-les proprement (sauvegarder vos fichiers), sans quoi votre environnement va les fermer brutalement sans forcément faire ce que vous auriez aimé qu'il fasse. Pensez également à vérifier chacun de vos bureaux virtuels (dans le coin en bas à droite ; ce TP n'a pas parlé de ces choses là, vous êtes libre de les découvrir par vous-même).
- Cliquez sur le bouton en haut à droite de l'écran et choisissez *Fermer la session*.
- Éteignez l'écran, cela fera quelques économies d'énergie.

## 1.9 Bilan de l'environnement graphique

Nous venons de réaliser un petit nombre de manipulations avec l'environnement Gnome et l'outil Nautilus. Le fonctionnement que vous avez pu expérimenter est globalement similaire à ce que vous pouvez connaître avec d'autres environnements graphiques sous Unix/Linux (ou même Windows pour certains aspects). Ne perdez cependant pas de vue que ce n'est qu'un environnement possible parmi plusieurs dizaines, qui sont eux-mêmes configurables à l'infini... Bien évidemment, à l'usage vous aurez vos petites habitudes, vos préférences, et vous vous sentirez peut-être plus confortable avec certains environnements qu'avec d'autres. Même si vous trouverez l'un meilleur que l'autre, votre souci en tant que futur ingénieur devra être de savoir s'adapter à n'importe quel environnement...

# 2 L'interface en ligne de commande

Le travail à l'aide des interfaces graphiques est souvent facile et intuitif, même si elles sont finalement assez mouvantes (le nombre de boutons à cliquer, leur nom, le rôle, peut varier d'une version à l'autre...). Cependant sous Unix il existe un très grand nombre d'outils accessibles de préférence en «ligne de commande» dans un terminal ou un émulateur de terminal. Ce sont «les commandes Unix». Il est souvent préférable de les utiliser pour un travail rapide et efficace, mais leur manipulation nécessite un apprentissage.

## 2.1 Notion de Shell et de commande

Ouvrez un émulateur de terminal : dans la barre de menus du haut, choisissez **Applications > Accessoires > Terminal**. Une fenêtre apparaît à l'écran. Elle affiche un petit texte abscons appelé l'*invite* ou «*prompt*» qui comme son nom l'indique vous invite à saisir une commande. En fait, vous avez deux choses à ce niveau : d'une part vous avez une application graphique qui sait afficher une fenêtre, gérer un petit ascenseur de défilement, afficher des caractères ASCII, et sous différents encodages..., et d'autre part vous avez un interpréteur de commande qui porte le nom de Shell.

Le Shell associé à votre fenêtre vous demande d'entrer une commande au clavier. Lorsque vous allez le faire, il va analyser cette commande puis lancer son exécution. Cette commande peut être complexe, elle correspond généralement à un fichier exécutable mais on peut paramétrer son exécution de différentes manières. L'analyse préalable à l'exécution consiste à prendre en compte ce paramétrage.

## 2.2 Les commandes Unix et le Shell

Pour ce qui suit, vous pouvez vous aider des pages web suivantes sur Moodle :  
<http://formations.telecom-bretagne.eu/fad/mod/url/view.php?id=17353>  
<http://formations.telecom-bretagne.eu/fad/mod/page/view.php?id=32652>

### 2.2.1 Les répertoires . et ..

Dans la fenêtre d'émulation de terminal tapez la commande `ls -al`. Vous devez voir apparaître la liste du contenu du répertoire dans lequel vous êtes incluant les fichiers et répertoires cachés. Par rapport à la liste affichée dans le navigateur Nautilus vous avez en plus les répertoires . et ..

Exécutez les commandes suivantes :

- `pwd` (print working directory)
- `cd ..` (change directory)
- `pwd`
- `cd .`
- `pwd`

Qu'en déduisez-vous sur le rôle de ces répertoires :

- le répertoire ..
- le répertoire .

Petit quizz freudien : peut-on avoir `.=..` ?

### 2.2.2 La racine

Remontez au niveau le plus haut de l'arborescence (soit avec de multiples `cd ..`, soit avec `cd /`), puis utilisez la commande `ls -ld`.

- Comment s'appelle la racine ?
- Qui est le propriétaire de la racine de l'arborescence ?
- Quels sont vos droits sur ce répertoire ?

### 2.2.3 Sauter de branche en branche

Placez-vous dans le répertoire `/usr/bin`. (Commande `cd`)

Sautez dans le répertoire `/usr/lib` (puis revenez) en utilisant chacune des deux méthodes : le *nommage absolu* et le *nommage relatif*.

Amusez à refaire la même chose en sautant de `/usr/share/man/man1` à `/usr/share/man/man2` et réciproquement.

- Dans quel cas peut-on préférer le *nommage absolu* et dans quel cas peut-on préférer le *nommage relatif* ?

### 2.2.4 Vos fichiers et ceux des autres

- Revenez à votre répertoire de travail : `cd`
- Remontez au répertoire parent : `cd ..`
- Vérifiez tout ça : `pwd` puis `ls`  
En principe, vous devriez voir un sous-répertoire, celui à votre nom, votre répertoire de travail.
- Essayez d'aller dans le répertoire d'un collègue : `cd login_du_collègue`
- Vérifiez tout ça : `pwd` puis `ls`
- Remontez au répertoire parent : `cd ..`
- Vérifiez tout ça : `pwd` puis `ls`  
C'était comme ça avant ? Que s'est-il passé ?

Vos fichiers ne sont pas présents physiquement sur chacun des PC d'école. Ils sont localisés dans une baie de stockage. Chacun des PC peut y accéder en réseau. Lorsqu'un PC en a besoin, il procède à un montage, puis après un certain temps d'inutilisation les fichiers sont démontés. C'est automatique et transparent. Typiquement, lorsque l'on se connecte à un PC, il a besoin de nos fichiers, et procède au montage. Mais si l'on demande à accéder au répertoire du collègue, il procède au montage également. (Ensuite, le fait que le collègue ait positionné des permissions Unix sur ses fichiers est une autre histoire.) C'est un bon moyen de partage simplement des fichiers (pas besoin de clef usb ou de transfert réseau, c'est déjà en réseau).

### 2.2.5 Les méta-caractères


- Lancez la commande suivante : `ls -l /bin/mk*`
- Que constatez-vous ?
  - En déduire le rôle du caractère \*

- Lancez maintenant la commande `ls -ld /usr/bin/zi?` et comparez avec `ls -ld /usr/bin/zi*`
- En déduire le rôle du caractère «?»
- Comparez par exemple `ls -ald .*` et `ls -ald .????*`

### 2.2.6 La «complétion»

Apprenez à taper plus vite que votre ombre!

**Note :** l'utilisation de la complétion dépend fortement du type de shell que vous avez sur votre système (tapez `echo $SHELL` pour savoir). La manipulation proposée ici fonctionne sous `bash`, mais sous `tcsh` on remplacera la séquence `<Tab> <Tab>` par `<Tab> <Ctrl-D>`<sup>6</sup>...

- Entrez la commande `ls -l /bin/mk` suivie immédiatement par la touche `<Tab>`. La touche tabulation est à gauche du clavier, et souvent représentée . Que constatez-vous?
- Faites à nouveau `<Tab>` et voyez le résultat. Quelle aide est alors apportée?
- Complétez la commande avec quelques caractères pour lever toute ambiguïté et réutilisez la touche `<Tab>` pour afficher finalement automatiquement le nom `/bin/mkdir`

Notez que la complétion est une fonctionnalité de confort offerte par le shell que vous utilisez (`bash`, `tcsh`, etc.). Avec des shells plus anciens comme `sh`, cela ne marche pas... Avec les plus modernes on peut faire de la complétion très subtile, sur des noms d'options de commande ou des arguments (par exemple après `man` on n'aura de la complétion que sur des noms de commande dont on a le manuel en ligne) ou encore de la correction orthographique.

### 2.2.7 Les redirections

- Retournez dans votre répertoire de travail en tapant simplement : `cd`
- Demandez l'heure : `date`  
Que constatez-vous?
- Redirigez maintenant la sortie de la commande précédente dans le fichier `maintenant.txt` dans votre répertoire d'accueil : `date > maintenant.txt`  
Que constatez-vous?
- Listez le contenu de votre répertoire de travail : `ls -l`  
Alors?
- Affichez le contenu de ce fichier : `cat maintenant.txt`  
Comparez avec la date maintenant : `date`

### 2.2.8 Les tubes de communication

Combien de fois avez-vous tapé la commande `cd` depuis tout à l'heure?

Non, ne cherchez pas dans votre tête, l'ordinateur a meilleure mémoire que vous.

- Consultez l'historique de votre interpréteur de commande : `history`  
C'est à la fois le contenu du fichier caché `~/.bash_history` et le contenu de ce qu'il a en mémoire depuis que vous avez ouvert votre terminal. Lorsque vous fermez votre terminal, ce fichier `.bash_history` est mis à jour.
- Tout s'affiche trop vite à l'écran. Pagez via la commande `more` ou la commande `less`<sup>7</sup> de la manière suivante : `history | less`  
Vous pouvez paginer avec les touches classiques `<Page Up>` `<Page Down>`. De plus, pour diverses raisons historiques (les terminaux n'ayant pas toujours eu des claviers si sophistiqués...), vous pouvez utiliser la touche `<Espace>` pour afficher page par page ou avec la touche `<Entrée>` pour afficher ligne par ligne. Pour sortir vous tapez `<Q>` (Quit).
- Bon, reprenons, nous n'allons pas faire le travail à la main, mais le faire faire par l'ordinateur. C'est pour cela qu'il est là. Faisons-le chercher les lignes qui contiennent `cd` :  
`history | grep cd`
- Il ne reste plus qu'à lui faire compter les lignes :  
`history | grep cd | wc -l`

6. Cette différence entre `<Tab>` et `<Ctrl-D>` ci-avant est une subtilité qui a dû paraître géniale aux auteurs de `tcsh` mais est finalement assez troublante pour bon nombre d'utilisateurs. Pour avoir la liste de complétion au bout d'une tabulation ou 2 comme sous `bash`, il suffit de le demander à `tcsh` avec l'option `set autolist` qu'on pourra mettre dans son `~/.cshrc` (ou plutôt son `.cshrc.PERSO`).

7. Le monde d'Unix est un monde d'humour, en particulier au niveau des noms de commande. Un humour un peu élitiste parfois...

### 2.2.9 Les commandes en premier et arrière plan (*foreground, background*)

- Lancez la commande `xeyes`. Dans le même terminal, lancez maintenant une commande quelconque (`pwd` par exemple). Que constatez-vous ?
- Faites `<Ctrl-C>` pour tuer la commande `xeyes` et relancez-la en arrière plan (cherchez comment faire soit dans le cours, soit sur la page web indiquée précédemment). Que constatez-vous comme différence ? Que pouvez-vous faire, alors, qui vous était impossible lorsque la commande était lancée en premier plan ?

## 2.3 La documentation des commandes Unix

Ayez le réflexe RTFM : *Read That Fine Manual*. Chaque commande Unix est documentée dans le manuel Unix en ligne et accessible via la commande `man`.

Dès que vous avez un problème avec une commande, faites : `man nom_de_la_commande`

Vous pouvez chercher un terme dans la page de manuel en tapant `/mot_a_rechercher` qui indique la première occurrence du terme. Pour visualiser les occurrences suivantes, utilisez la touche `n` (*next*). Pour quitter la page de manuel, utilisez `q` (*quit*). (En fait ces touches sont gérées par votre *pageur*, typiquement la commande `more` ou `less`, et dont vous pouvez lire le man...).

Chaque documentation est structurée de la même manière. Vous trouverez essentiellement :

- le nom de la commande et sa description courte ;
- un synopsis, résumé de la manière d'utiliser la commande ;
- une description plus complète du fonctionnement de la commande, avec souvent des exemples concrets d'utilisation ;
- la liste des options acceptées, avec leur signification ;
- à la fin de la page de manuel, la section SEE ALSO peut vous guider vers d'autres commandes connexes.

Dans la suite de ce TP (et de manière générale), prenez le temps et la peine de consulter ces pages de manuel. (J'insiste car c'est comme cela que l'on apprend beaucoup!)<sup>8</sup>

À l'aide du manuel de référence, répondez aux questions suivantes :

- Comment faire une copie récursive de fichiers et de répertoires (commande `cp`) ?
- Comment afficher le contenu d'un répertoire dans l'ordre alphanumérique inverse (commande `ls`) ?
- Quelles sont les options de la commande `man`, et plus particulièrement, que fait l'option `-k` ?
- Quelles sont les pages du manuel traitant du mot *listing* ?
- Quelles sont les pages du manuel traitant du mot *image* ?

Notez également que l'interpréteur de commande (le shell) comprend également quelques commandes internes (`cd` `pwd` ...). C'est donc dans le `man` de votre shell que vous trouverez des explications sur ces commandes. De plus certains shell modernes peuvent vous fournir de l'aide en ligne sur leurs commandes internes. Par exemple sous `bash` on peut faire `help cd`.<sup>9</sup>

## 2.4 Paramétrage de votre environnement de travail en mode terminal

Votre environnement est déjà configuré pour vous permettre d'utiliser la plupart des commandes classiques. Cependant, si l'on veut utiliser une commande non-classique (p.ex. pour faire un exercice de TP particulier, où parce que c'est une commande que l'on a installée soi-même), il faut paramétrer un certain nombre de chose dans notre environnement.

La chose la plus importante est de savoir où se trouve cette commande. Ainsi on va positionner une variable de votre environnement de façon à ce que le shell trouve cette commande sans problème. La variable en question est `PATH` (en majuscules) et contient la liste des répertoires dans lesquels se trouvent les commandes que l'on peut appeler par leur nom simple. Pour consulter son contenu, utilisez la commande `"echo $PATH"`. sont disponibles à l'URL

<http://formations.telecom-bretagne.eu/fad/mod/page/view.php?id=32655>

Notez que cette variable `PATH` peut être vide, par exemple si votre shell a été installé (et compilé) avec un chemin par défaut comme `/bin:/usr/bin`

### 2.4.1 La commande SETUP

La modification environnementale à effectuer est complexe pour le néophyte, aussi, à l'école, nos administrateurs système ont développé une commande spéciale appelée `SETUP` (en majuscules) qui permet de faciliter les choses.

---

8. Vous pouvez aussi vous aider des pages suivantes :

<http://formations.telecom-bretagne.eu/fad/mod/url/view.php?id=17353>

9. Sous `tcsh` cela deviendrait `cd <Alt-h>` si les fichiers de documentation étaient présents.

- Exécutez cette commande pour lister les logiciels accessibles
- Affichez le *chemin* connu par votre environnement pour rechercher les commandes utilisateur :  
`echo $PATH`
- Faites la commande `java -version`, et repérez la version par défaut de votre java.
- À l'aide de la commande `SETUP`, demandez à changer de version (p.ex. `SETUP JAVA17` ou `SETUP JAVA18`)
- Vérifiez que désormais vous travaillez avec une autre version de java : `java -version`
- Vérifiez les changements dans votre configuration : `echo $PATH`

### 2.4.2 Le fichier `.profile.PERSO`

Ouvrez une nouvelle fenêtre terminal et retentez d'exécuter la commande `java -version`. Vous devez à nouveau constater le problème. La modification apportée par la commande `SETUP` a été prise en compte dans le shell associé à la fenêtre précédente mais pas dans celui associé à la nouvelle. Il faut donc recommencer, c'est fastidieux, donc le contraire de l'informatique. Pour éviter ce genre d'ennui il faut que le paramétrage soit pris en compte chaque fois qu'on lance un shell (donc chaque fois qu'on ouvre un terminal). Pour cela il faut écrire la commande de paramétrage dans le fichier `.profile.PERSO` (fichier propre à l'environnement Unix à Télécom Bretagne).

Mais pourquoi donc ce fichier là ? Lors de la question sur les tubes vous avez notamment pu afficher le nom de votre shell de login. Le shell par défaut à l'école est `bash`. (On peut le remplacer par l'un des shells déclarés dans le fichier `/etc/shells` à l'aide de la commande `ypchsh`, lire le `man` avant.) Si vous lisez attentivement le `man bash` vous comprendrez qu'au démarrage de votre session shell (juste après votre login) il exécute entre autres le fichier `.profile`. Ce fichier est déjà configuré par les administrateurs de l'école, et a priori on ne le modifie pas. Mais si vous le lisez vous constaterez que la dernière instruction consiste à exécuter les commandes qui se trouvent dans un autre fichier, le fichier `.profile.PERSO`. Ceci explique cela... Sachez donc ce que vous faites avant de modifier votre shell par défaut ou votre fichier `.profile`... En particulier testez dans de nouvelles fenêtres avant de vous déconnecter car il est facile de scier la branche sur laquelle repose votre environnement et vous par la même occasion.

Donc, dans l'état actuel des choses, le fichier `.profile.PERSO` est le bon endroit pour des paramètres personnels. Modifiez ce fichier pour y incorporer la bonne commande `SETUP` indiquée à la question précédente. Ouvrez une nouvelle fenêtre terminal (qui va lancer un nouveau shell qui va exécuter `.profile.PERSO`), et tentez de lancer de nouveau la commande `jcvs`. Cela devrait marcher cette fois et les suivantes... (Notez que vous n'aurez peut être pas besoin de l'application `jcvs` tout de suite.)

## 3 L'impression de documents

L'impression est un problème important pour tout établissement, c'est un centre de coût non négligeable. Il en va de même pour une école, il faut que nous en soyons tous bien conscients (élèves et personnels), d'ailleurs on se voit imposer des quotas...

L'impression pose aussi des problèmes d'utilisation car il existe plusieurs outils pour la mettre en œuvre. Les documents créés via des outils de traitement de texte sont imprimables à l'aide des outils eux-mêmes (typiquement menu `Fichier > Imprimer`). C'est le cas pour OpenOffice (outil similaire aux outils Office de Microsoft) ou pour Mozilla ou Firefox (navigateurs Web).

Les imprimantes étant généralement accessible en réseau, il convient de connaître le nom «réseau» de celle que l'on souhaite utiliser. Aussi, les administrateurs ont généralement la bonne idée de coller une petite étiquette sur chaque imprimante indiquant son nom. Les imprimantes disponibles depuis les salles Unix/Linux portent les noms suivants : `IMP-DF-200`, `IMP-DF-300`, etc. Imprimez donc préférentiellement sur l'imprimante qui est dans votre salle.

### 3.1 La commande `a2ps`

La plupart des applications permettant de manipuler des documents offrent des fonctions natives pour imprimer ces documents. Pour de simple fichiers textes, tels que des listings de programmes vous pouvez utiliser la commande `a2ps` (*Ascii to Postscript*) qui a le bon goût d'imprimer en deux pages par feuilles et d'embellir le contenu imprimé en fonction de son type (par exemple si c'est un source de programme, mettre en gras les mots clés, numéroter les lignes...) et être hautement paramétrable.

Syntaxe : `a2ps -Pimprimante nom_du_fichier_listing`

Le mot *imprimante* est bien entendu à remplacer par le nom réel de l'imprimante destinatrice. En principe, chaque poste est configuré de telle sorte que l'imprimante par défaut soit l'imprimante la plus proche. Dans ce cas on peut se passer de l'option `-Pimprimante`.

La commande `a2ps` a de nombreuses options (voir le `man`). Par exemple pour une impression avec numérotation des lignes il faut rajouter l'option `--line-numbers=x`. Si `x` vaut 1, toutes les lignes seront



numérotées, s'il vaut 5, les numéros figureront toutes les 5 lignes...

Questions :

- Dans quelle salle vous trouvez-vous et quelle est l'imprimante la plus proche ?  
La meilleure méthode pour le savoir est d'aller voir. D'ailleurs, il faudra bien faire le chemin pour aller récupérer ses impressions, alors autant repérer l'itinéraire maintenant.
- Imprimez votre fichier `.profile.PERSO`  
Non, je rigole, pas la peine de gaspiller du papier si vous n'en n'avez pas besoin. Mais faites semblant.

Note : `a2ps` ne sait malheureusement pas gérer nativement l'encodage *UTF-8* utilisé pourtant de plus en plus pour gérer les caractères accentués ou autres caractères non purement *ASCII*... Aussi, les commandes `iconv` ou `recode` peuvent vous aider à convertir votre document texte encodé en *UTF-8* dans l'encodage *latin1* voir *latin9*, avant d'utiliser `a2ps`...

### 3.2 Autres commandes

- Pour connaître les imprimantes accessibles depuis son poste : `lpstat -v`
- Pour savoir si l'une des ces imprimantes a été désignée comme imprimante par défaut : `lpstat -d`
- Pour envoyer directement à l'imprimante des fichiers dans un langage qu'elle comprend (*ASCII*, *Post-Script*) : `lp -d imprimante fichier`
- Pour lister les travaux en cours : `lpstat -t`
- Pour annuler un travail (possible pour l'utilisateur qui a lancé le travail et depuis le même poste) : `cancel numéro_travail` (Le numéro du travail est obtenu avec `lpstat -t`)  
Note : la commande `lprm` est équivalente à la commande `cancel`

**Plus de papier !** Vous observerez qu'à proximité de chaque imprimante une petite affichette vous indique où vous procurer du papier s'il en manque. On évitera donc d'aller piller le bac à feuilles d'une autre imprimante, et on ira chercher du papier là où il se doit, et sans rouspéter «c'est toujours moi qui y vais»... Bien évidemment, les personnes qui peuvent vous procurer des ramettes de papier ont des horaires. Il faut donc anticiper. Si vous découvrez à 19h30 qu'il n'y a plus de papier pour imprimer les truc super important, on ne pourra pas vous aider. Vous observerez également qu'il y a des périodes de l'année comme ça où tout le monde a besoin d'imprimer son rapport ou son CV le même jour (soir)... Anticipez.

## 4 Travail sur les répertoires et fichiers

Vous êtes connecté(e). Lancez une fenêtre émulant un terminal. Dans cette fenêtre une chaîne de caractères (le prompt) vous indique que vous pouvez saisir une ligne de commande. Dans le tableau qui suit vous trouverez en colonne de gauche sur chaque ligne un travail à réaliser. En colonne de droite vous indiquerez la ou les commandes utilisées ainsi que leurs options et arguments, vous indiquerez aussi leur résultat.

	Travail	Aide	Réponse
4.1	Affichez le nom de votre répertoire de travail (c'est aussi votre répertoire d'accueil si vous venez de vous connecter).		-
4.2	Listez le contenu de votre répertoire d'accueil (listing long : option <code>-l</code> ). Listez les fichiers cachés (idem).	<code>ls</code>	- -
4.3	Déplacez vous dans le répertoire <code>/etc</code> Vérifiez que vous y êtes bien.	<code>cd</code>	- -
4.4	Revenez dans votre répertoire de travail en utilisant la commande la plus simple possible (sans aucun argument). Vérifiez que le changement est correct.		- -
4.5	Créez le répertoire Unix et déplacez vous dedans. Vérifiez le déplacement.	<code>mkdir</code>	- - -

4.6	Copiez dans ce répertoire le fichier <code>/etc/passwd</code> . Vérifiez que la copie est correcte (avec arguments donnant au moins la taille des fichiers).	<code>cp</code>	- -
4.7	Renommez cette copie en lui donnant le nom <code>test</code> . Vérifiez...	<code>mv</code>	- -
4.8	Déplacez vous dans le répertoire <code>/usr</code> . Vérifiez que vous y êtes.		- -
4.9	Depuis cet endroit, listez le contenu de votre répertoire Unix en utilisant le raccourci donné par le caractère <code>~</code> (sorte de constante contenant le nom du répertoire d'accueil <sup>10</sup> )	<code>ls</code>	-
4.10	Vous êtes toujours dans <code>/usr</code> , tentez d'y copier le fichier <code>test</code> que vous avez créé précédemment (réutilisez le raccourci <code>~</code> pour nommer votre fichier). Vous devez normalement rencontrer un problème... Lequel et pourquoi? Pouvons-nous remédier à ce problème?	<code>cp</code> , <code>ls</code> , <i>droits</i>	-  Commentaires :
4.11	Retournez dans votre répertoire Unix. Faites une copie du fichier <code>test</code> que vous nommerez <code>test1</code> . Vérifiez le résultat.		- - -
4.12	Supprimez le fichier <code>test</code> . Vérifiez...	<code>rm</code> , <code>ls</code>	- -
4.13	Retirez le droit de lecture à tout autre que vous sur le fichier <code>test1</code> en utilisant la notation symbolique. Vérifiez...	<code>chmod</code>	- -
4.14	Rajoutez le droit d'écriture pour le groupe et ceci en notation octale (les droits devront donc être : <code>rw-rw----</code> ). Vérifiez	<i>idem</i>	- -
4.15 <sup>(11)</sup>	Quelle est la valeur de votre <code>umask</code> ? Quels droits auront vos fichiers par défaut (les fichiers non exécutables d'abord, les fichiers issus d'une compilation/édition de liens ensuite)? Créant des fichiers, des répertoires et vérifiez.	<code>umask</code> <code>umask -S</code> <code>man 2 umask</code> <code>ls -l</code> <code>stat</code>	
4.16 <sup>(11)</sup>	Quels sont vos quotas?	<code>quota -vm</code> <sup>12</sup>	
4.17 <sup>(11)</sup>	Quel est l'espace disque accessible depuis votre machine?	<code>df</code>	
4.18 <sup>(11)</sup>	Quel est l'espace disque occupé par tous vos fichiers? (Attention, n'oubliez pas les fichiers cachés).	<code>du</code>	(Attention à l'unité d'affichage : option <code>-k</code> )

10. Le caractère `~` désigne mon répertoire à moi, et `~dupont` celui de l'utilisateur *dupont*.

12. Vous pouvez également appeler le script école `/opt/bin/campux-quota.sh`, ou via le menu Applications > Accessoires > Synthèse des quotas

4.19 <sup>(11)</sup>	Si vous avez déjà au moins une fois lancé <b>mozilla</b> ou <b>firefox</b> vous devez avoir un répertoire <b>.mozilla</b> . Sinon lancez-le, ce répertoire sera créé. Déplacez vous dedans et affichez la taille du répertoire cache qui s'y trouve. À l'aide du menu préférences d'un de ces navigateurs, videz votre cache	<b>du</b>	(Attention à l'unité d'affichage)
----------------------	--	-----------	-----------------------------------

**Les commandes `chmod` et `umask`** manipulent traditionnellement les permissions Unix sous leur représentation en octal. L'octal, c'est la base 8, lorsque l'on compte de 0 à 7. Les nombres de 0 à 7 se codent sur 3 bits. Les permissions Unix **rwX** sont des drapeaux sur 3 bits. Voilà pourquoi on représente souvent les permissions Unix en octal. Cependant, pour plus de confort, les commandes **chmod** et **umask** modernes comprennent également la notation symbolique. Par exemple, pour ajouter les droits d'écriture au groupe pour un fichier, on peut faire **chmod g+w fichier**.

Notation symbolique	Notation octale
-x -w- r-	124
rwX r-x r-x	755
rw- -- --	600

Notez que l'on positionne ici les 3 triplets **rwX**. On a donc une notation à trois nombres en octal. Il est également possible de positionner en même temps les drapeaux *suid bit*, *sgid bit*, et *sticky bit*. On a alors un quatrième nombre en octal qui est placé en tête des trois autres.

Autre chose : comprenez bien que le **umask** décrit un masque, c'est à dire les droits que l'on retire. C'est la négation binaire des permissions que les fichiers obtiendrons par défaut.

## 5 Gérer des archives de fichiers

Il est souvent pratique de manipuler des *archives*, pour donner des fichiers à quelqu'un, pour stocker, etc. Les puristes distinguent deux opérations :

1. L'archivage proprement dit, c'est à dire le fait de regrouper plusieurs fichiers dans le même sac, un fichier archive qui contient les autres.
2. Puis la compression du fichier archive.

Les outils de gestion d'archive combinent volontiers les deux, mais on n'est pas obligé, chaque opération pouvant être réalisée par plusieurs algorithmes différents.

Pour faire simple : dans le monde MS-Windows, on utilise traditionnellement la technique ZIP qui combine les deux. Le fichier archive a alors une extension **.zip**. Dans le monde Unix on préférera l'outil **tar** (voir le **man tar**), puis la compression **gzip**<sup>13</sup> (voir le **man gzip**). Le fichier archive a alors une extension **.tar.gz**, ou parfois plus simplement **.tgz** (c'est pareil).

Autre tradition : dans le monde Unix l'archive contient habituellement un sous-répertoire avec tous les fichiers dedans, alors que dans le monde MS-Windows les fichiers sont à plat dans l'archive.

### 5.1 Ouvrir une archive

1. Téléchargez depuis Moodle le fichier **exo\_tar.tgz**, et enregistrez ce fichier dans votre espace de travail. (Si votre environnement vous propose un *Gestionnaire d'Archive* ou un autre machin graphique du genre, refusez !)
2. Dans votre espace de travail, ouvrez cette archive avec la ligne de commande : **tar xvf exo\_tar.tgz** (lisez le **man tar** pour comprendre ce que font les options "x", "v", et "f". N'hésitez pas à en essayer d'autres.)
3. Regardez le répertoire et les fichiers créés avec la commande **ls**

11. Question optionnelle : sachez que cette commande existe, mais on ne vous demande pas de la connaître par coeur. Vous ne serez pas évalué dessus.

13. Quoi que l'algorithme **bzip2** soit de plus en plus utilisé car il offre un meilleur taux de compression.

## 5.2 Créer une archive

1. Créez un sous répertoire (par exemple `./toto/`).
2. Créez quelques fichiers dedans (par exemple `foo.txt bar.txt`).
3. Revenez au répertoire parent, c'est à dire celui qui contient `./toto/`
4. Créez une archive *tar gzipée* avec la commande : `tar cvfz toto.tgz ./toto`
5. Donnez à votre voisin l'archive créée `toto.tgz` pour qu'il vérifie ce qu'elle contient.

## 6 Exercices sur l'environnement

	Travail	Aide	Réponse
6.1	Quel est le nom de votre Shell	Déjà vu plus haut...	
6.2	Affichez les variables d'environnement de votre Shell.	<code>env</code>	-
6.3	À l'aide de la commande <code>echo</code> , affichez les valeurs des variables suivantes : <code>TERM</code> , <code>MAIL</code> , <code>PATH</code> , <code>USER</code> , <code>LOGNAME</code> , <code>SHELL</code> . Rappel : pour accéder à la valeur d'une variable, faire précéder son nom par le caractère <code>\$</code> . Exemple : <code>echo \$var</code>		<code>echo \$NOM_VAR</code> - - - -
6.4	Créez une variable <i>utilisateur</i> de nom <code>VAR1</code> et de valeur <code>abc</code> et une variable d' <i>environnement</i> de nom <code>VAR2</code> et de valeur <code>xyz</code> . Vérifiez la portée des variables <code>VAR1</code> et <code>VAR2</code> avec la commande <code>env</code> (vous pouvez affiner l'affichage en utilisant un tube de communication : <code>env   grep VAR</code> ) <sup>14</sup>	<code>=</code> <code>export</code>	- -
6.5	Ouvrez deux autres fenêtres terminal, l'une depuis votre shell actuel avec la commande <code>xterm</code> ou <code>gnome-terminal</code> , l'autre depuis le menu graphique. Vérifiez si les shells associés à ces fenêtres connaissent vos variables déclarées dans la première fenêtre. Pour plus de rigueur, et si vous êtes courageux, vous pouvez repérez la filiation des processus shell concernés avec la commande <code>ps tree -h</code> <sup>11</sup> . Expliquez...		-
6.6 <sup>(11)</sup>	La commande <code>mount</code> existe sur le système. Elle est utilisable par tout le monde pour lister les montages de systèmes de fichiers (il faut normalement être <code>root</code> pour monter un système). Avec la commande <code>which</code> , vérifiez que cette commande est accessible Sinon, cherchez avec <code>locate</code> . Indiquez quel est son chemin.	<code>which</code>	-
6.7 <sup>(11)</sup>	Avec la commande <code>whereis</code> cherchez si <code>mount</code> existe dans un répertoire standard. Indiquez le résultat. Avec la commande <code>ls -l</code> , listez chaque élément du résultat précédent. Que constatez-vous ?	<code>whereis</code> <code>whatis</code>	- -

14. Les variables du shell <http://formations.telecom-bretagne.eu/fad/mod/page/view.php?id=32654>

6.8 <sup>(11)</sup>	Quels sont les systèmes de fichiers montés sur votre machine ? Indiquez les systèmes locaux ainsi que les systèmes montés en réseau. Pour ces derniers vous indiquerez le nom des serveurs. Sur quel disque se trouve le système de fichiers racine (le /) ?	mount	-
6.9 <sup>(11)</sup>	Quelle est l'imprimante par défaut de la machine sur laquelle vous êtes connectés actuellement ? À quelles imprimantes avez-vous accès ?	lpstat	-
6.10 <sup>(11)</sup>	Quel est votre numéro d'utilisateur ?	id \$UID	-

## 7 Exercices sur les processus

	Travail	Aide	Réponse
7.1	Testez la commande <code>ps</code> , d'abord sans argument puis avec les options <code>-u \$LOGNAME</code> , et enfin en rajoutant l'option <code>-f</code>		Quelles différences dans les affichages ?
7.2 <sup>(11)</sup>	Testez la commande <code>ps -ef</code>		Que voit-on ?
7.3	Lancez une commande qui ne se termine pas tout de suite, par exemple <code>xeyes</code> . Ouvrez une autre fenêtre terminal et dans celle-ci recherchez le numéro du processus correspondant à la commande <code>xeyes</code> . En restant dans cette seconde fenêtre, utilisez le résultat de <code>ps</code> pour supprimer le processus <code>xeyes</code> .	<code>ps -ax</code> , <code>kill</code>	fenêtre 1 - fenêtre 2 - -
7.4	Relancez la commande <code>xeyes</code> . Votre terminal ne répond plus, les commandes que vous y entrez ne sont pas prises en compte. Stoppez alors le processus <code>xeyes</code> (faites <code>Ctrl-Z</code> ) et passez le en background ( <code>bg</code> ). Tapez la commande <code>jobs</code> . Que vous répond t-elle ? <sup>15</sup>		
7.5 <sup>(11)</sup>	Lancez un nouveau <code>xeyes</code> suivi cette fois du caractère <code>&amp;</code> . Que se passe-t'il ? Faites <code>jobs</code> . Que voyez vous ? À l'aide de la commande <code>kill</code> et du résultat de <code>jobs</code> , supprimez le second <code>xeyes</code> . <sup>15</sup>	<code>jobs</code> <code>kill</code>	- - -
7.6	Repassez le premier <code>xeyes</code> en premier plan ( <code>fg</code> ) et supprimez le ( <code>Ctrl-C</code> ).		- -
7.7	À l'aide de la commande <code>top</code> ou <code>htop</code> relevez les caractéristiques de votre machine : taille mémoire, taille du swap, charge moyenne, par CPU...	<code>top htop</code>	-

<sup>15</sup>. Le contrôle des tâches : `help jobs bg fg disown` <http://formations.telecom-bretagne.eu/fad/mod/page/view.php?id=32653>

7.8	Avec la commande <code>tty</code> relevez le nom du terminal virtuel associé à votre fenêtre terminal. Avec <code>ps -aux</code> retrouvez le numéro de processus associé à votre fenêtre. Trouvez le numéro du shell associé à la fenêtre.	<code>tty ps</code>	- -
7.9	En utilisant le mécanisme des tubes de communication affichez la 23 <sup>ème</sup> personne à s'être connectée dernièrement sur votre machine. L'historique des dernières connexions est accessible avec la commande : <code>last</code> Vous utiliserez les commandes <code>head</code> et <code>tail</code> pour filtrer la sortie standard de <code>last</code> et obtenir le résultat souhaité. <sup>16</sup>		<code>last   .....</code>
7.10	Rediriger la sortie standard de la commande <code>ls</code> dans un fichier. Vous listerez le contenu de <code>/usr</code> .		

---

16. Si personne ne s'est connecté dernièrement, ça ne sera pas rigolo. Dans ce cas faites l'exercice en regardant l'historique des dernières commandes que vous avez tapées (commande `history`).