

# Série 1 :

## Découverte du cours et de l'environnement *Unix* : fichiers, éditeurs, commandes, ...

### Buts

Le but de cette série d'exercices est de vous

1. offrir une première prise de contact avec l'environnement informatique de l'EPFL ;
2. permettre de vous familiariser avec votre compte informatique ;
3. faire connaître des principales manipulations de base pour évoluer dans un environnement Unix que vous serez amenés à utiliser : manipulation et édition de fichiers, commandes Unix, etc....

**NOTE :** Vous devez impérativement réaliser **tous** les (dix) exercices de cette série afin d'être parfaitement à l'aise avec l'environnement informatique de ce cours. Il ne faut en particulier pas rater la **fin de l'exercice 5**.

## Présentation général du cours Information, Calcul et Communication

Si ce n'est pas encore fait, je vous demande avant de commencer cette série d'exercices, de prendre connaissance du document suivant qui présente le cours dans son ensemble :

- [\[PDF\] Présentation du cours](#)

## Prérequis pour la suite de la série

[ Pour plus de détails, voir le site <http://studying.epfl.ch/bienvenueBachelor> ou sinon le [guichet des étudiants](#). ]

### Pré-requis :

1. Avoir son mot de passe *Gaspar* ;
2. Disposer d'une *adresse e-mail* à l'EPFL.

**Ces prérequis sont remplis dès votre inscription à l'École**, et vous devriez normalement pouvoir **passer directement au premier exercice**. En effet, en tant qu'étudiant inscrit à l'EPFL, vous avez du recevoir document/email contenant : votre adresse e-mail EPFL, un username et un mot de passe. C'est de cela dont il s'agit ici.

Si ce n'est pas le cas,

- si c'est un problème **technique** : contacter le Help-desk comme indiqué ci-dessous;
- ou sinon par le [guichet des étudiants](#) pour tous les problèmes *administratifs* ou d'*inscription*.

### Rappel :

Le **portail GASPAS** sert à gérer les accès aux ressources sécurisées du Service Informatique Central. C'est le coffre fort pour vos données d'authentification. Vous pouvez par exemple y modifier votre mot de passe.

### Note importante :

En cas de problème **technique** avec les VM ou l'infrastructure EPFL (p.ex. liés au login et à GASPAS, ou pas de répertoire *myfiles*, etc.), un support est assuré par le **«Help-desk» (MA A0 388, téléphone interne : 1234, email : [1234@epfl.ch](mailto:1234@epfl.ch))**.

## Exercice 1 : Se connecter (ou *login*)

Ce premier exercice va vous permettre de vous « connecter » à un (ou plusieurs) des environnements de travail

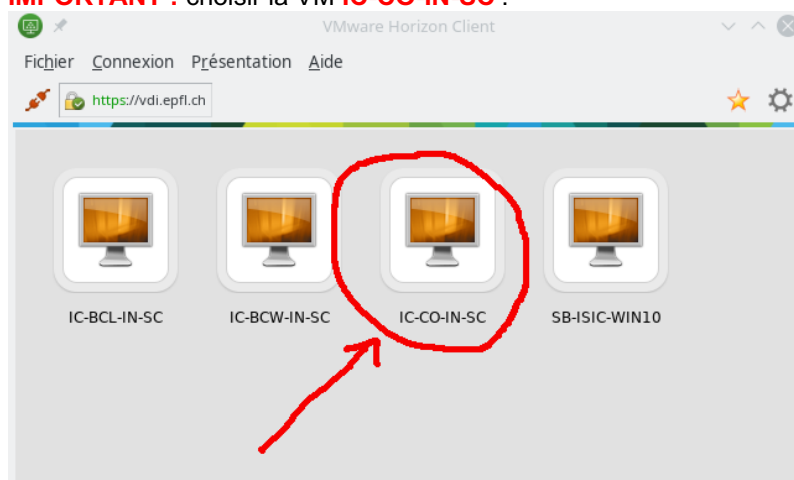
proposés pour ce cours.

Vous avez en effet **HUIT** (ou neuf) façons différentes de pratiquer pour ce cours (non exclusif ! Il peut être intéressant, suivant vos contraintes, d'utiliser *plusieurs* de ces solutions) :

- **1.** soit venir vous connecter directement dans les salles CO-020 à CO-023 (vous y aurez un accès à [myfiles](#) [a.k.a. « MyNAS » ; tout ça est [expliqué plus loin](#)]) ;

**IMPORTANT** : choisir la VM **IC-CO-IN-SC** ;

- soit travailler sur votre propre machine, par ordre de recommandation :
  - **2.** soit en utilisant votre propre environnement (CLion, XCode, Code::Blocks, Emacs, vi, ...) sur votre machine usuelle « comme d'habitude » ; mais cela suppose que vous ayez déjà un tel environnement et sachiez l'utiliser ! Je ne pense donc pas que ce soit le meilleur moyen pour la majorité d'entre vous ;
  - **3.** soit en vous connectant à distance depuis votre machine sur les mêmes machines virtuelles (VM) que celles utilisées en salles CO (vous y aurez un accès à [myfiles](#)) ;  
**MAIS** attention : nombre limité de machines dans le « pool » :
    - (à ne faire qu'une seule fois) installer l'outil de connexion « VMware Horizon Client » comme indiqué dans [la documentation fournie là-bas](#) ;
    - utilisez ce client pour vous connecter à `vdi.epfl.ch` à l'aide de votre identifiant Gaspar ;
    - **IMPORTANT** : choisir la VM **IC-CO-IN-SC** :



**Note** : toutes les autres solutions ci-dessous n'ont **pas** d'accès direct à [myfiles](#), mais vous devez le créer vous même « à la main » comme expliqué dans [cette page de documentation \(lien\)](#).

- **(3.5.)** sur Windows depuis Windows 10 UNIQUEMENT : soit en utilisant Linux directement sur Windows comme indiqué [ici en format HTML](#), [ici en format PDF](#) et [ici en format Markdown](#) ;
- **4.** soit en installant un environnement de développement comme indiqué dans la première semaine du MOOC <https://www.coursera.org/learn/initiation-programmation-cpp> (une telle installation **n'est pas** du tout obligatoire, ce n'est qu'une façon de travailler, parmi les autres citées ici) ; l'idée de ce point est de vous ramener au point 2 ; -) ;
- **5.** soit en travaillant carrément à distance (dans « le cloud », rien du tout à installer) sur des sites comme [repl.it](#) ou comme [tutorialspoint.com](#) (une liste détaillée de ces sites se trouve [sur cette page](#)) ;
- **6.** soit en téléchargeant une VM EPFL et la faisant tourner en local sur votre propre machine ; cf [https://support.epfl.ch/epfl?id=epfl\\_kb\\_article\\_view&sysparm\\_article=KB0012478](https://support.epfl.ch/epfl?id=epfl_kb_article_view&sysparm_article=KB0012478) ;
- **7.** soit en installant une VM Linux : c'est presque comme le point 5 précédent, mais allez télécharger une image à installer directement chez Ubuntu (<http://ubuntu-fr.org/telechargement>) au lieu de la prendre chez nous ; ceci est expliqué en détail sur cette page : <http://wiki.epfl.ch/co02x/vmself> ;
- **8.** soit en installant directement Linux sur votre machine (déconseillé aux débutants, mais l'association **GNU Generation** organise chaque année une « Install Fest ») : voir <http://doc.ubuntu-fr.org/debutant>.

Vous pouvez suivant vos contraintes choisir l'une ou *plusieurs* des solutions ci-dessus.  
Tout ceci est également décrit ici : <http://wiki.epfl.ch/co02x/maison>.

Choisissez pour le moment l'une des méthodes précédentes.

Si, pour une raison ou une autre vous n'arrivez pas pour le moment à avoir accès à une machine similaire à celle du cours, vous pouvez continuer sur votre propre machine par **l'exercice 8** et suivants, et reviendrez faire cette série une fois une machine accessible pour vous.

Pour vous connecter sur les machines fournies (soit directement, soit à distance), tapez votre nom d'utilisateur (*username* Gaspar) puis tapez sur la touche **Return** et attendez que le système vous demande votre mot de passe (*password*).

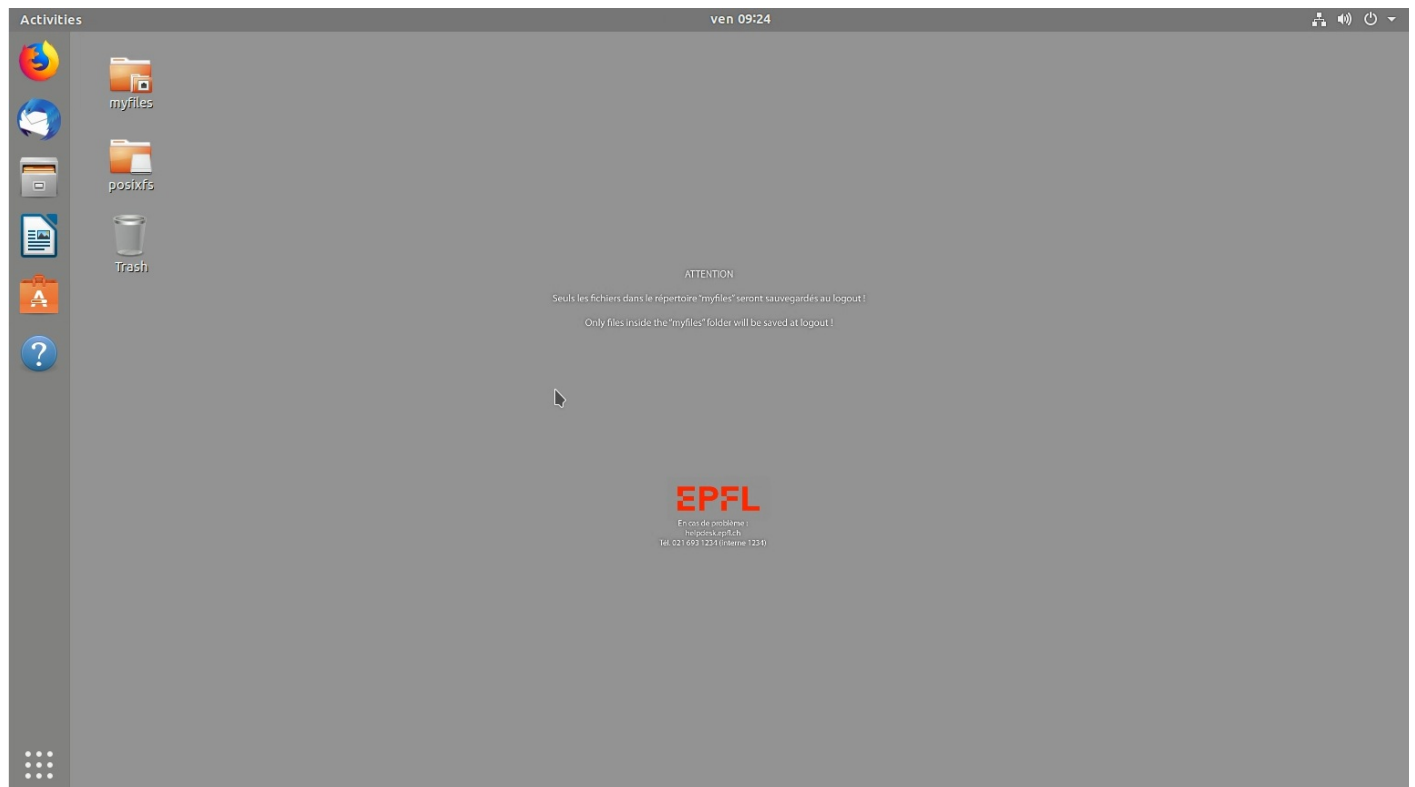
#### Remarques:

- Les majuscules/minuscules sont significatives dans le username et dans le password : 'a' n'est pas la même chose que 'A'.
- Il faut faire attention à la différence entre '0' (zéro) et 'O' (la lettre o majuscule) ainsi qu'entre '1' (chiffre un) et 'l' (la lettre L minuscule).
- Certains caractères (tels que '@' ou '#') ne semblent pas passer sur le login de la machine virtuelle. N'utilisez donc pas de tels caractères dans votre mot de passe.
- **IMPORTANT : ne tapez pas** une seconde fois le mot de passe lors de la seconde fenêtre de login, mais attendez (même si c'est un peu long au début), laissez faire la machine. En clair : **ne saisissez qu'UNE SEULE FOIS votre mot de passe !**

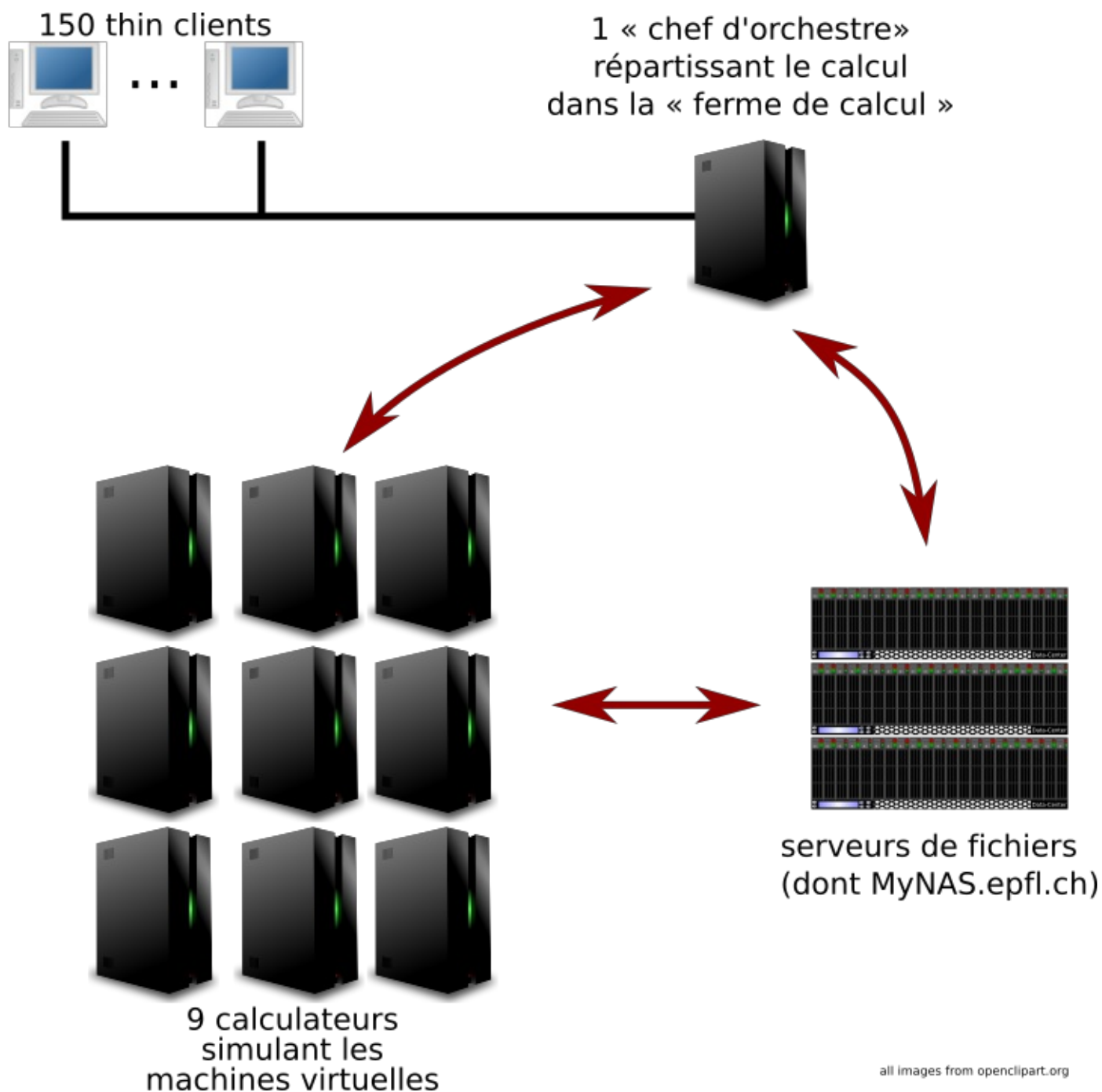
Vous arrivez ensuite sur un écran intermédiaire (écran bleu) ou vous devez choisir une « machine virtuelle ». Pour ce cours, il faut **choisir la machine IC-CO-IN-SC**.

Vous devriez normalement ensuite arriver **automatiquement** sur une machine virtuelle. Si jamais vous avez encore un 2<sup>e</sup> écran bleu intermédiaire (ça ne devrait pas) : il ne faut **RIEN FAIRE** sur ce 2<sup>e</sup> écran bleu de login ; il faut juste attendre.

Si tout s'est bien passé, après quelques instants un écran qui *ressemble* à celui de la figure ci-dessous va apparaître. Sinon recommencez... Au bout de *trois* échecs, appelez un assistant.



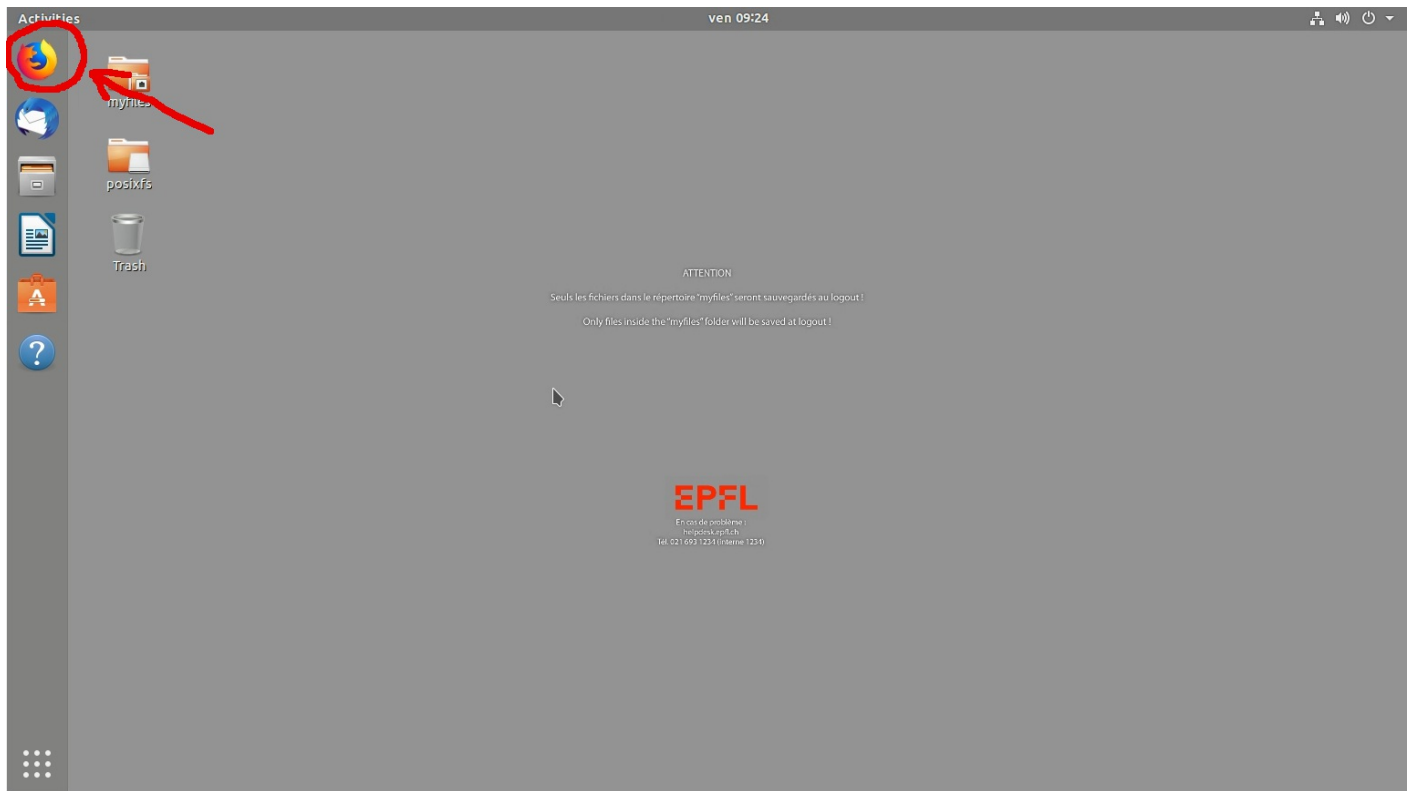
Cette opération en deux étapes est nécessaire car l'architecture est «un peu plus compliquée» que votre ordinateur personnel habituel : vous êtes en effet connecté sur des clients qui s'adressent à un serveur «chef d'orchestre» («*load balancer*») qui demande à un autre serveur de calcul de vous créer (et ouvrir) une «machine virtuelle». C'est cette machine virtuelle que vous voyez finalement sur votre écran. Pour résumer :



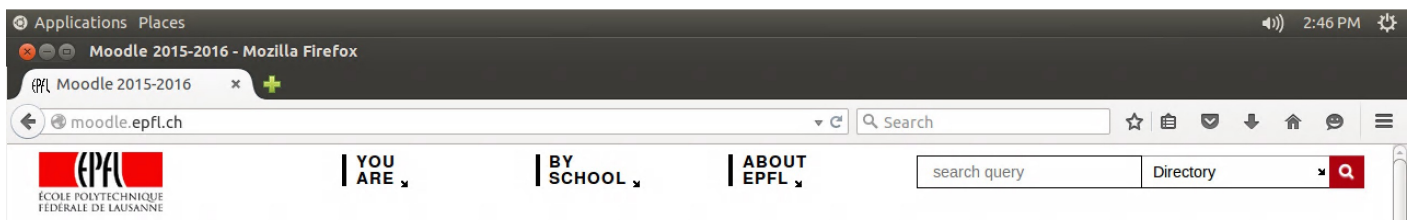
C'est pour cela que vous avez un login en deux étapes : une première pour accéder au «chef d'orchestre», et une seconde sur la machine virtuelle. Normalement, ce second login consiste à **ne rien faire**, le «chef d'orchestre» ayant passé vos paramètres à la machine virtuelle.

## Exercice 2 : Utiliser Firefox

Firefox est un navigateur Internet. Pour le lancer, il suffit de cliquer sur son icône en haut à gauche :



Une fois Firefox lancé, il vous suffit de taper l'URL (adresse à laquelle vous voulez vous connecter) dans le champ prévu à cet effet :



Connectez-vous à la page Moodle de ce cours :

<https://moodle.epfl.ch/course/view.php?id=14023>. Vous pouvez aussi y accéder en vous connectant simplement sous `moodle.epfl.ch` et en allant dans la liste de vos cours.

Cette page (ou celle principale de Moodle, <https://moodle.epfl.ch/>) étant relativement importante, on peut vouloir la garder (pour ne pas avoir à retaper à chaque fois son adresse) dans ses *bookmarks* («signets» en français) :

- Cliquez sur «*Bookmarks*» dans la barre de menu en haut, puis sur «*Bookmark This Page...*», ce qui ouvre une fenêtre.  
Sous «*Folder:*», sélectionnez l'endroit où sera stocké ce signet (bookmark), par «*Bookmarks Toolbar*». Cliquez enfin sur le bouton «*Done*» en bas de la fenêtre qui vient d'être ouverte.

La fenêtre de dialogue doit alors se fermer et, dans votre barre de navigation, vous devez voir un nouvel onglet «*Vue d'ensemble...*» ou bien «*Cours : Informatique...*» (juste au dessus de la fenêtre principale affichant le contenu).

Vous pouvez aussi plus simplement saisir (clic gauche de la souris) la petite icône à coté de l'URL (au début de la barre blanche précédemment citée, juste à coté de la petite maison) et la glisser (en gardant le clic gauche de la souris enfoncé) sur la barre des signets, c'est-à-dire la 3e barre en haut, celle juste au dessus du contenu affiché.

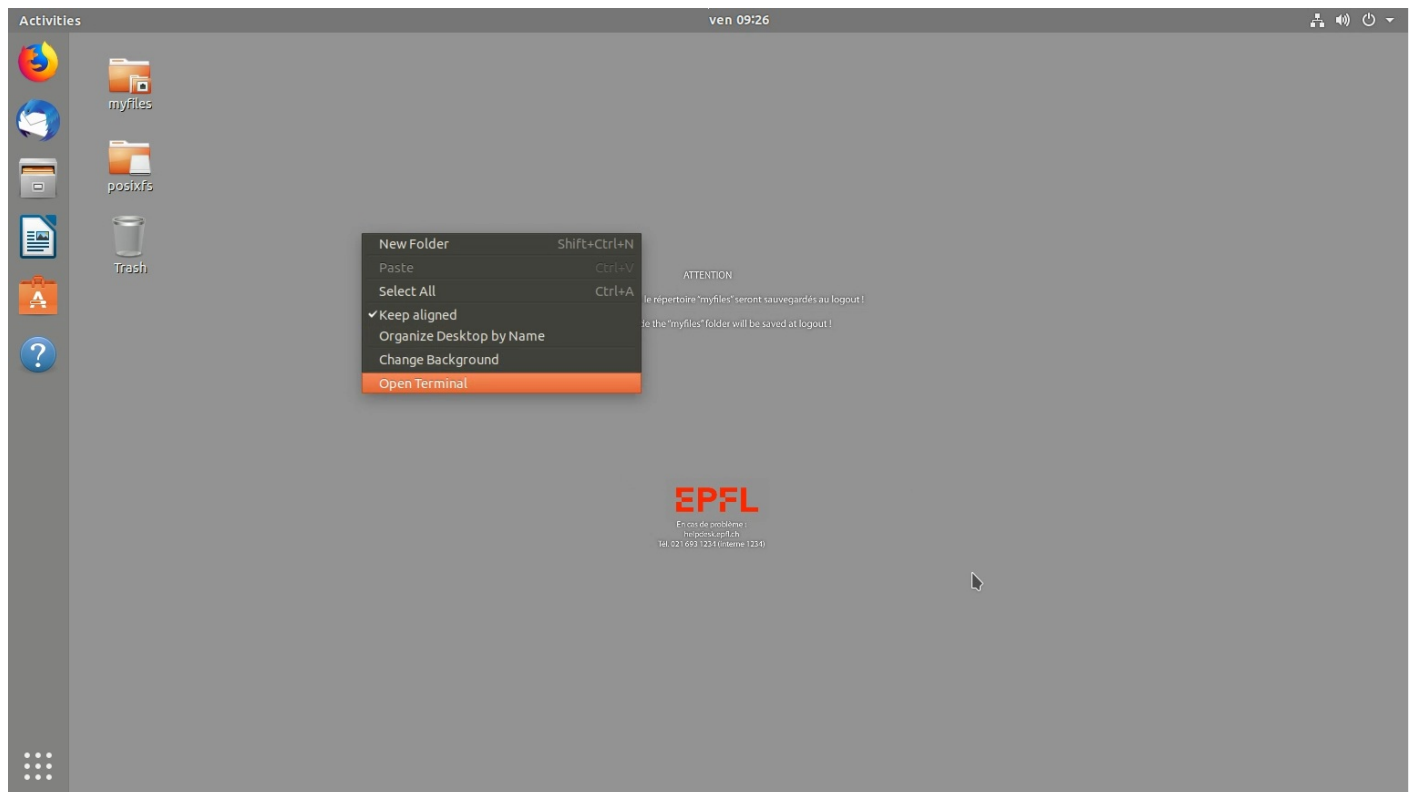
Notez également que la page *spécifique à la partie programmation*, dont l'URL direct est <http://progmap.epfl.ch/>. Vous y trouverez toutes les informations nécessaire à la partie programmation du cours.

Enfin, dans firefox, si vous ne voyez pas la barre «*Bookmarks*», il faut aller au «*View*», puis «*Toolbars*», puis cliquez sur «*Bookmarks Toolbar*». La barre devrait apparaître.

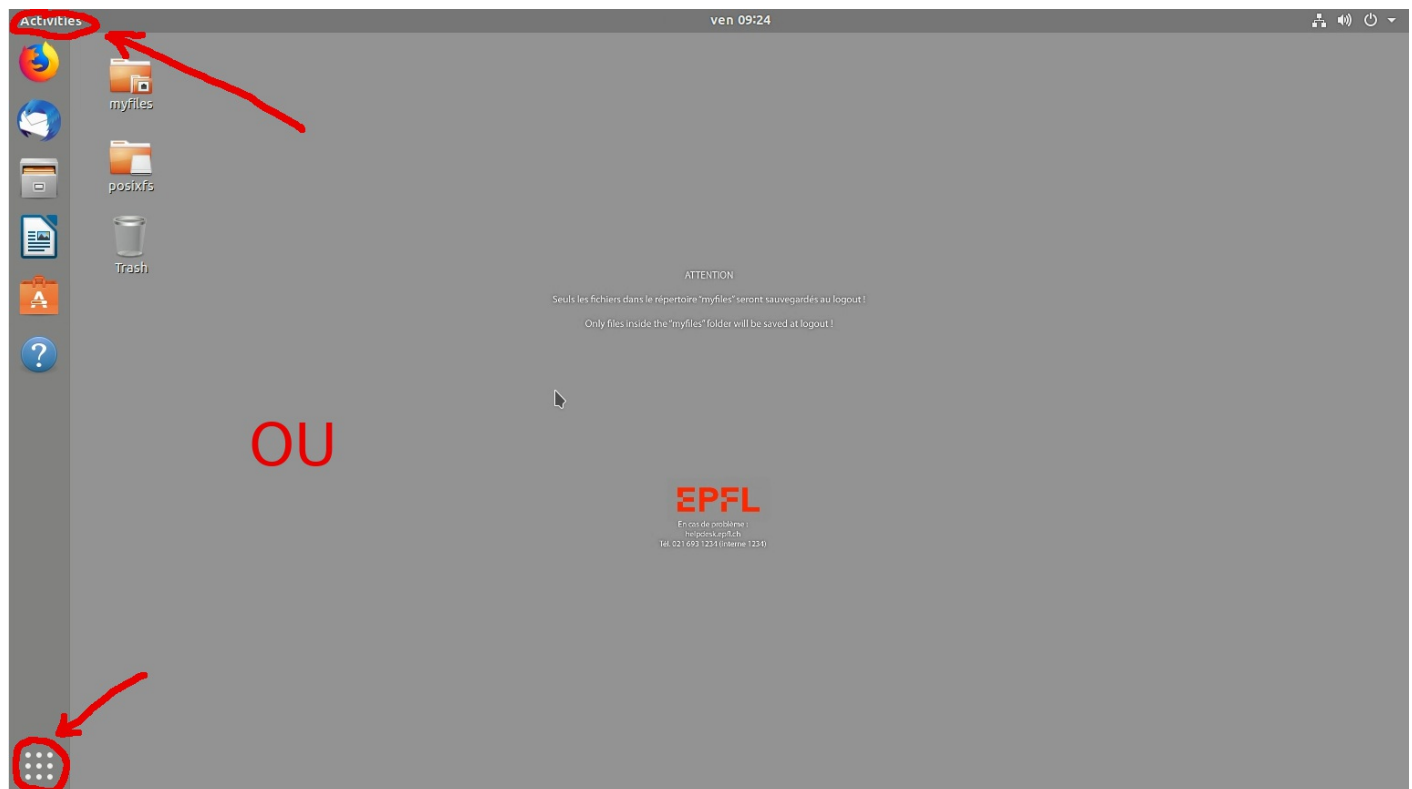
### Exercice 3 : Ouvrir un *terminal* (Linux ou Mac uniquement)

Un «terminal» est une fenêtre qui vous permet de donner directement des commandes au système. Pour en ouvrir un

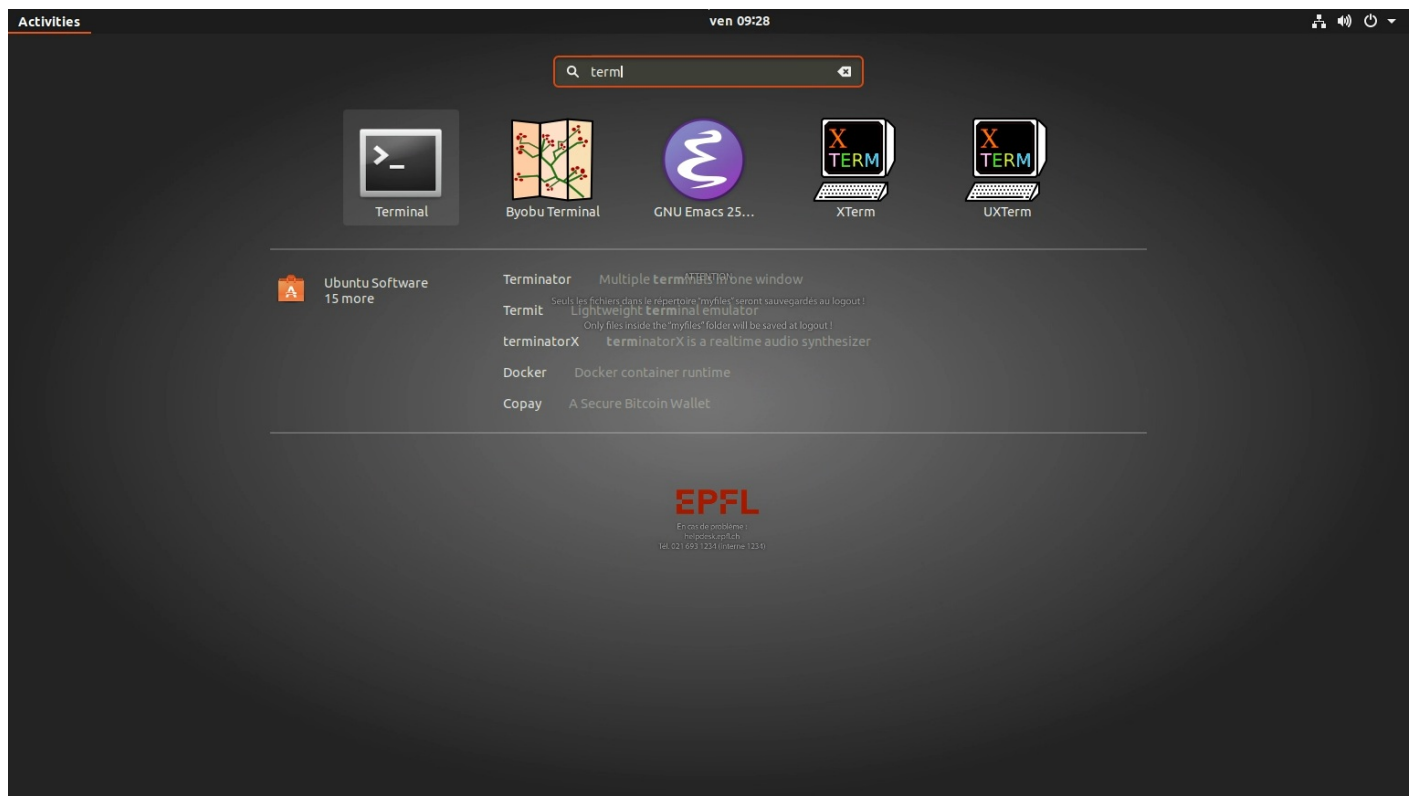
(on peut tout à fait avoir plusieurs terminaux ouverts à la fois), le plus simple est d'utiliser le menu qui apparait en faisant un clic-droit sur le fond d'écran :



Vous pouvez aussi rechercher l'application «terminal» dans les «Activities», soit en cliquant sur le mot «Activities» en haut à gauche, soit en cliquant sur la matrice de 9 points en bas à gauche :



Saisissez ensuite le texte recherché, par exemple «term» :



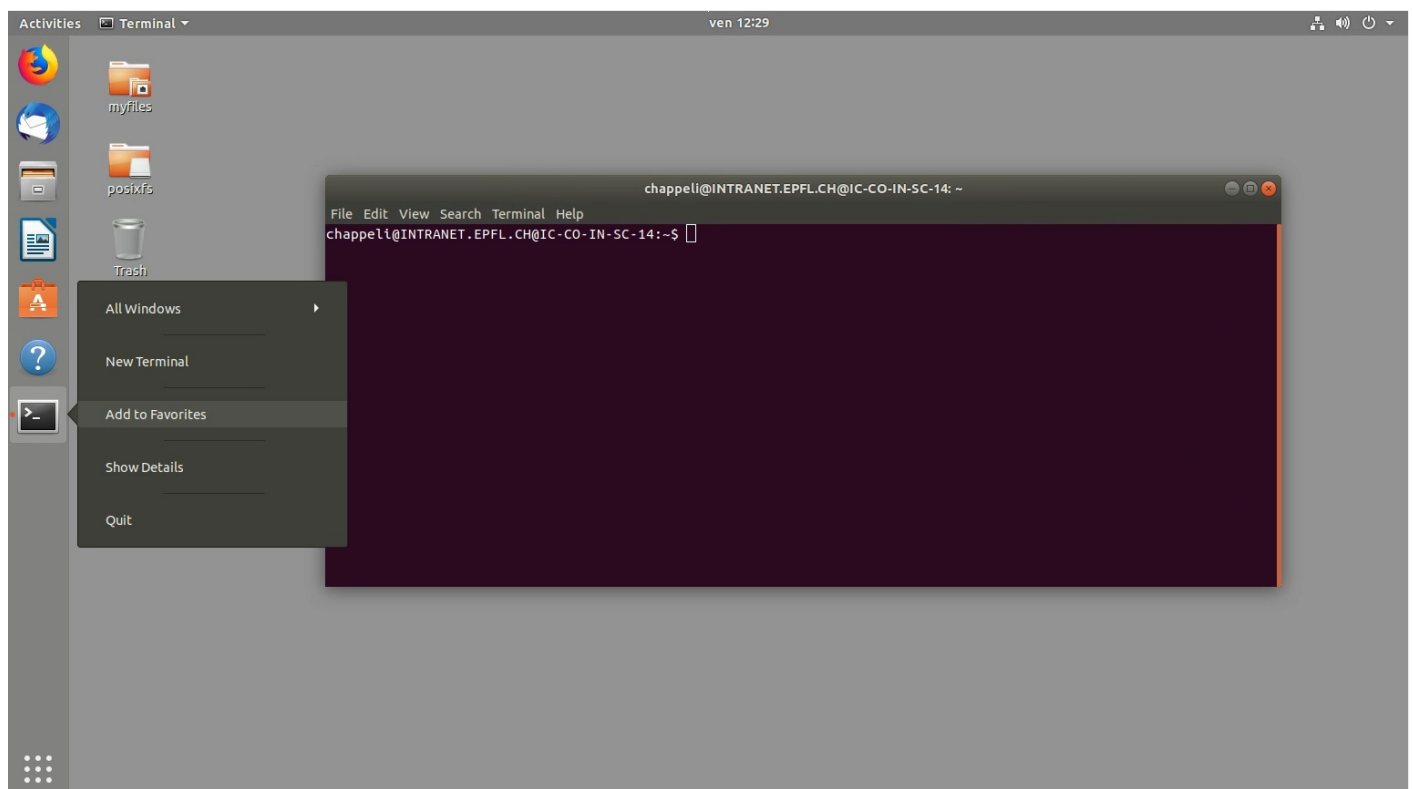
Une fois un «terminal» ouvert, cliquez dessus/dedans pour l'activer et taper la commande

```
ls
```

qui liste le contenu du répertoire courant (voir les exercices en bas pour plus de détails. **ATTENTION !** le premier caractère est un `l` minuscule, pas le chiffre `1` !).

Pour l'instant, seul des fichiers et répertoires pré-existants sont listés. Dans **l'exercice suivant**, nous allons justement voir comment créer vos propres fichiers.

Si vous souhaitez garder un raccourci vers une application que vous utilisez souvent (et c'est conseillé de le faire pour le terminal), il suffit de faire un clic-droit sur l'icône correspondante à gauche de l'écran et choisir «Add to favorite» :





## Exercice 4 : Premier contact avec un éditeur

Supposons que vous ayez envie d'écrire un message, juste en texte (pas de formatage, ni gras, ni couleur, etc.) et de le stocker sur votre compte. Il vous faut pour cela utiliser un utilitaire particulier appelé « éditeur » (plus simple qu'un « traitement de texte »).

La première fonctionnalité d'un éditeur est justement de vous permettre de saisir du texte et de le stocker sous la forme d'un *fichier*. Un programme n'étant au départ rien de plus qu'un fichier texte, nous utiliserons aussi un éditeur pour écrire nos programmes. Il existe cependant des éditeurs plus avancés, spécifiquement dédiés à l'écriture de programmes, c'est ce que l'on appelle des «IDE» («EDI» en français) pour «Integrated Development Environment».

Plusieurs éditeurs ou IDEs sont à votre disposition dans le cadre de ce cours : **Geany, Visual Studio, Sublime Text, Qt Creator, Emacs, Vim, Code::Blocks, GEdit, ...**

Pour lancer l'un ou l'autre de ces éditeurs/IDE, il suffit de les rechercher dans les «Activités» en haut à gauche ou en bas à gauche, exactement comme indiqué précédemment pour le terminal.

Pour commencer, choisissez un éditeur particulier (par exemple/conseillé [geany](#)) et lancez le.

Pour ouvrir un nouveau fichier dans l'éditeur que vous avez choisi, il suffit de sélectionner dans la barre menu l'option *File -> New* ou *File -> Open File* (selon l'éditeur. Dans Emacs, il faut donner un nom de fichier en bas de la fenêtre, par exemple `mon-1er-message.txt`).

Saisissez ensuite un texte quelconque dans la zone de l'éditeur dédiée à la saisie, puis sauvegardez-la au moyen de l'option *File -> Save*. Un nom vous sera alors demandé pour désigner le fichier dans lequel sera stocké votre document. Saisissez par exemple `mon-1er-message.txt` en guise de nom. Une fois que la sauvegarde est effectuée, vous pouvez quitter l'éditeur.

**Note :** pour certains éditeurs (comme [emacs](#) par exemple), l'ouverture d'un nouveau fichier vous demandera un nom de fichier avant même que vous ayez pu saisir du texte.

**A savoir :** un nom de fichier est aussi bien un nom «simple», comme `maphoto.png`, qu'un nom «complet» indiquant où le fichier est stocké sur le disque, comme `/home/chaps/Images/maphoto.png`.

On parle

- de nom *absolu* pour le second, car où qu'il soit donné il désigne toujours l'emplacement précis du fichier ;
- et de nom *relatif* pour le premier (simple) car il n'a de sens que par rapport à (=relativement à) l'endroit où il est effectivement stocké.

Quand on parle de «nom de fichier», on parle sans distinction de l'un ou de l'autre. Si un nom relatif est donné, il sera toujours interprété **PAR RAPPORT** À l'endroit où la commande s'exécute.

(Linux ou Mac ; pour Windows utilisez simplement le navigateur de fichiers) Placez-vous ensuite dans un terminal et tapez à nouveau la commande

```
ls
```

qui liste le contenu du répertoire courant. Vous devriez alors voir apparaître le fichier `mon-1er-message.txt` que vous venez de créer via l'éditeur.

Ce fichier peut maintenant être ouvert à nouveau et modifié par n'importe quel éditeur (identique ou différent de celui qui l'a créé). Il restera stocké sur votre compte d'une session de travail à l'autre.

Essayez de l'ouvrir (option *File -> Open*), le modifier et le sauvegarder à nouveau avec chacun des autres éditeurs à tour de rôle (n'ouvrez pas plusieurs éditeurs en même temps !).



Ceci vous permettra de vous faire une première idée sur les différents éditeurs mis à votre disposition et vous permettra de porter votre choix par la suite sur celui qui vous semble le plus confortable. Nous reviendrons plus en détail sur les fonctionnalités des éditeurs dans les exercices suivants.

**Note :** Il est important de ne pas ouvrir plusieurs fois le même fichier (via différents éditeurs par exemple). La cohérence du fichier modifié par l'un ou l'autre des éditeurs n'est alors plus garantie !

Concernant les fichiers et leur manipulation, nous en verrons plus dans les exercices suivants, mais déjà une note importante (qui sera expliquée un peu plus en bas) :

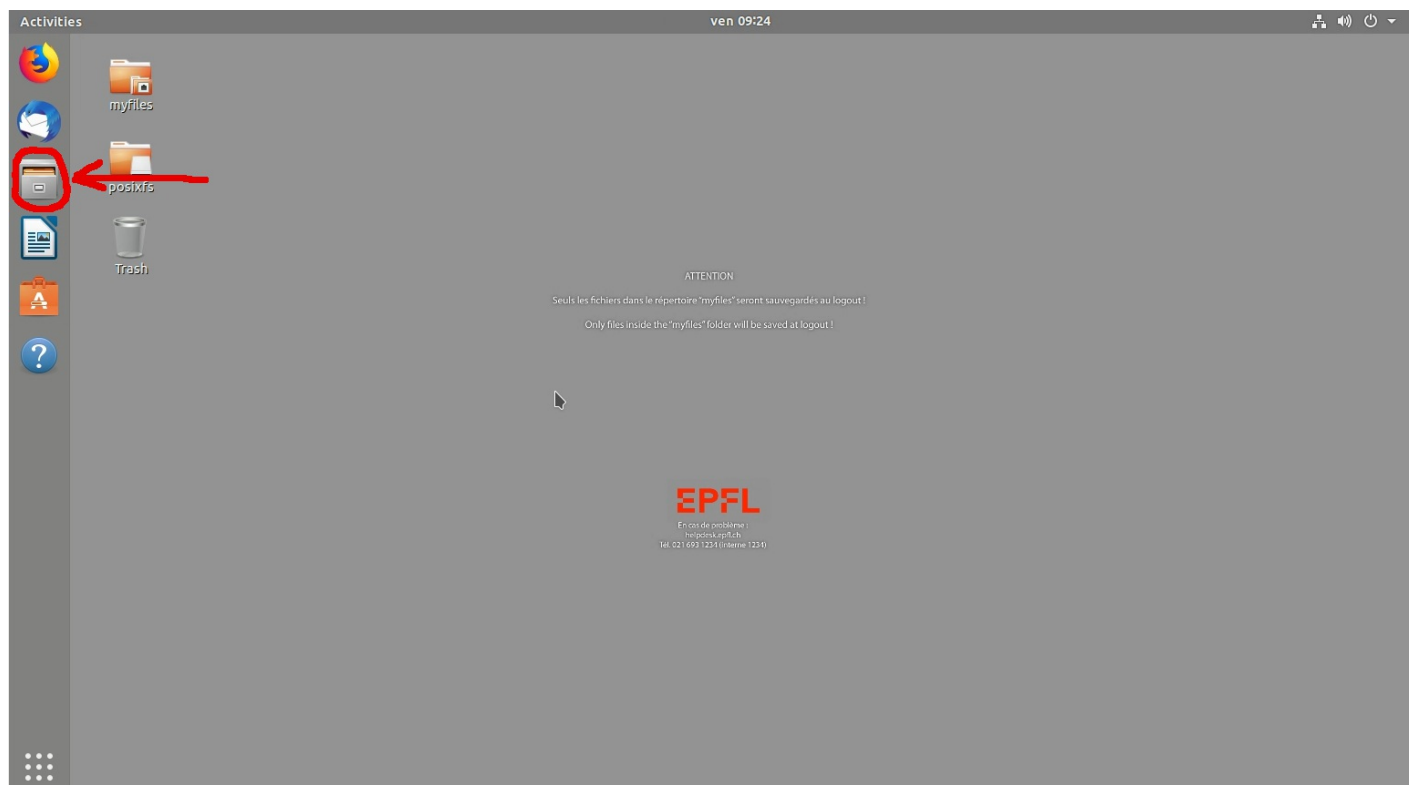
**Note importante :** les fichiers créés sur le disque local de la machine virtuelles (c.-à-d. en dehors de `myfiles`) sont effacés (= **perdus** !) en fin de connexion. Il est donc **impératif** que vous stockiez tous les fichiers que vous souhaitez conserver sous le répertoire `myfiles`, lequel se trouve sur votre bureau. Son nom relatif par rapport à votre « dossier home » est donc `Desktop/myfiles`, et son nom absolu est `/home/username/Desktop/myfiles` où `username` est votre nom d'utilisateur.

Une fois que vous avez trouvé un environnement de développement qui vous plait (conseillé: Geany), vous pouvez la mémoriser à gauche dans vos favoris comme indiqué pour le terminal.

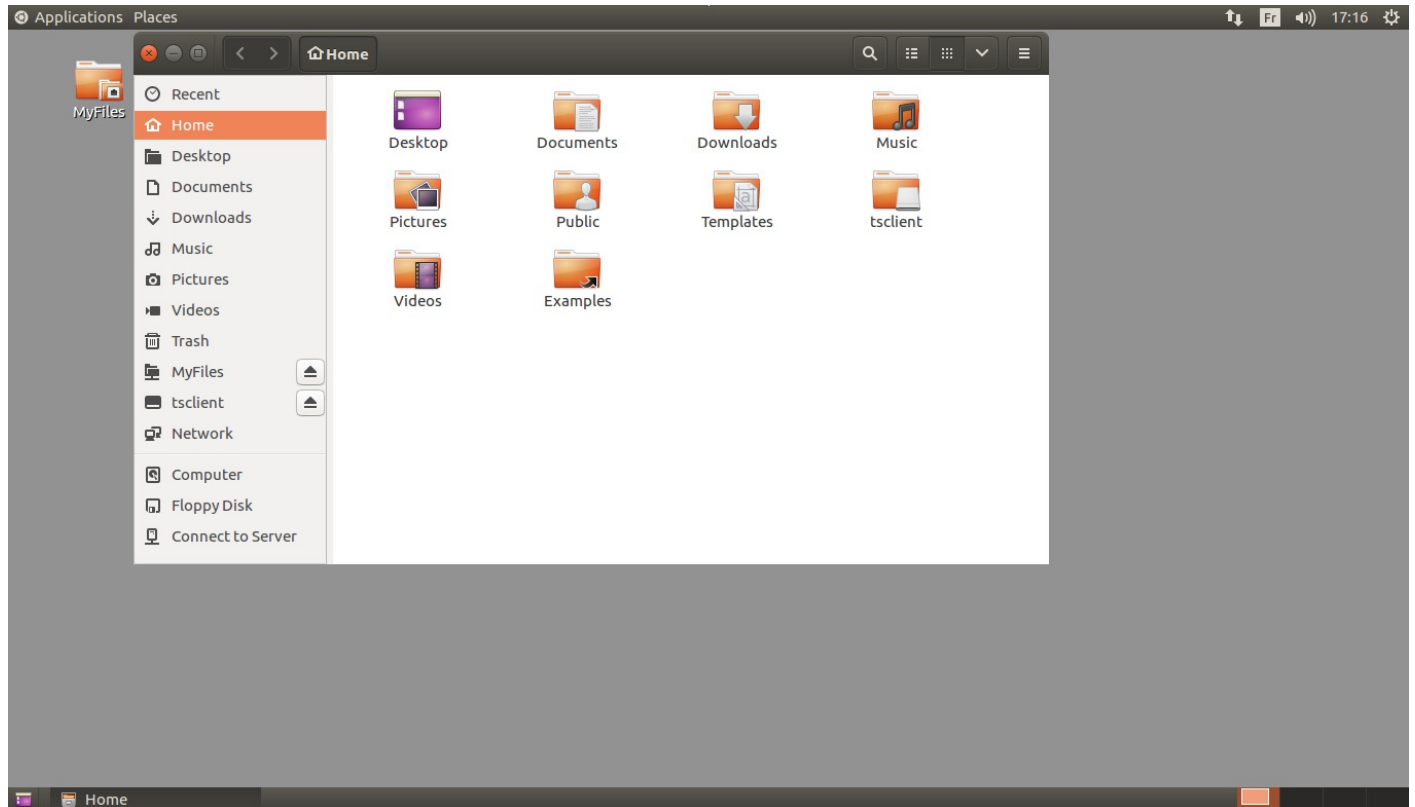
## Exercice 5 : Utilisation du gestionnaire de fichiers

Lorsque vous vous connectez sur votre compte, vous arrivez à la racine de votre répertoire personnel, c'est-à-dire à l'endroit de l'arborescence Linux qui vous appartient. Le chemin absolu de ce répertoire est `/home/username/` (où `username` est votre nom d'utilisateur). Il devient votre répertoire courant, c'est-à-dire le répertoire utilisé par défaut par les commandes Linux.

Pour ouvrir ce répertoire dans le gestionnaire de fichiers, sélectionnez l'icône «Home Folder» dans le menu «Places» en haut à gauche :



Vous verrez alors s'afficher une fenêtre qui ressemble à celle-ci :



Cliquez alors sur `Desktop`

Vous y verrez alors un répertoire nommé `myfiles` et un répertoire nommé `posixfs` qui sont extrêmement importants.

Ce sont, bien sûr, aussi ceux que vous voyez directement sur votre bureau, en haut à gauche.

### Quelques explications :

A l'EPFL, toutes les données liées aux utilisateurs sur un serveur central, le service myNAS (<https://mynas.epfl.ch>). C'est un service offert à tous les étudiants et collaborateurs de l'EPFL. Sauf erreur, le quota y est de 2 Go pour les étudiants. Ce répertoire est global à toute l'Ecole et suivra l'étudiant tout au long de ses études. Il est le même sous Windows, Mac et Linux dans les différentes salles de l'École.

#### Important :

- Sur les machines virtuelles (VM) utilisées ici, c'est ce répertoire `myfiles` qui vous donne l'accès à ces données.
- **Toutes les données stockées sur les VM en dehors de ce répertoire sont détruites à chaque fois que vous vous déconnectez!**
- Il est donc **impératif de systématiquement stocker dans `myfiles` tout ce que vous souhaitez garder**.

Pour ceux d'entre vous qui sont un peu plus avancés: cela signifie aussi que les fichiers de configuration et de préférence situés directement dans le répertoire principal («Home Directory») sont perdus à chaque déconnexion. Il faut, pour le moment, veiller à les archiver soi-même dans `myfiles` (et les restaurer au bon endroit).

Concernant la **PROGRAMMATION** cependant, le contenu *direct* de `myfiles` ne peut pas être utilisé sur les VM pour créer des programmes (technique : pas de possibilité d'y donner le droit `+x`). C'est pour cela que, *sur ces machines* (VMs salles CO) un répertoire supplémentaire `Desktop/posixfs` a été créé.

**C'est dans ce répertoire Desktop/posixfs** que vous devez programmer / stocker vos programmes. Le contenu de ce répertoire est également préservé d'une session à l'autre.

Deux remarques importantes à son sujet :

1. le contenu de ce répertoire n'est visible **que** sur les VMs ; il ne l'est pas sous Windows ni sur les autres machines par lesquelles vous cherchiez à accéder à `myfiles` ;  
  
si donc vous souhaitez transférer/rendre visibles vos programmes sur d'autres systèmes que les VMs, recopiez simplement les codes correspondant à un endroit de votre choix dans votre `myfiles` ;
2. ne touchez jamais, et en particulier ne détruisez pas, le fichier `Desktop/myfiles/fs/posix-1G-disk.fs` !!

Revenons maintenant à la fenêtre que nous venons d'ouvrir. Vous pouvez entre autres :

- parcourir simplement la structure des répertoires en double-cliquant sur les dossiers qui sont présents dans la fenêtre.
- créer de nouveaux fichiers ou répertoires en utilisant le menu qui apparaît avec le clic droit de la souris ;

**Note :** pour créer un nouveau fichier, vous pourrez aussi bien sûr utiliser un éditeur de texte, exactement comme vous l'avez fait dans **l'exercice 4** et sur lequel **nous allons revenir**.

- copier des fichiers ou des répertoires ;
- effacer des fichiers ou des répertoires en les sélectionnant avec le bouton gauche de la souris, et en sélectionnant «Move to Trash» dans le menu qui apparaît en cliquant sur la sélection avec le bouton droit (ou alors dans la barre de menu «Edit»). Vous pouvez aussi plus simplement appuyer sur la touche «Del».

**Note :** Pensez à vider régulièrement votre poubelle ! Je vois chaque année des étudiants bloqués par manque de quota disque parce qu'ils ont laissé trainer des gros fichiers dans leur poubelle.  
Vous pouvez pour cela :

- soit cliquer avec le bouton gauche sur le menu «File» du gestionnaire de fichier, puis cliquer sur «Empty Trash» ;
- soit cliquer avec le bouton droit sur l'icône de poubelle en bas tout à droite de l'écran (ce qui ouvre un menu) puis choisir «Empty Trash».

**Récupération de données perdues :** dans `myfiles` (uniquement), il vous est possible de retrouver des données indûment effacées (pour peu qu'elles aient eu le temps d'être archivées ; c.-à-d. qu'elles restent présentes au moins une heure, vu qu'il y a une archive que toutes les heures). Pour ce faire, il suffit d'aller dans le répertoire `.snapshot` à l'endroit de l'arborescence que vous souhaitez. Ce répertoire contient des archivages périodiques du contenu du répertoire courant.

**Attention !** Ce répertoire n'est pas automatiquement présent, mais chargé à la demande lorsque vous y accédez. Il est donc normal que vous ne le voyez pas. Pour y accéder :

- soit par la ligne de commande avec par exemple un  
`cd .snapshot`  
(ou autre, `cp`, etc., voir **l'exercice suivant pour les commandes unix**) ;

- soit depuis le gestionnaire de fichiers en tapant `Ctrl-L` et ajoutant `.snapshot` (puis 'Entrée') à la fin de la «Location».

## Application : Mise en place d'une architecture de travail

Vous allez maintenant créer une architecture de travail utile pour le reste de ce cours de programmation. Faites-le avec le gestionnaire de fichiers.

Dans le répertoire `posixfs`, créez le répertoire `cpp`.

Dans ce nouveau répertoire `cpp`, créez ensuite les sous-répertoires `serie01`, `serie02`, `serie03`, etc.

**Note** : on pourrait bien sûr le faire avec l'interpréteur de commandes (c.-à-d. dans un «terminal»), qui est justement l'objet du prochain exercice.

## Exercice 6 : Quelques commandes Unix (Linux ou Mac)

**Remarque préalable** : Il ne s'agit pas dans cet exercice d'uniquement répondre aux questions ni d'effectuer aveuglément les commandes données (ce qui est la plupart du temps trivial), mais plutôt de bien *comprendre* ce qui se passe et pourquoi on obtient tel résultat.

Pour gérer les fichiers, on peut soit utiliser le gestionnaire de fichiers comme vu précédemment, soit utiliser un éditeur comme déjà vu (et sur lequel nous reviendrons dans le prochain exercice), soit utiliser directement des commandes du langage *Shell* d'**UNIX** dans une fenêtre `Terminal`, ce qui est l'objet du présent exercice.

Les principales commandes (`ls`, `cd`, `cp`,...) sont présentées dans **la mini-référence** du cours.

### Application 1 : copie de fichiers

Ouvrez un terminal (voir **l'exercice 3**), et allez dans le répertoire `posixfs/cpp/serie01` créé à l'exercice précédent :

```
cd ## revient simplement à votre répertoire racine (home) si vous n'y étiez pas
cd Desktop/posixfs/cpp/serie01
```

#### Notes :

1. pour copier efficacement ces commandes depuis cette page vers votre terminal vous pouvez :

- triple-cliquez (gauche) sur la ligne de texte dans firefox ; cela sélectionne toute la ligne;
- cliquez-milieu (sur la roulette) dans le terminal ; cela « colle » la ligne précédemment sélectionnée.

N'hésitez pas à demander une démo à un(e) assistant(e) si nécessaire

2. dans le cas présent, une alternative au « copié-collé souris » est la *complétion automatique* (essayez les deux): dans le terminal lorsque vous tapez sur `TAB` (la touche de tabulation) le terminal essaye de compléter automatiquement le nom. Essayez par exemple de taper:

```
cd De
```

puis tapez sur `TAB`, le terminal devrait automatiquement remplir par `Desktop/` ;

tapez `p` (P minuscule) puis une nouvelle fois sur `TAB`, le terminal

devrait automatiquement remplir par `posixfs/` puisque dans `Desktop/` il n'y a rien d'autre que `posixfs/` qui commence par 'p' ;

tapez encore trois fois sur `TAB` pour arriver à «serie0» et terminez en mettant le numéro 1.

À l'aide des commandes Unix suivantes, copiez dans ce répertoire les fichiers `poeme1.txt` et `poeme2.txt` :

```
wget https://progmap.h.epfl.ch/doc/poeme1.txt
wget https://progmap.h.epfl.ch/doc/poeme2.txt
```

#### Notes :

1. Sur certains systèmes (Mac typiquement), utilisez `curl` à la place de `wget` :

```
curl https://progmap.h.epfl.ch/doc/poeme1.txt -o poeme1.txt
curl https://progmap.h.epfl.ch/doc/poeme2.txt -o poeme2.txt
```

2. Apprenez tout de suite à tirer avantage des fonctionnalités de la souris et du clavier dans cet environnement :

- pour recopier les commandes précédentes, au lieu d'essayer de le faire par vous même au clavier, utilisez le «copier-coller» de la souris comme expliqué ci-dessus et réexpliqué ici : il suffit de sélectionner avec la souris ce que vous voulez copier (par exemple avec un double clic gauche pour un mot, **triple clic** pour toute la ligne), puis de faire un clic-milieu (c.-à-d. bouton du centre) là où vous voulez copier ce qui a été sélectionné. Bref, en 4 clics vous avez cette commande compliquée qui est saisie !
- pour la seconde commande vous pouvez aussi (au lieu du «copier-coller») utiliser l'historique des commandes : dans le terminal, tapez sur la flèche vers le haut. La commande précédente réapparaît. Modifiez alors le 1 de `poeme1.txt` en un 2 et appuyez sur 'Entrée'. Simple, non ?

3. Les fichiers ci-dessus sont codés (pour les accents) en UTF-8. Si vous préférez un codage en latin-1 (ISO 8859-1), utilisez alors :

```
wget -O poeme1.txt http://progmap.h.epfl.ch/series/poeme1-latin1.t
wget -O poeme2.txt http://progmap.h.epfl.ch/series/poeme2-latin1.t
```

## Application 2 : commandes de base et diverses manipulations de fichiers

Créez, à l'aide de la commande Unix `mkdir`, un sous-répertoire `test` (dans `serie01` ou alors à la racine, hors de `myfiles`). Dans le premier cas le nom sera simplement `test` tout seul (ou `./test`, ce qui est pareil). Dans le second cas, ce sera `~/test`, avec le «tilde» devant. Ce «tilde» est un raccourci pour dire « `$HOME` », c.-à-d. votre répertoire personnel). Dans la suite nous utilisons `test`. Remplacez par `~/test` si vous avez créé ce répertoire à votre racine.).

1. Quelle est la différence entre les deux commandes suivantes :

```
cp poeme1.txt test/fl.txt
```

et

```
mv poeme1.txt test/fl.txt
```

?

2. Quelle est l'information retournée par la commande `which ls` ? par `which man` ? Que représentent-elles ?

3. Que vous donne la commande `ls -l *.txt` ?

**Note** : ci-dessus, c'est un « -L minuscule », pas un « -1 (-un) ».

4. Lors de l'utilisation de caractères spéciaux (\*, ? ou [ ]), il faut bien comprendre que c'est le Shell (c.-à-d. l'interpréteur de commandes) qui fait le remplacement des arguments *avant* d'envoyer la liste des arguments à la commande. Cela signifie que la commande ne « voit » que des arguments déjà évalués par le Shell. Mais comment tester ?

Cette partie de l'exercice vous propose de vous familiariser avec les commandes suivantes :

- `echo` affiche simplement ses arguments. Le nom vient du fait que la commande renvoie ses arguments comme un écho, sans modification :

```
echo 1 2 3
1 2 3

echo 'Bonjour tout le monde !'
Bonjour tout le monde !

echo SHELL
SHELL

echo $SHELL
/bin/bash
```

Dans ce dernier exemple, le Shell remplace la variable `$SHELL` par son contenu. La commande `echo` ne fait aucune modification de ses arguments.

```
echo *.txt
poeme1.txt      poeme2.txt
```

Dans ce dernier exemple, le Shell remplace `*.txt` par tout les fichiers correspondant, *puis* lance la commande `echo` avec les arguments trouvés. La commande `echo` reçoit donc bien comme arguments «`poeme1.txt poeme2.txt`» et ne fait elle-même aucune modification de ses arguments.

- `touch` modifie la date d'accès et la date de modification de chaque fichier indiqué. Les fichiers qui n'existent pas sont créés. On peut se servir du dernier fait pour créer des fichiers vides.

```
ls -l toto
ls: toto: No such file or directory

touch toto

ls -l toto
-rw-r--r--  1 dupont ma-bal 0 Sep 23 13:32 toto

rm toto
```

Dans cet exemple, on a créé et détruit le fichier `toto`.

**ATTENTION !** la commande `rm` ne met **pas** le fichier la poubelle mais le détruit de façon irrévocable. Une commande pour « mettre à la poubelle » existe sous forme d'un paquet Ubuntu : `trash-cli`. Ceux qui ont Ubuntu sur leur machine personnelle peuvent l'installer par `sudo aptitude install trash-cli`. Voir [cette page de documentation](#) pour plus de détails.

À présent :

a. Allez dans le répertoire `test` et créez plusieurs fichiers commençant par `f`, et au moins les trois *fichiers* (pas répertoires !) `f1`, `f2` et `f4` (c'est bien un `4`, ce n'est pas une faute de frappe).

Pour que ce soit plus clair ensuite, mettez quelque chose de différent dans chacun de ces fichiers (en

utilisant par exemple un éditeur).

Quel est le résultat de la commande `ls ??` (`ls` suivi de deux points d'interrogation (`??`)) ?

Quel est le résultat de la commande `ls f[123]` ?

Quel est le résultat de la commande `cat f[123]` ?

b. Créez quelques fichiers supplémentaires : `albert`, `a.txt`, `zoulou`, ...

c. Testez les caractères de substitution :

- Affichez tous les noms de fichiers avec `echo` ;
- Afficher les noms qui commencent par un `a` ;
- Afficher tous les noms de longueur 6 ;

5. L'exécution de la commande `ls -l` produit, pour un certain utilisateur, quelque part, le résultat suivant :

```
-rw-r--r--  1 dupont  ma-bal      75180 Mar 28  2003 divers
drwxr-x--x  2 dupont  ma-bal      4096 Mar 14  2003 autchose
```

- Est-ce que `divers` est un fichier ou un répertoire ? Même question pour `autchose`.
- À qui (utilisateur et groupe) appartient ce fichier ?
- Qu'est-ce que les utilisateurs du groupe `ma-bal` ne peuvent pas faire avec `autchose` ?
- Qu'est-ce qu'un utilisateur quelconque (pas dans le groupe `ma-bal`) a le droit de faire avec ce fichier/répertoire ?

6. Quelle est la spécificité des fichiers dont le nom commence par le caractère `.` ?

Exécutez les commandes suivantes (`~` représente votre répertoire racine («*home directory*»)) :

```
ls ~      puis
ls -a ~
```

Quelle différence observez vous ?

7. Allez dans un autre répertoire (quelconque, par exemple avec « `cd ..` ») et créez ensuite un « lien symbolique » (un alias) :

```
ln -s ~/Desktop/posixfs/cpp/serie01/poemel.txt nom2.txt
```

où `~/Desktop/posixfs/cpp/serie01/poemel.txt` est le nom d'un fichier qui existe quelque part (ici c'est un nom absolu, mais on peut aussi mettre un nom relatif) et `nom2.txt` est un nouveau nom (= alias).

Regardez ce qu'il en est :

```
ls -lF
```

`nom2.txt` est un lien symbolique vers le fichier `poemel.txt`. Ceci est très utile pour éviter de copier des fichiers (taille, intégrité, ...) tout en gardant un accès vers ceux-ci.

Notez que l'on peut faire des liens symboliques sur n'importe quel fichier auquel on a accès. Par exemple :

```
ln -s /usr/share/man/man1/ls.1.gz lsmanpage
```

Vérifiez en lisant ce nouveau lien :

```
gzip -cd lsmanpage | nroff -man | more
```

Dans votre répertoire de travail actuel, créez deux sous-répertoires : `rep1` et `rep2`.

Dans le sous-répertoire `rep1`, créez le fichier `bidon`.

Créez ensuite dans le sous-répertoire `rep2` un lien symbolique `bidule` sur le fichier `bidon` du sous-répertoire `rep1`.

Modifier maintenant le contenu du fichier `rep1/bidon` (en utilisant la commande `cat` ou alors en utilisant un



éditeur), puis lire le fichier `rep2/bidule`.  
Qu'observe-t-on?

Modifier ensuite le fichier `rep2/bidule`, puis lire le fichier `rep1/bidon`.  
Qu'observe-t-on?

Détruire finalement le fichier `rep1/bidon`, puis lire le fichier `rep2/bidule`.  
Qu'observe-t-on?  
Listez (`ls -l`) le contenu du répertoire `rep2`.

8. Pour finir, un petit exercice sur les noms de fichiers (et leurs raccourcis) :
    1. listez le contenu du répertoire `/etc` ;
    2. listez tout ce qui est dans le fils `man1` du fils `man` du père de `/usr/share/fonts` et qui commence par `rm`.
- 

## Exercice 7 : Utilisation d'un éditeur

Comme vu la [précédemment](#), vous avez plusieurs éditeurs ou IDE à votre disposition. Nous avons résumé les commandes essentielles de **Geany** (conseillé) et d'**Emacs** :

- [mini-référence Geany](#)
- [mini-référence Emacs](#)

mais vous êtes bien sûr libre d'utiliser un autre éditeur ou un autre IDE suivant votre choix.

Lancer un éditeur (ou un IDE) de votre choix et ouvrez-y les fichiers `poeme1.txt` et `poeme2.txt` (faites-le dans le même éditeur, c'est-à-dire sans lancer plusieurs fois **Geany**). Exercez-vous à modifier/manipuler le contenu de ces fichiers de différentes manières, par exemple en recherchant une chaîne de caractères quelconque et en la remplaçant par une autre. Essayez par exemple de supprimer tous les é (E accent aigu) du texte, ou de sauvegarder un nouveau fichier qui contienne les textes des deux fichiers. Apprenez aussi à annuler vos dernières modifications (*Edit -> Undo*).

Répéter cet exercice pour une autre éditeur de votre choix en prenant garde de ne pas ouvrir le même fichier avec deux éditeurs différents. Il est bien sûr possible d'ouvrir plusieurs fichiers dans un même éditeur. Chaque fichier occupe alors un «buffer» de l'éditeur. Assurez-vous donc d'ouvrir tous les fichiers dans le même éditeur après avoir pris soin de fermer l'éditeur précédent.

Comparer les fonctionnalités respectives des deux éditeurs :

- copier/coller
- chercher/remplacer
- ouverture, sauvegarde, fermeture d'un fichier
- fonctionnement des buffers (ouverture, fermeture, navigation)
- visualiser en même temps le contenu de deux fichiers différents

Pour les utilisateurs d'Emacs, nous vous conseillons également de suivre le didacticiel, en le lançant avec la commande **C-h t** (ce qui veut dire presser la touche **Control** en même temps que la touche **h**, puis la touche **t** mais **sans** la touche **Control** cette fois).

---

## Exercice 8 : Utiliser le «Forum de discussion»

Le [forum de discussion du cours](#) est accessible depuis la [page Moodle du cours](#) sous le lien « **ICC : Questions/Discussions** » (lequel va sur Piazza, la plate-forme que nous utilisons pour cela). C'est un outil important de ce cours et vous devez régulièrement le lire.

**Accédez à ce forum** et lisez le message de bienvenue que j'y ai posté. Ce n'est pas nécessaire d'y répondre : je n'aimerais pas avoir déjà 150 messages bidons sur le forum ;-) Par contre tout commentaire constructif ou toute question est déjà le/la bienvenu(e) !

N'hésitez donc pas à utiliser cet outil à bon escient : poster tous les messages qui vous semblent pertinents (questions, commentaires, ...) en relation avec le cours, le C++, Unix, la programmation, ou sur la partie théorie du vendredi, bien entendu.

---

## Exercice 9 : Navigation

Commencez par vous *inscrire* au [cours d'introduction à la programmation en C++ sur le site de Coursera](#) .

Veuillez si possible utiliser **votre adresse email EPFL** pour cela (sinon, envoyez moi ([jean-cedric.chappelier@epfl.ch](mailto:jean-cedric.chappelier@epfl.ch)) un email avec l'adresse utilisée).

Vous pouvez ensuite explorer les sites Web du cours et celui des salles CO :

- Naviguez sur **le site Moodle du cours**, parcourez en particulier **les références du cours** ;
- S'il vous reste du temps, butinez peut être sur le site **studinfo**,
- ou approfondissez vos connaissances en parcourant la **mini-référence de la semaine** (commandes Unix).

---

## Complément 1 : Utiliser le *mail*

Pour pouvoir accéder à votre compte e-mail, vous devez répondre aux **pré-requis mentionnés au départ de cette série**.

Voir ensuite la **la page d'accès à votre courrier électronique**.

Pour lire votre email dans les salles CO, vous pouvez soit passer par l'interface Web précédente, soit par un outil spécifique comme p.ex. «Thunderbird» (configuration : mot de passe GASPARD ; entrant : IMAP SSL/TLS sur [imap.epfl.ch](mailto:imap.epfl.ch), port 993 ; sortant : SMTP SSL/TLS sur [mail.epfl.ch](mailto:mail.epfl.ch), port 465).

---

## Complément 2 : Impression

### Visualisation des documents

Pour visualiser un document au format PDF, cet environnement utilise le logiciel *evince*.

Pour un fichier sur le Net, il suffit depuis votre navigateur Internet (firefox) de cliquer sur le lien vers un tel document. Un « *plug-in* » (une portion d'*evince* insérée dans le navigateur) se charge d'effectuer la visualisation.

Pour un fichier local, il suffit aussi de cliquer sur l'image dans **le gestionnaire de fichiers**.

Depuis la ligne de commande, on lance ce programme ainsi :

```
evince [document]
```

où *document* est un nom de fichier (absolu ou relatif) ; par exemple :

```
evince ~/Desktop/myfiles/exemple.pdf &
```

voire carrément une URL (adresse Internet) :

```
evince http://progmap.epfl.ch/slides/cours02.pdf &
```

### Impression des documents

Pour imprimer un document autre que PDF, vous devez au préalable le rapatrier sur votre système (pour les PDF votre navigateur le fera pour vous). Lorsqu'il s'agit de document que vous ne voulez pas conserver, utiliser le répertoire `/tmp` pour les stocker temporairement, ce qui permettra d'économiser votre quota disque.  
(Pensez tout de même à le nettoyer quand vous avez fini !)

Pour imprimer un fichier PDF utiliser le bouton d'impression dans *acroread*. De même pour imprimer une page Web, utilisez le bouton d'impression du navigateur (Firefox).

Pour plus de détails voir le site **myprint**.

### Application

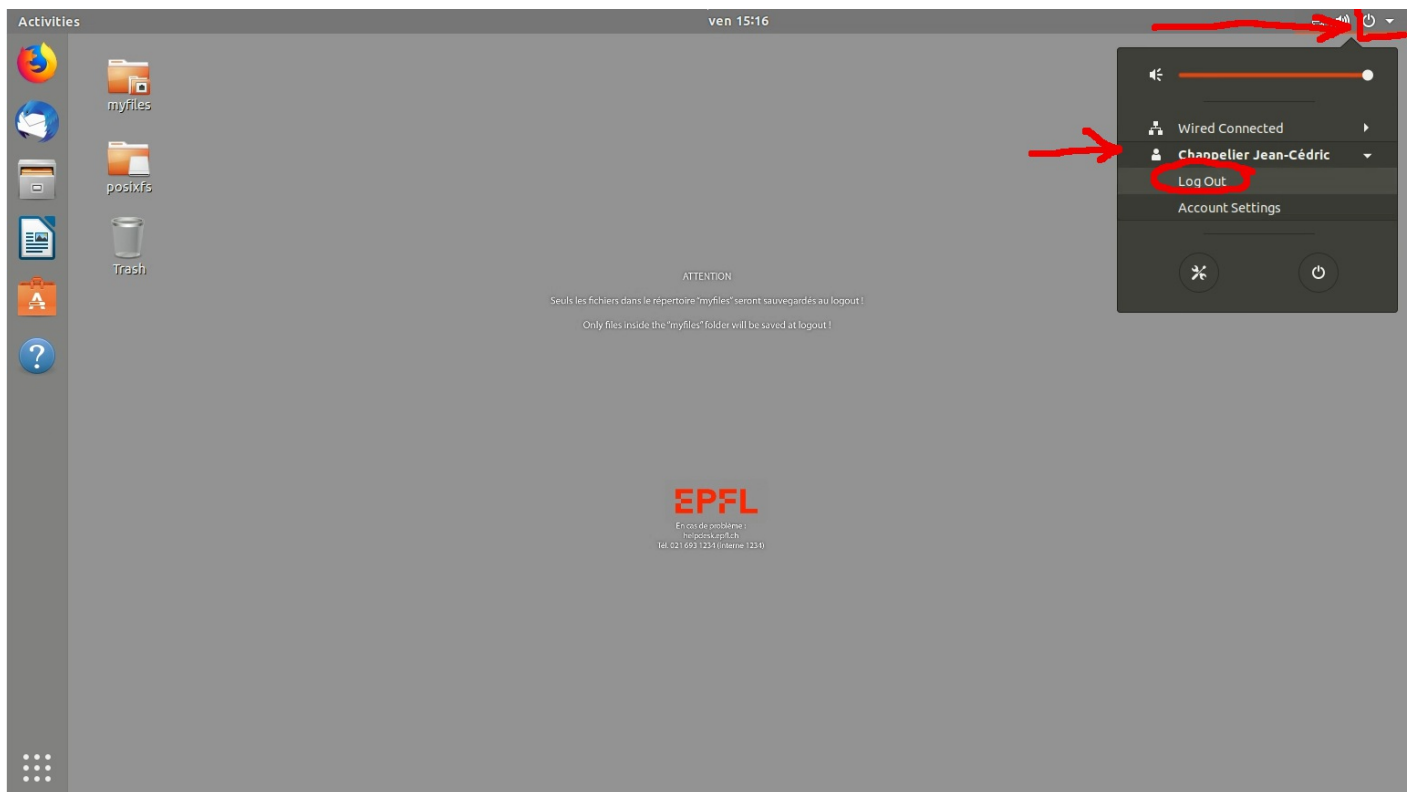
Pour voir si vous avez bien compris, essayez d'imprimer quelque chose, par exemple les transparents du cours (accessibles à l'URL <http://progmap.epfl.ch/slides/cours02.pdf>).

---

## Exercice 10 : Se déconnecter (ou *logout*)

**ATTENTION !** Pour vous déconnecter de votre session de travail, il vous faut vous déconnecter **DEUX FOIS !**

Une première fois de la machine virtuelle : option *Log Out* du menu sous votre nom dans le menu apparaissant en cliquant sur le bouton en haut à droite :



et une **seconde fois** du server de VMs, là où il y a la liste des choix des VMs (dont la IC-CO-IN-SC).

---

Dernière mise à jour le 8 septembre 2020  
Last modified: Tue Sep 8, 2020