


Home, compilation et configuration

— Ensimag 1A

Grégory MOUNIÉ (CC-BY-SA )

2024-2025

Le but de l'atelier est de travailler des notions importantes des systèmes d'exploitation comme les processus, la mémoire et les fichiers. Les différents résultats obtenus sont utiles ou rigolos, mais ils ne sont pas le but. Le but est de vous offrir une meilleure compréhension des concepts fondamentaux. N'hésitez donc jamais à poser des questions à votre encadrant.

Votre HOME est le répertoire dont vous êtes le propriétaire. Il contient vos fichiers, rangés dans des répertoires. Ces fichiers peuvent être vos documents, vos données, des fichiers exécutables, ou la configuration des logiciels que vous utilisez. Les manipulations proposées dans ce TP montrent un panel des possibilités.

Le TP est calibré pour occuper les plus rapides d'entre vous jusqu'à la fin de la séance. Une partie de l'exercice est que les plus rapides entraînent les autres pour que l'ensemble du groupe termine le sujet. Le sujet donne des indications générales avec un peu d'aide. Si vous êtes bloqué, vous trouverez à la fin du sujet la liste des commandes.

Ce document peut être téléchargé depuis l'adresse suivante :
<http://systemes.pages.ensimag.fr/unix-gitlab/atelier/home-compile-config.pdf>.

Ouvrez un terminal! Le gros du TP sera volontairement dans ce terminal.

1. Votre HOME

À la création de votre compte, un répertoire devient votre propriété (votre HOME). Ce répertoire est celui qui sera utilisé par les différents logiciels comme répertoire par défaut.

Taper les commandes `cd` puis `pwd` dans un terminal, vous permettent de lire sa position dans un terminal. À l'Ensimag, elle sera de la forme `/users/[123]/arecoque` pour l'utilisatrice Alice Recoque. Cette forme est un peu différente du `/home/arecoque` typique d'une installation Linux sur une machine personnelle.

1.1. L'organisation des répertoires avec vos données (XDG)

Comment est-ce que vos navigateurs, Firefox et Chrome, savent-ils où ranger vos téléchargements ? Comment les autres logiciels savent où chercher vos images, vos documents ou vos fichiers musicaux ?

Ouvrez un terminal.

Demandez au chat d'afficher les défauts de l'installation Linux à l'Ensimag `cat /etc/xdg/user-dirs.defaults`.

Votre personnalisation de ce défaut se trouve dans votre home dans le fichier suivant `~/.config/user-dirs.dirs` que vous pouvez aussi demander au chat.

Pour l'utiliser dans vos propres programmes, ou dans vos scripts shells, vous pouvez en obtenir facilement le répertoire où ranger les téléchargements la commande `xdg-user-dir DOWNLOAD`

`man xdg-user-dir` indique aussi les autres valeurs possibles.

1.2. Les répertoires et fichiers de configuration (fichiers et répertoires cachés, `.config/`)

Vous utilisez des centaines de logiciels. La configuration de chaque logiciel peut être personnalisée. Il faut donc un peu d'organisation pour éviter de remplir votre HOME de centaines de fichiers rangés n'importe où. Les règles de cette organisation du stockage des configurations a évolué au cours du temps : d'abord un fichier unique, puis un répertoire par application, enfin, le regroupement dans un sous-répertoire unique des fichiers ou répertoires de configuration de tous les logiciels.

Mais les configurations sont aussi là pour se faire oublier, donc elles se cachent.

1.2.1. Les noms des fichiers cachés commencent par un point

Affichez la liste des fichiers et répertoires cachés dans votre HOME (`ls -la ~/.*` ou `ls -lad .*`). Ils sont nombreux, non ? Et certains étaient là avant votre première connexion, dès la création de votre compte.

Affichez les fichiers et répertoires cachés dans le répertoire `/etc/skel/`. C'est ce répertoire qui a servi à faire le premier remplissage de votre HOME à sa création.

Vous pouvez comparer avec `diff` les contenus de votre fichier `~/.profile` avec celui de `/etc/skel/.profile`. Rien ne devrait s'afficher.

Votre fichier `~/.bashrc`, initialisant vos shells est probablement le fichier caché que vous changerez le plus à la main pour y ajouter, ou enlever, des configurations ad'hoc, en particulier de la variable d'environnement PATH.

Si vous avez lancé vim, demandez au chat¹ d'afficher sa configuration qui se trouve dans le fichier `~/.vimrc`.

1.2.2. Les noms des répertoires cachés commencent par un point

Pour les logiciels ayant une configuration complexe, ou téléchargeant des greffons avec des milliers de fichiers, un répertoire unique contient l'arborescence de l'organisation de sa configuration.

Si vous avez déjà lancé thunderbird, vscode ou emacs, vous pouvez afficher le contenu de `~/.thunderbird/`, `~/.vscode/` et `~/.emacs.d/`.

Affichez le nombre de fichiers dans le répertoire `~/.vscode` de VS Code avec `du --inodes ~/.vscode` (ou succinctement avec en plus l'option `-s`). Vous verrez dans le futur votre quota de fichiers être dévoré par les griffons, ... , pardon, les greffons.

1.2.3. Le répertoire `.config/`

Pour les quelques « gros » logiciels, avoir un répertoire de configuration dédié dans votre HOME est la solution la plus pratique. Elle permet par exemple de migrer votre configuration thunderbird d'une machine à l'autre, y compris en changeant de système d'exploitation.

Lancez la commande `ps` avec des options complètes comme `-efl` ou `auxc` et compter avec `wc -l` le nombre de processus (programme en exécution) de votre session actuelle.

Pour cette myriade de logiciels dont vous ne connaissez pas toujours le nom, comme ceux de votre environnement graphique Gnome, leurs configurations sont regroupées dans le répertoire `/.config/`. C'est maintenant le défaut, et la plupart des logiciels utilisent ou peuvent utiliser ce regroupement.

Listez le contenu de votre répertoire `~/.config`. Vous y trouverez aussi un répertoire `autostart` que nous utiliserons plus tard dans ce TP.

Si vous avez lancé néo-vim, vous pouvez trouver sa configuration dans `~/.config/nvim/init.vim`.

1.2.4. Le répertoire `.local`

Le répertoire `~/.local/` sert à y installer l'équivalent pour l'utilisateur de l'installation globale sur la machine : application, documentation, man, bibliothèques, etc.

C'est ce répertoire qui sera utilisé dans la partie suivante.

Vous pouvez retrouver l'ensemble de la convention d'utilisation des répertoires sous les Unix libres à <https://www.freedesktop.org/software/systemd/man/file-hierarchy.html>. Vous y trouverez le détail de conventions utilisées sous Linux.

La description de votre HOME est dans la 7ème section.

1. UUOC - [https://en.wikipedia.org/wiki/Cat_\(Unix\)](https://en.wikipedia.org/wiki/Cat_(Unix))

2. Installer et compiler un exécutable

Il est possible de faire l'installation d'un logiciel dans votre HOME, en tant qu'utilisateur standard. En effet, un logiciel est juste un ensemble de fichiers et vous pouvez créer arbitrairement des fichiers dans votre HOME. La principale limite à ces créations sont vos deux quotas : un maximum 10 Gi octets au total et un maximum de deux cent mille (200K) fichiers.

Une difficulté est la gestion des dépendances logicielles. Fabriquer votre logiciel demande l'utilisation d'autres logiciels et la disponibilité des bibliothèques qu'il utilise. Certains langages le font pour vous, mais vous aurez peut-être à installer d'abord chez vous ces dépendances à la main. Nous utiliserons ici un logiciel sans dépendances. Ce ne sera pas toujours aussi simple.

2.1. Récupérez les sources

Les sources du logiciel que nous utiliserons sont disponibles à <https://ftp.gnu.org/gnu/hello/>

Récupérer sous forme d'un fichier d'archive TAR compressé avec `wget` à l'url <https://ftp.gnu.org/gnu/hello/hello-2.12.1.tar.gz>. Extraire les sources et se déplacer dans le répertoire.

Comme vous le devinez, il s'agit d'un programme multilingue qui affiche « Hello World! ».

Les étapes suivantes vont consister à configurer sa fabrication, le fabriquer, l'installer, le faire fonctionner, puis le désinstaller.

2.2. Configurer le processus de construction pour installer dans son HOME

La première étape est de pouvoir le fabriquer. Cette étape permet de le compiler sur Linux, Windows, MacOS, les BSD, voir un PDP11 de 1970 (la machine sur laquelle a été conçu Unix et le langage C).

La configuration de la fabrication est le rôle du script `./configure` (créé par les `autotools`, `autoconf` et `automake`)

Vous pouvez obtenir les différents paramètres en lançant `./configure --help`

Configurer l'installation pour l'installer dans votre HOME dans le répertoire `${HOME}/.local/` avec l'option `--prefix=CHEMIN_DE_L_INSTALLATION`.

2.3. Compiler et installer dans son `~/.local/`

Le script `configure` a créé un fichier `makefile` adapté à votre machine.

Pour terminer la fabrication, il suffit de lancer la commande `make`.

Vous pouvez le tester en lançant `./hello`, mais il ne parlera qu'anglais.

Après son installation chez vous avec `make install`, il saura parler une quarantaine de langues.

2.4. L'exécuter en configurant votre environnement pour trouver le logiciel installé chez vous

La bonne nouvelle ? Normalement tout marche. Le répertoire `${HOME}/.local/bin` fait partie des répertoires classiques ajoutés automatiquement au PATH. (c'est dans votre `.profile`)

Lancez la commande `hello` et vérifiez le répertoire où le programme est lu avec `command -v hello`.

La mauvaise nouvelle ? Parfois cela ne marche pas si...

- le script d'installation vient juste de créer le répertoire `./local/bin`, il ne sera mis automatiquement dans votre PATH qu'au prochain login !
- vous utilisez Zsh, Zsh ne le met peut-être pas par défaut dans le PATH
- vous utilisez Zsh, il est bien dans le PATH, mais il faut penser à faire la commande `rehash`

Vous pouvez toujours résoudre temporairement ce petit problème en ajoutant à la main ce répertoire dans le PATH courant de votre terminal

```
1 alice@ensipc$ PATH=${PATH}:~/local/bin # puis 'rehash' si zsh
2 alice@ensipc$ hello
3 Bonjour, le monde !
4 alice@ensipc$ command -v hello
5 /VOTRE_HOME_PATH/local/bin/hello
```

2.5. Désinstaller

Les autotools font des scripts makefile très propres qui suivent bien ce qu'ils créent. Pour enlever le logiciel, il suffit de retourner dans le répertoire avec les sources et de lancer la commande `make uninstall`.

Nettoyer aussi le répertoire des résidus de la compilation avec `make clean`

3. Exemple de configuration avancée : Firefox, Emacs et org-capture

Cette partie du TP concerne la configuration d'interactions complexes entre différents logiciels. Nous allons configurer tout ce qu'il faut pour maintenir votre journal de bord efficacement.

Pour comprendre en profondeur l'intérêt d'un journal de bord pour un scientifique, vous pouvez essayer de suivre l'excellent MOOC sur la recherche reproductible de FUN (<https://www.fun-mooc.fr/fr/cours/recherche-reproductible-principes-methodologiques-pour-une-science-transparente/>)

En résumé, il sert à noter **succinctement** les événements et les choix au fur et à mesure de leurs déroulements. Pour la Science, avec un grand S, cela permet de retrouver,

comprendre et corriger les expériences en permettant de pouvoir retrouver à tout moment un résultat précédent. C'est une pratique commune en Physique et Biologie depuis deux siècles. En Informatique, c'est très récent.

Un journal est un fichier textuel unique dans lequel vous allez noter tout ce qui vous est arrivé : un concept, un site web utile, une tâche à faire plus tard, les décisions prises, le pourquoi de ces décisions, etc. Cela permet de retrouver, des jours ou des années plus tard, les informations utiles que vous y avez mis. Une entrée du journal ne fait souvent que quelques lignes.

Nous allons utiliser les capacités d'éditeur programmable d'Emacs pour rendre ce processus aussi léger que possible afin qu'une prise de note dans le journal soit la plus fluide possible et ne soit pas perçue comme une contrainte, mais bien comme une opportunité de retrouver ce que vous n'avez pas encore parfaitement mémorisé.

Le format textuel utilisé sera le format ORG. Il permet, dans Emacs, toute sorte de filtres et manipulations simples ou complexes, adaptés à un journal. Mais surtout cela reste du texte. Vous pourrez donc aussi l'ouvrir facilement avec toute sorte d'autres outils (grep), ou un autre éditeur, si besoin.

3.1. En bref, les actions à faire

- Configurer systemd pour qu'il lance emacs en tâche de fond au début de votre session, en mode serveur
- Configurer emacs pour qu'il enregistre vos captures dans un fichier spécifique. Cela vous permettra de garder facilement une trace de vos activités.
- Ajouter un plugin org-capture à Firefox
- Ajouter un traitement standardisé pour les URL du type "org-capture-protocol", qui sera utilisé par le plugin. Ce traitement discutera avec l'emacs lancé en tâche de fond.

3.2. Systemd

Systemd est le système de démarrage de votre Linux. C'est lui qui lance tous les autres programmes au début. Il lance aussi, plus tard, votre session utilisateur. Vous pouvez observer l'arbre de lancement de votre session avec `systemctl --user status`.

3.3. Emacs server et emacsclient

Demandez à systemd de lancer un serveur emacs au début de votre session par la commande `systemctl --user enable emacs.service`. Pour le lancer immédiatement, il faut le démarrer `systemctl --user start emacs.service` et voir que tout va bien par `systemctl --user status emacs.service`.

Vous pouvez demander à votre serveur d'afficher une fenêtre emacs instantanément avec la commande `emacsclient -c`.

Ajoutez dans votre `.bashrc` un alias pour démarrer emacsclient. Une ligne de la forme `alias ec='emacsclient -c'`

Démarrez un nouveau terminal et vérifiez que votre alias fonctionne.

3.4. Indiquer où se trouve votre journal à emacs

Dans votre HOME, créez un répertoire `~/org/`. C'est le répertoire utilisé par défaut par org-mode. Il est évidemment configurable, mais nous resterons avec le défaut pour diminuer la taille de la configuration.

Ajoutez (créez si besoin) dans le fichier `~/.emacs.d/init.el` les quelques lignes suivantes, pour :

- retarder la configuration d'org-mode, pour ne la faire que lorsque ce sera utile (defer)
- ajouter le raccourci clavier pour ajouter une entrée dans votre journal (« Control-c » puis « c »)
- Un patron (template "j") de capture interactive de votre nouvelle entrée dans le journal. Emacs écrira automatiquement si besoin l'année, le mois, le jour, l'heure de la note. Deux autres patrons "L" et "p" seront utilisés depuis votre navigateur web.

```
1 (use-package org
2   :defer t
3   :init
4   (global-set-key (kbd "C-c c") #'org-capture)
5   :config
6   (setq org-capture-templates
7     '(("j"
8       "=> Journal"
9       entry
10      (file+olp+datetree "journal.org")
11      "* %?\nEntered on %U\n %i\n %a")
12      ("L"
13       "org-capture: URL => Journal"
14       entry
15       (file+olp+datetree "~/org/journal.org")
16       "* %?\nEntered on %U\n [[%:link] [%:description]]\n")
17      ("p"
18       "org-capture: selection dans une page web => Journal"
19       entry
20       (file+olp+datetree "~/org/journal.org")
21       "* %?\nEntered on %U\n
   ↪ [[%:link] [%:description]]\n#+BEGIN_QUOTE\n%i\n#+END_QUOTE\n"))))
```

Vérifiez en lançant un nouvel Emacs que votre configuration a bien été retranscrite correctement. Puis relancez le serveur Emacs avec la commande `systemctl --user restart emacs.service`

En utilisant votre alias pour lancer un emacsclient, entrez votre première note. Pour la terminer, il faudra faire, comme indiqué par Emacs, « Control-c » puis « Control-c », ou

bien « Control-c » puis « Control-k » pour annuler la prise de note.

Vérifiez le contenu du journal.

3.5. Plugin Firefox

Une grande partie des informations que vous voudrez conserver dans votre journal viendront d'un navigateur web. Nous allons dans cette partie permettre de capturer une note directement depuis Firefox.

Ajoutez le greffon (plug-in) « org-capture » à Firefox.

3.6. Desktop et MIME

Maintenant, il va falloir expliquer à toutes les applications, dont Firefox, que votre emacsclient sera l'application chargée d'afficher les documents du type « org-protocol ».

Pour les détails, vous pouvez les trouver dans <https://orgmode.org/worg/org-contrib/org-protocol.html>.

La configuration a l'option `-c` qui fait apparaître une nouvelle fenêtre de Emacs serveur à chaque capture. Cela nous paraît moins troublant pour les débutants. Néanmoins, il n'y a ici qu'un seul serveur. Fermer une écriture dans une fenêtre a aussi une influence sur les autres fenêtres.

En bref, ajoutez à votre `~/.emacs.d/init.el` la ligne `(require 'org-protocol)`
Créez un fichier `~/.local/share/applications/org-protocol.desktop` contenant les lignes suivantes

```
1 [Desktop Entry]
2 Name=org-protocol
3 Comment=Intercept calls from emacsclient to trigger custom actions
4 Categories=Other;
5 Keywords=org-protocol;
6 Icon=emacs
7 Type=Application
8 Exec=emacsclient -c -- %u
9 Terminal=false
10 StartupWMClass=Emacs
11 MimeType=x-scheme-handler/org-protocol;
```

Puis mettez à jour votre base de données personnels sur vos applications.

`update-desktop-database ~/.local/share/applications/` (uniquement pour votre Gnome),

ou bien la commande générique

`xdg-mime default org-protocol.desktop x-scheme-handler/org-protocol`
(Dans tous les environnements graphiques respectant Freedesktop.org/XDG)

Dans les deux cas, vous devriez pouvoir vérifier que tout va bien avec `xdg-mime query default x-scheme-handler/org-protocol`

C'est prêt ! Plus qu'à essayer depuis votre navigateur.

4. En résumé

Pour rappel, notre but est de vous faire appréhender les subtilités des processus, des fichiers et de leurs interactions.

Dans ce TP, nous vous avons parcouru quelques aspects liés à votre HOME, à la personnalisation des logiciels que vous pouvez utiliser sur une machine en les installant directement dans votre HOME, et à la configuration impliquant différents logiciels (Emacs, Systemd, Firefox) et les spécifications neutres d'interopérabilités de FreeDesktop.org.

Les deux TP suivants sont centrés sur les machines virtuelles et les conteneurs.

A. Solutions

Cette section indique, dans l'ordre la liste des commandes correspondantes aux questions du TP. Vous pouvez aussi partir de ces commandes et reconstruire les questions.

Mettre les commandes à taper ici. Il me semble qu'il faudra un maximum d'environ 90 commandes non triviales pour tenir en 1h30 (1 minute par commande)

```
1  # HOME
2  cd
3  pwd
4  cat /etc/xdg/user-dirs.defaults
5  cat ~/.config/user-dirs.dirs
6  xdg-user-dir DOWNLOAD
7  man xdg-user-dir
8  ls -ad .*
9  ls -ad /etc/skel/
10 diff /etc/skel/.profile ~/.profile
11 cat ~/.bashrc
12 cat -l ~/.vimrc
13
14 ls -l ~/.thunderbird/
15 ls -l ~/.vscode/
16 ls -l ~/.emacs.d/
17
18
19 ls -l ~/.config/
20 ls -l ~/.config/nvim/init.vim
21
22 ls -l ~/.local/
23
24 # pas clair pour xdg-mime et xdg-open: mailcap ?
   ↪ ~/.config/mimapps.list ? autre ?
25 xdg-mime query default image/jpeg
```

```

26
27 # compiler, installer et configurer son home avec hello
28 wget https://ftp.gnu.org/gnu/hello/hello-2.12.1.tar.gz
29 tar xzvf hello-2.12.1.tar.gz
30 cd hello-2.12.1/
31 ./configure --help
32 ./configure --prefix=${HOME}/.local/
33 make
34 ./hello
35 make install
36 echo ${PATH}
37 hello
38 command -v hello
39 # alternative
40 PATH=${PATH}:~/local/bin # puis 'rehash' si zsh
41 make uninstall
42 make clean
43
44 # Un exemple de configuration complexe
45 # Tenir son journal (un journal de bord) est important pour vous,
46 ↳ mais
47 # son utilité est décalée dans le temps. Il va falloir nous faire
48 ↳ confiance,
49 # jusqu'à ce que vous en ayez besoin, lors de vos révisions
50 ↳ d'examens
51
52 systemctl --user status
53 systemctl --user enable emacs.service
54 systemctl --user start emacs.service # ou restart
55
56 emacs ~/.bashrc # ajouter l'alias ec='emacsclient -c'
57 source ~/.bashrc # ou simplement démarrer un nouveau terminal
58 ec
59 ec ~/.emacs.d/init.el # ajouter le template
60 systemctl --user restart emacs.service
61 firefox # ajouter le greffon org-capture
62 emacs ~/.local/share/applications/org-protocol.desktop # ajouter
63 ↳ application gérant org-protocol
64 update-desktop-database ~/.local/share/applications/
65 xdg-mime default org-protocol.desktop x-scheme-handler/org-protocol
66 xdg-mime query default x-scheme-handler/org-protocol

```