

Outils de travail (recommandés)

Soufiane Belharbi ^{1*}

*LITIS EA 4108, INSA de Rouen, Saint Étienne du Rouvray
76800, France

April 23, 2015

Résumé

Nous présentons dans ce rapport, et de manière pratique, quelques outils de travail qui rendent votre façon de travail plus facile. Dans une première partie, nous aborderons la connexion à une machine de salle TP de l'extérieur de l'INSA. Ensuite, nous présenterons un outil de rédaction qui est assez connu dans l'académie et la recherche (\LaTeX). Nous finirons par présenter quelques outils de travail collaboratif qui permettent de plusieurs personnes de travailler en même temps sur les mêmes documents.

Keywords: ssh, LaTeX, google drive, sharelatex, overleaf, git

Table des matières

1	Connexion de l'extérieur de l'INSA à une machine à l'intérieur de l'INSA	2
1.1	Connexion	2
1.2	Manipulation	3
2	Rédiger ses documents avec \LaTeX	7
3	Travail collaboratif	10
3.1	Outils de partage des rapports	10
3.2	Outils de partage code, rapport et tout type de fichier : le git . .	10
4	Questions ?	12

Chapitre 1

Connexion de l'extérieur de l'INSA à une machine à l'intérieur de l'INSA

1.1 Connexion

Pour se connecter à une machine à distance nous utilisons l'outil `ssh`.

- Linux : Installez OpenSSH. Sur votre machine, installez OpenSSH-client avec la commande : `sudo apt-get install openssh-client`. Sur la machine cible, c-à-d la machine de salle tp, il faut installer OpenSSH-server (c'est déjà fait par les administrateurs système). Plus de détails <https://help.ubuntu.com/lts/serverguide/openssh-server.html>.
- Windows : Utilisez l'outil PuTTY (<http://www.putty.org/>)
- Mac : C'est possible que c'est déjà installé (à chercher).

Vous ne pouvez vous connecter directement à la machine de tp, il faut passer par un *tunnel* qui est une autre machine à l'INSA. Tout est décrit dans <http://wiki.insa-rouen.fr/doku.php?id=insa:salles-tp:start>, ensuite *accès à distance*.

Il faut choisir quelle machine vous voulez vous y connecter. La liste de toutes les machines disponibles se trouve à <https://dsi.insa-rouen.fr/>, ensuite *Pédagogie, Salles de TP*. Vous trouvez les salles groupées par leurs localisations. Le nom des machines indique quel département a le droit d'y se connecter. Un STPI ne doit pas se connecter à une machine ASI. Vous aller voir quelques informations sur les machines, par exemple : le nom, l'état (allumée, éteinte), nombre d'utilisateurs connectés. Quand vous cliquez sur le nom d'une machine, une page d'informations s'ouvre. Les trois derniers ligne sont les plus importantes. Si la machine est éteinte, allumez la. Attendez jusqu'à ce que la machine soit allumée. Dans cette exemple nous utilisons la machine stpi-01. Si vous êtes à l'intérieur de l'INSA, utilisez l'accès direct :

1. Ouvrez un terminal
2. Tapez la commande : `ssh sbelharbi@stpi-01.insa-rouen.fr -X`, en remplaçant le login par le votre. Remarquez le `-X` à la fin, ça vous permet d'afficher des objets graphiques.
3. Tapez votre mot de passe. Si c'est la première fois, `ssh` vous demande si vous voulez vraiment vous connecter à cette machine. Tapez `yes`.

Maintenant, si vous êtes à l'extérieur, suivez ces étapes (tout le détails est dans http://wiki.insa-rouen.fr/doku.php?id=insa:salles-tp:acces_a_distance:start :

1. Création du tunnel : Ouvrez un premier terminal et tapez la commande : `ssh -L2200:stpi-01.insa-rouen.fr:22 sbelharbi@ssh.insa-rouen.fr`, ensuite votre mot de passe. Ce terminal sera le tunnel. Pendant toute votre connexion, ne jamais fermez ce terminal, sinon la connexion sera immédiatement fermée. La connexion est ouverte pendant 10h. Après cette période, elle sera fermée. Vous pouvez la ré-ouvrir.
2. Utilisation du tunnel : Ouvrez un *deuxième* terminal, et tapez la commande : `ssh -X -p2200 -o 'HostKeyAlias stpi-01.insa-rouen.fr' sbelharbi@localhost`, ensuite votre mot de passe. Maintenant la machine `tp` s'appelle `localhost` et pas `stpi-01.insa-rouen.fr` (redirection).

Maintenant, vous êtes connecté à votre machine `tp`.

Au moment de la connexion avec `ssh`, si vous trompez de mot de passe, vous avez trois tentatives. Au delà de ça, vous serez banni (interdiction de se connecter pendant quelques minutes). Pour éviter ça, arrêtez les tentatives après la deuxième avec `Ctrl+c`.

Après votre connexion, et avant de commencer de travailler, il faut vérifier la charge de la machine (CPU, RAM). Si la machine est surchargée, il ne faut pas travailler dessus parce que vous allez encore la surcharger et ralentir tout le système. Vous fermez tout et vous recommencez. Pour vérifier l'état de la machine utilisez la commande `htop` dans un terminal. Vous allez voir l'état des CPU et celle de la mémoire. Vous pouvez aussi utiliser la commande `top`. Pour sortir du `htop`, tapez `q`. Ne laissez pas cette commande ouverte, ça consomme assez de ressources.

1.2 Manipulation

Maintenant que vous êtes connecté à la machine vous pouvez commencer à travailler.

Lancement des programmes :

Une règle à respecter : *Ne jamais lancer des programmes en mode graphique.* (sauf si vous n'avez pas de choix). Vous allez pouvoir visualiser des objets graphique grâce au `-X` dans la commande `ssh`. Pourquoi cette règle?

- Ça nécessite un haut débit de connexion.

- Le temps de réponse est assez lent.
 - Ça surcharge la machine cible (CPU, RAM).
- Vous pouvez essayer une fois et regarder l'effet.

Vous allez travailler en mode terminal. Par exemple, pour lancer Matlab en mode terminal, utilisez l'option `-nodesktop`. Si vous ne voulez pas voir l'image de démarrage, utilisez l'option `-nosplash`. La commande est : `matlab -nodesktop -nosplash`. Maintenant vous êtes à la commande de Matlab. Pour quitter Matlab, tapez `exit`.

Exploration des fichiers

Pour explorer vos fichiers/dossiers, voici quelques commandes :

- `ls` : Lister le contenu d'un répertoire. Exemples : `ls`, `ls ./`, `ls -a`, `ls -l`, `ls -lha`, `ls rep1/rep2`, `ls ../rep4/rep5`, `ls ../../rep6`.
- `cd` : Changer de répertoire. Exemples : `cd rep1`, `cd ../rep4`, `cd ...`
- `rm` : Supprimer un fichier(s)/dossier(s). Exemples : `rm readme.txt`, `rm ../readme.txt`, `rm rep1/readme.txt`, `rm *.txt`, `rm -r rep1/`, `rm -r rep1`.
- `mkdir` : Créer un répertoire. Exemple : `mkdir rep12`, `mkdir ../rep13`.

Transfert des données

Vous pouvez transférer des données (fichiers, dossiers) en local comme à distance. Les commandes fonctionnent sous la forme : `commande source destination`.

- Local : via la commande `cp`. Exemples : `cp readme.txt ./rep1`, `cp ../rep12/readme.txt ./rep1`, `cp -r rep1 ../rep12`, `cp -r rep1/* rep2`.
- Distance : via la commande `scp`, ça permet de copier des données entre deux machines distantes. Si vous voulez copier un fichier qui se trouve dans la machine tp (son chemin : `/home/sbelharbi/STPI2_M8/TD3/readme.txt`) vers un dossier sur votre machine (chemin : `/home/TPS/M8/TD3/`), suivez les étapes suivantes :

1. Ouvrez un terminal.
2. Tapez la commande :

```
scp -P 2200 sbelharbi@localhost:/home/sbelharbi/STPI2_M8/TD3/readme.txt
/home/TPS/M8/TD3/
```

Ça nécessite le mot de passe. Le `-P 2200` est le numéro de port que vous avez utilisé le moment de création du tunnel.

Copier tout le dossier :

```
scp -r -P 2200 sbelharbi@localhost:/home/sbelharbi/STPI2_M8/TD3 /home/TPS/M8/
```

Un autre outil plus avancé qui permet de synchroniser les dossiers à distance pour copier seulement les données modifiées : `rsync` (à vous de chercher).

Travail multi-tâche

Quand vous faites une seule tâche, une seule fenêtre suffit. Par contre, quand vous faites plusieurs tâches à la fois, par exemple : éditer un fichier Matlab,

explorer des fichiers, rédiger votre rapport, dans ce cas il faut ouvrir une fenêtre par tâche ce qui est encombrant. Heureusement, il y a des outils qui permettent de créer des fenêtres à l'intérieur d'une seule fenêtre. Nous citons deux outils : **byobu**, **screen**. Nous vous conseillons **byobu** qui donne plus d'informations visuelles (plus d'ergonomie). Voici quelques indications pour **byobu** :

- Au tout premier lancement, **byobu** vous demande quel mode vous préférez, mode **screen** ou mode **emacs**. Si vous n'êtes pas familier avec **emacs**, le mode **screen** est préférable.
- Pour créer une fenêtre, tapez **Ctrl+a c**, c-à-d appuyer sur la touche **Ctrl** et **a** à la fois, ensuite vous relâchez, ensuite appuyez sur la touche **c** pour créer.
- Pour fermer une fenêtre, tapez **exit**.
- Pour naviguer dans les fenêtres : **Ctrl+a n** aller à la fenêtre suivante, **Ctrl+a p** aller à la fenêtre précédente. La navigation est circulaire.

Quand vous finissez d'utiliser une fenêtre, fermez là avec la commande **exit**. **byobu** est utilisé généralement pour lancer des programmes et les laisser tourner même après la fermeture du terminal. Mais vous êtes pas censé laisser rien tourner après que finissiez votre tp. Si par accident, vous fermez le terminal, **byobu** est tout les programmes lancés dedans restent tourner. Il suffit de taper **byobu** dans un terminal pour retrouver la session précédente. Nous allons pas parler de détachement de **byobu**. **byobu** peut jouer un rôle de sécurité. Si par exemple, vous éditez un fichier et vous n'avez pas encore sauvegardé les dernières modifications et soudainement vous perdez la connexion, vous pouvez vous connecter et trouver tout les programmes ouverts.

Sauvegarder *constamment* les modifications apportées sur vos documents.

Matlab

Pour coder/déboguer votre code Matlab sans l'interface graphique, vous avez plusieurs solutions :

1. Lancez Matlab, ensuite lancez à l'intérieur de Matlab l'éditeur graphique (mauvaise démarche) avec la commande Matlab : **edit ./monscript.mat**. Ça ouvre l'outil graphique d'édition de Matlab. Ceci est lent et ça prend du temps pour réagir, mais ça peut être un début. La commande **edit ./monscript.mat** peut éditer un fichier existant ou créer un nouveau.
2. Ouvrez deux fenêtres séparées (soit dans **byobu** ou dans deux terminaux séparés) : la première pour lancer Matlab et la seconde pour éditer le fichier Matlab avec n'importe quel éditeur de fichier texte, par exemple **emacs** avec la commande : **emacs -nw monscript.m** (**-nw** pour ne pas afficher de fenêtre). Il faut naviguer à l'endroit où vous créer le fichier, sinon il faut mettre tout le chemin du fichier. Maintenant, vous éditez le fichier avec la seconde fenêtre et vous le lancer dans la première fenêtre où Matlab est ouvert (il faut se positionner dans l'endroit où le fichier se trouve, naviguer à l'intérieur de Matlab avec les commandes : **cd**, **ls**, **mkdir**, **rm**. Pour lancer un script, soit vous tapez son nom **monscript** (sachant que vous êtes dans le bon endroit), soit vous utilisez la commande Matlab : **run('./monscript.m')**.

3. Utilisez l'éditeur **emacs**, ensuite divisez son écran en deux : une partie pour éditer le fichier, et la seconde partie pour lancer le script. (à vous de chercher).

Quand vous finissez le travail, veuillez à fermer tout les programmes et les terminaux.

Chapitre 2

Rédiger ses documents avec \LaTeX

\LaTeX est l'outil de rédaction le plus répandu dans le monde académique et celui de la recherche. En tant qu'étudiant à l'INSA, c'est hors question de rédiger ces documents avec des outils tel que word, libre office, open office . . . Vous devez vous mettre dès maintenant au \LaTeX . C'est assez simple : vous rédiger un fichier texte avec quelques commandes spécifiques, compilez et vous obtenez un fichier de très haute qualité avec le moindre d'effort de formatage.

\LaTeX est très connu par sa capacité et simplicité d'écrire des formulations mathématiques.

\LaTeX vous permet de vous concentrer sur le fond du document et passer très peu de temps sur la forme en indiquant juste quelques commandes.

Pour rédiger un document en \LaTeX , il suffit d'écrire un document texte (avec l'extension : `.tex`), *compiler* ce fichier avec la commande `pdflatex` (il se trouvent autres commandes de compilation pour obtenir différents formats de documents de sortie) et vous obtenez un fichier `.pdf` de haute qualité. Ce turotiel a été rédigé avec \LaTeX .

Pour installer \LaTeX sous Linux, tapez cette commande sur un terminal : `sudo apt-get install texlive-full`.

Pour éditer un fichier `.tex`, vous pouvez utiliser n'importe quel éditeur de texte. Parmi les éditeurs spécifiques à \LaTeX , nous citons `textmaker`. Pour l'installer, tapez cette commande : `sudo apt-get install textmaker`. `TeXworks` est un autre éditeur, pour l'installer : `sudo apt-get install texworks`.

\LaTeX permet de tout faire : écrire de texte, formules mathématiques, insérer des images, tableaux et tout ce que vous pouvez imaginer. Il suffit d'utiliser les *packages* nécessaires. \LaTeX est *packagé*.

\LaTeX permet de référencer des livres, article, siteweb, cours, . . . (bibliographie). Pour cela, utiliser la commande de compilation `bibtex`. Il faut utiliser la commande `pdflatex nomfichier.tex` ce qui génère un fichier `nomfichier.aux`. Ensuite, appeler la commande `bibtex nomfichier.aux`, ensuite appeler deux

fois la commande `pdflatex nomfichier.tex`. \LaTeX permet aussi de lister du code source.

\LaTeX permet aussi de faire des présentations (`beamer`, à chercher).

Voici un document \LaTeX minimal :

```
\documentclass{article}
\begin{document}
Mon premier document \LaTeX. Je suis content.
\end{document}
```

Pour plus de détails, <http://www.infotheque.info/cache/10172/supports.goualard.free.fr/files/master-agma/redaction/cours-latex-Master-2-ALMA.pdf>

Pour rédiger vos rapports, il suffit de créer une fois un modèle (un template), ensuite la ré-utiliser pour chaque rapport.

Voici quelques notes pour la rédaction de vos rapports :

- Utiliser \LaTeX
- Ne mettez pas d'image à la couverture du rapport. C'est réservé pour les magazines.
- Le titre doit être raisonnable et objectif.
- Ne mettez pas l'expression : *à l'attention de x*, ça donne l'effet d'une lettre administrative. Vous mettez le nom de la personne censé lire le rapport, votre enseignant ou autres personnes en spécifiant son lien avec le rapport.
- Référez les figures, tableaux et autres objets. En \LaTeX c'est facile, il suffit d'utiliser la commande `\ref{}`. Tout objet doit avoir une référence que vous pouvez utiliser plus tard pour le citer. Par exemple, quand vous insérer une image, vous mettez sa référence, ensuite citez la, ... dans la figure `\ref{fig:fig1}`, nous présentons la projection de ...
- Mettez que l'essentiel.
- Limitez vous à 10 pages maximum (pour le projet M8). Si nous demandons 10 pages, c'est hors question de rendre un rapport de 50 pages. Synthétiser votre travail.
- Ne mettez pas de code sauf si nécessaire.
- N'utilisez pas de couleurs. Quand vous tracez des courbes, utilisez différentes lignes.
- Numérotez les pages.

Vous trouvez, joint avec ce document, le dossier `template` qui contient un modèle exemple pour rédiger vos rapports. Il est composé de trois fichiers :

1. `rapport.tex` : ça contient le rapport.
2. `bibliographie.bib` : ça contient la bibliographie.
3. `insa.eps` : le logo de l'INSA.

Note :

En \LaTeX , nous pouvons créer un document *principal* (`main.tex`) qui fait appel à autres documents secondaires (`part1.tex`, `part2.tex`, ...) via la

commande `\input{fichier.tex}`. Ceci permet de faciliter le partage des fichiers en mode collaboratif où chacun travaille sur une partie séparément. (à chercher)

Chapitre 3

Travail collaboratif

Dans votre travail, vous serez ramenés à travailler avec d'autres personnes, et par conséquent la notion de *partage* se présente. Nous parlons des objets *partageable* tel que un rapport, code.

3.1 Outils de partage des rapports

Nous présentons dans cette section deux types d'outils qui permettent de partager des fichiers de type rapport.

Google drive :

`Google drive` permet de partager tout type de fichier. Par exemple, pour un fichier de type document (`.docs`). `Google drive` offre la possibilité de modifier le document en *parallèle* par plusieurs collaborateurs et en temps réel. Il suffit de partager le document et donner son lien.

Outils pour L^AT_EX :

Plusieurs sites web offrent la possibilité de créer un projet L^AT_EX, les modifier, compiler tel que `sharelatex` : www.sharelatex.com et `overleaf` : www.overleaf.com. `sharelatex` limite le nombre de collaborateurs. Nous vous recommandons `overleaf` qui a récemment introduit le `git`.

Pour `overleaf`, il suffit qu'une personne crée le projet, ensuite envoyer le lien vers ses collaborateurs pour pouvoir modifier le projet L^AT_EX. Ça permet aussi de le modifier via le `git`.

3.2 Outils de partage code, rapport et tout type de fichier : le git

Un outil assez puissant pour le travail collaboratif, le `git`. L'idée est de centraliser le projet dans un *dépôt*. Ensuite, chaque collaborateur récupère une

version du projet (`clone`, `pull`), faire ses modifications en local (`commit`), ensuite soumettre ses modifications vers le dépôt central (`push`).

Il existe des sites pour créer des dépôts `git` tel que `github` : `github.com`. À l'INSA, nous disposons d'un gestionnaire des dépôts `git` sur l'adresse : `monprojet.insa-rouen.fr`. Nous recommandons de l'utiliser, c'est plus pratique. Voici le lien de la documentation `http://wiki.insa-rouen.fr/doku.php?id=insa:monprojet:index`.

Il faut installer sur votre machine l'outil `git`, selon le système d'exploitation utilisé. Vous regarder ici : `https://git-scm.com/book/fr/v1/D%C3%A9marrage-rapide-Installation-de-Git`. (lien `https://git-scm.com/book/fr/v1/Démarrage-rapide-Installation-de-Git`) N'oubliez pas d'installer d'abord les dépendances.

Un outil proche du `git` : `svn`. (à chercher plus tard)

Chapitre 4

Questions ?

Pour poser vos questions, vous avez des forums sur moodle (moodle.insa-rouen.fr) dans chaque formation. C'est un bon utile de partage questions/réponses.