

Initiation à Linux, Gnome, Emacs et SciLab

Introduction

Le but de cette première séance de TP est de vous familiariser avec les outils que vous utiliserez pour programmer : le gestionnaire de fenêtres Gnome, le système d'exploitation Linux et les logiciels Emacs et SciLab.

Il est donc **indispensable** que chacun des points abordés dans cette séance soit parfaitement compris au plus vite, si vous voulez pouvoir utiliser la salle machine (disponible en libre service), ce qui est fortement recommandé pour compléter les séances de TP.

Notations : Dans la suite, tout texte écrit avec une police de caractère de largeur constante (par exemple, **comme ceci**) correspond à une commande et doit donc être tapé. Par contre, tout texte encadré par les signes inférieur et supérieur (<comme cela>, par exemple) correspond à du méta-langage et doit par conséquent être remplacé par un texte approprié. Ainsi, si on vous demande de taper la commande **rm** <nom-de-fichier>, vous devrez taper au clavier **rm** suivi d'un nom de fichier que vous choisirez.

On désignera de même les touches du clavier par le nom qui les recouvre, un tiret entre plusieurs noms signifiant que les touches correspondantes doivent être enfoncées simultanément : ainsi, si on demande de taper **Ctrl-x Ctrl-s**, il faut maintenir enfoncée la touche **Control** et enfoncer successivement les touches **x** puis **s**. **Cas particulier :** dans la suite, on désignera respectivement par les symboles \leftarrow , \rightarrow et \leftrightarrow les touches BackSpace, Tabulation et Enter (ou Return).

1 Ouverture d'une session de travail

Les machines que vous allez utiliser sont des PC dont le système d'exploitation est Linux. Ces machines ont pour nom **pc122n** ou **pc123n** selon la salle dans laquelle elles se trouvent, suivi d'un numéro variant de 1 à 15.

1.1 Utilisation de la souris

On peut définir une bijection entre la zone affichée à l'écran (et gérée par le gestionnaire de fenêtres) et la région sur laquelle la souris se déplace. L'image, par cette bijection, de la position de la souris est indiquée à l'écran par ce qu'on appelle *le curseur souris*. Lorsque ce curseur se déplace sur le fond (coloré), il a pour forme une flèche noire, et pointée vers la gauche et le haut. Lorsqu'au contraire il traverse une fenêtre de texte (comme une fenêtre de commandes ou d'édition), il a une forme proche d'un I dont le haut et la base sont incurvés vers la barre centrale. D'autres types de fenêtre peuvent engendrer différents types de curseur.

Vous remarquerez que la souris possède trois boutons. Le bouton de gauche est celui que vous utiliserez le plus souvent : lorsqu'on vous demande de "cliquer", on signifie par cela que

vous devez appuyer, puis relâcher un bouton (celui de gauche, si on ne précise pas le contraire) de la souris.

1.2 Connection sur un ordinateur

Avant de vous connecter, vous devez voir à l'écran une fenêtre indiquant le nom de votre machine, et vous demandant votre identifiant (ou nom d'utilisateur) et votre mot de passe. Tapez votre nom d'utilisateur, puis le mot de passe qui vous a été donné. Ce mot de passe n'apparaissant pas (pour des soucis de sécurité, chaque lettre est remplacée par une étoile), prenez votre temps pour le taper.

Attention : Aussi bien pour le nom d'utilisateur que pour le mot de passe, il convient de faire la différence entre les lettres minuscules et majuscules : vérifiez que la touche verrouillant les majuscules (nommée **Caps Lock** ou contenant un verrou avec une flèche vers le haut) n'est pas enfoncée (une diode verte en haut à droite du clavier, sous laquelle est écrit "Caps Lock" doit être éteinte). Faites aussi bien attention à différencier le caractère moins (-) du souligné (_).

Si vous n'arrivez pas à vous connecter, faites appel à l'enseignant.

1.3 Manipulation des fenêtres

Normalement, vous êtes maintenant connectés. Vous devez voir à l'écran un fond coloré, contenant en haut à gauche des icônes et deux panneaux de boutons, celui du haut étant le *tableau de bord*.

Commencez par ouvrir une fenêtre de commandes. Pour cela, vous pouvez utiliser le bouton du tableau de bord représentant un écran.

Il est aussi possible d'obtenir une fenêtre de commandes en utilisant le menu **Applications**, accessible par le premier bouton (à gauche) du tableau de bord. Lorsque vous cliquez sur ce bouton, il apparaît un menu déroulant, contenant plusieurs lignes. Cliquez ensuite sur la ligne nommée **Outils système**, ce qui fait apparaître un sous-menu. Choisissez dans celui-ci la ligne **Terminal**.

Quoi que plus compliquée, cette seconde méthode permet de lancer un grand nombre de logiciels, qui ne sont pas forcément associés à un bouton du tableau de bord (nous verrons à la section § 4 comment associer un de ces logiciels à un tel bouton).

Ouvrez une fenêtre de commandes, en utilisant une des deux méthodes présentées précédemment. Avant d'être utilisée pour découvrir les commandes Linux (cf. § 2), elle va nous servir à expérimenter les diverses manipulations que l'on peut effectuer sur les fenêtres. Pour cela, nous allons nous concentrer sur le cadre de la fenêtre plutôt que sur son intérieur.

- Positionnez la souris sur la (large) barre supérieure de la fenêtre, puis enfoncez le bouton gauche et déplacez la souris (en maintenant le bouton enfoncé). Vous constaterez que la fenêtre reste solidaire du curseur souris, et peut ainsi être déplacée.
- La même opération sur la bordure droite, gauche, inférieure ou supérieure du cadre permet de modifier la taille de la fenêtre **dans une seule dimension** associée à la bordure (hauteur pour les bordures inférieure et supérieure, largeur pour celles de droite et de gauche).

- Les quatre coins du cadre permettent de modifier la taille de la fenêtre **dans les deux dimensions** (hauteur et largeur).
- Pour finir, le coin droit de la barre supérieure de la fenêtre peut contenir deux ou trois boutons : l'un, contenant une barre basse, permet de fermer la fenêtre, qui reste désignée dans un rectangle du tableau de bord (un clic sur son nom la rouvre) ; le second, contenant une (ou deux) fenêtre(s) stylisée(s), permet d'agrandir la fenêtre de manière à remplir tout l'écran, ou inversement de revenir à la taille précédente ; le dernier, contenant une croix, détruit **définitivement** la fenêtre, et doit donc être utilisé avec précaution.

Faites quelques essais de manipulation avec la fenêtre que vous avez créée. Cette fenêtre devant nous être utile par la suite, évitez le dernier cas (bouton X) si vous ne voulez pas rouvrir une autre fenêtre. Vous pourrez constater que le curseur souris se modifie lorsqu'il est positionné sur le cadre de la fenêtre, prenant une forme qui représente l'action possible à cet endroit.

2 Présentation de commandes Linux

La fenêtre de commandes permet d'exécuter des commandes Linux, c'est-à-dire de donner des instructions au système d'exploitation. Le système d'exploitation signale qu'il attend vos commandes en affichant une invitation de la forme `<machine> :<répertoire>>`, indiquant le nom de la machine où vous êtes connectés et le répertoire où vous vous trouvez. Les commandes que nous allons vous présenter concernent principalement la sécurité, les répertoires et les fichiers.

2.1 Modifier le mot de passe

La première chose à faire (nous n'avons que trop tardé) est de modifier votre mot de passe. Pour cela, vous allez utiliser la commande `yppasswd`, suivie de `←`. Vous devrez alors entrer (à l'aveugle) l'ancien mot de passe, c'est-à-dire celui que vous avez utilisé pour vous connecter (ceci est une routine de sécurité). Ensuite, Linux vous demandera d'entrer deux fois de suite le nouveau mot de passe.

Si ces deux nouveaux mots de passe (tapés encore une fois à l'aveugle) sont identiques, l'ancien mot de passe sera remplacé par le nouveau. Sinon, l'ancien mot de passe sera conservé. **Lisez attentivement** le message terminant la commande : il vous indique (en anglais) si la modification du mot de passe a été réellement effectuée.

2.2 Manipulation de répertoires

Dans cette section, nous allons décrire rapidement la hiérarchie des répertoires sur les PC, et les commandes sous Linux correspondant aux répertoires. Nous vous proposons ensuite quelques manipulations pour mettre en pratique les notions présentées.

La figure 1 présente une partie de la hiérarchie des répertoires sur les PC. Les répertoires et les fichiers que vous n'utiliserez pas dans un premier temps ont été enlevés (et remplacés par ...) dans un souci de clarté.

Dans une telle hiérarchie, on appelle *chemin d'accès* à un répertoire R_1 depuis un répertoire R_2 , la branche (dans cette représentation) qui va du nœud correspondant au répertoire

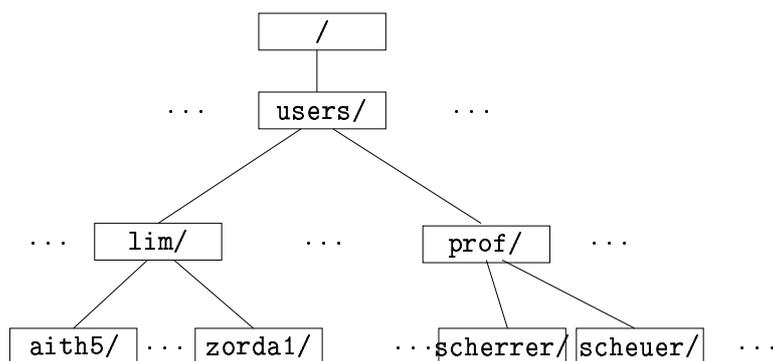


FIG. 1 – *Extrait de la hiérarchie des répertoires sur les PC.*

R_2 jusqu'à celui correspondant au répertoire R_1 . Pour écrire ce chemin, on utilise soit les noms des répertoires, soit `../` pour désigner le répertoire contenant le répertoire où on est arrivé.

Ainsi, le chemin le plus direct du répertoire `aith5` au répertoire `scheuer` s'écrit `../../prof/scheuer` : le premier `../` permet de remonter de `aith5` à `lim`, le second fait remonter de `lim` à `users`, puis on descend dans `prof` et `scheuer`. Cette écriture est la plus courte pour ce chemin, mais elle n'est pas unique : on peut aussi écrire ce chemin sous la forme `../../../users/prof/scheuer`, ou `../zorda1/../../prof/scherrer/..scheuer`, par exemple...

Un répertoire est habituellement désigné soit uniquement par son nom (s'il n'y a pas d'ambiguïté), soit par l'écriture du plus court chemin menant de la racine `/` jusqu'au répertoire en question.

Parmi ces répertoires, il en existe deux particulièrement importants : votre *répertoire principal*, qui contient tous les répertoires et les fichiers vous appartenant, et votre *répertoire courant*, qui est celui dans lequel vous êtes à un instant donné.

Le premier est fixe et correspond à `/users/lim/<votre-nom-d'utilisateur>`. Plus simplement, ce répertoire est représenté par le caractère `~`. Plus généralement, `~<utilisateur>` représente le répertoire principal de l'utilisateur `<utilisateur>`.

Le répertoire courant (`./`) correspond au départ, lorsque vous vous connectez, à votre répertoire principal. Pour savoir où vous êtes, vous pouvez utiliser la commande `pwd` ("Print Working Directory", afficher le répertoire courant). Cependant, vous pouvez changer ce répertoire, c'est-à-dire vous déplacer dans la hiérarchie de répertoires. Pour cela, il suffit d'utiliser la commande `cd` ("Change Directory", changer de répertoire). La commande `cd <répertoire>` vous emmène dans le répertoire `<répertoire>`, alors que la commande `cd` (sans argument) est équivalent à `cd ~` et vous ramène dans votre répertoire principal. **Remarque** : le nom des répertoires étant quelquefois un peu long, et leur orthographe ne vous étant pas toujours parfaitement connue, je vous invite à utiliser la touche Tabulation (`→|`) ; sa frappe est interprétée par Linux comme une demande de complétion : s'il n'y a qu'une seule possibilité de compléter ce que vous avez tapé, Linux le complète ; sinon, il vous affiche les différentes possibilités (rien si aucune complétion n'est possible). Cela vous permet de taper plus vite, et de vérifier que vous ne faites pas de faute de frappe (il n'y aura alors pas de complétion possible).

Parcourir des répertoires n'est vraiment intéressant que si on peut consulter leur contenu. Pour ce faire, on dispose de la commande `ls` ("LiSt content", énumérer le contenu), qui donne

la liste des fichiers et des répertoires contenus dans le ou les répertoires passés en paramètre, ou dans le répertoire courant s'il n'y a pas de paramètre. L'utilisation de l'option `-l` ("Long") permet d'avoir, en plus de la liste des fichiers et des répertoires, des informations quant à leur date de création et aux droits d'accès. On a alors, dans l'ordre, sept caractères indiquant la nature et les droits de l'objet (décrits par la suite), le nombre de points d'accès à l'objet (souvent 1), le nom d'utilisateur de son propriétaire et son groupe, sa taille en octet, sa date de dernière modification et enfin son nom. Les sept premiers caractères se décomposent en quatre groupes : un caractère donnant le type du fichier (`-` pour un fichier standard, `d` pour un répertoire, ...), trois caractères indiquant les droits du propriétaire, trois indiquant ceux du groupe et trois ceux des autres utilisateurs. Ces trois caractères sont `rwX` si tous les droits (`r` = "Read" / lecture, `w` = "Write" / écriture et `x` = "eXecute" / exécution) sont possédés, chaque droit absent étant remplacé par un tiret (`-`). Dans le cas d'un fichier, le dernier droit indique si ce fichier peut être exécuté (comme une commande). Dans le cas d'un répertoire, il indique si le répertoire peut être accédé (c'est-à-dire si on peut le traverser). Ces droits sont important lorsqu'on modifie les fichiers ou les répertoires.

Enfin, la dernière commande dont nous allons parler dans cette section est la commande `mkdir` ("MaKe DIRectory", créer un répertoire). Elle permet de créer un répertoire désigné par le paramètre. On doit **impérativement** avoir le droit d'accéder en écriture au répertoire dans lequel on va rajouter un élément (ici un répertoire), sinon la commande échoue. De même, la commande `rmdir` ("ReMove DIRectory", enlever un répertoire) ne fonctionne que si on a le droit d'accéder en écriture au répertoire dans lequel on va enlever un répertoire, et si le répertoire à enlever est vide (il faut le vider sinon, voir plus loin pour les fichiers).

Précisions sur la représentation des répertoires : Linux ne reconnaît que deux manières de désigner un répertoire : une représentation *absolue* et une *relative*. La première commence forcément par le caractère `/` (qui désigne la racine de la hiérarchie des répertoires) ou par un caractère `~` (suivi éventuellement d'un nom d'utilisateur). Si la représentation ne commence pas par un de ces deux caractères, c'est une représentation relative : on considère qu'il faut lui rajouter `./` au début, c'est-à-dire qu'il faut considérer qu'elle désigne le chemin partant du répertoire courant et menant au répertoire en question.

Ainsi, si vous êtes dans votre répertoire courant (un répertoire `/users/lim/<votre-nom-d'utilisateur>`), `/users/prof/scheuer`, `~scheuer` et `~/../prof/scheuer` sont trois représentations absolues du même répertoire, dont une représentation relative sera `../../prof/scheuer` (ou `../../prof/scheuer`, ou `../../users/prof/scheuer...`).

Application : Vous êtes, au départ, dans votre répertoire principal. Vérifier son nom, en tapant `pwd` \leftarrow .

Rendez-vous dans mon répertoire principal (`cd ~scheuer` \leftarrow). Consultez en le contenu (`ls`), puis allez dans `LicMath/Algo/`. Lorsque vous avez consulté le contenu de ce répertoire, retournez dans votre répertoire principal (`cd`), et créez un répertoire `Algo` (`mkdir Algo`), puis un répertoire `TP1` dans ce répertoire `Algo` (`mkdir Algo/TP1`).

2.3 Manipulation de fichier

Maintenant que vous savez manipuler les répertoires, voyons comment les remplir de fichiers (;-) ¹. Encore une fois, nous allons présenter les commandes de Linux en rapport avec les fichiers, puis nous vous en proposerons une mise en pratique.

La commande `more` permet d'en savoir plus sur un fichier : elle affiche son contenu dans la fenêtre de commandes. Il faut cependant **impérativement** avoir le droit d'accéder en lecture au fichier en question (*cf.* § 2.2, à propos de la commande `ls`). Lorsque ce contenu occupe plus de place que la taille la fenêtre, il est possible de le parcourir soit ligne par ligne (en tapant `↵`), soit page par page (en tapant sur la barre d'Espace). Il est alors aussi possible de revenir une page en arrière, en tapant `b` ("Backward", en arrière). Il est enfin possible de terminer la visualisation du fichier, en tapant `q` ("Quit", quitter).

La commande `cp` ("CoPy", copier) permet de copier un fichier. Le premier paramètre désigne le fichier *source* (le fichier d'origine), le second paramètre est soit un répertoire (le fichier résultat aura alors le même nom que le fichier d'origine), soit un fichier. Il faut, ici encore, pouvoir accéder en lecture au fichier source, et en écriture au répertoire dans lequel on ajoute un fichier.

La commande `mv` ("MoVe", déplacer) permet de déplacer un fichier, de la même manière que la commande `cp`. Il faut ici pouvoir accéder en écriture au répertoire du fichier source (qui sera enlevé) et au répertoire de destination. Il est possible que le répertoire source et celui de destination soient confondus : cela permet de changer le nom d'un fichier dans un répertoire (nous verrons un exemple de ce cas dans la mise en pratique).

Enfin, la commande `rm` ("ReMove", enlever) permet de supprimer le fichier passé en paramètre. Il faut bien sûr avoir les droits d'écriture pour le répertoire contenant le fichier en question. En fait, la commande `mv <fichier1> <fichier2>` est équivalente à la succession des commandes `cp <fichier1> <fichier2>` et `rm <fichier1>`.

Application : Vous êtes normalement dans votre répertoire principal (vérifiez le). Copiez le fichier `~scheuer/LicMath/Algo/debut.sci` dans le répertoire `Algo/TP1/` que vous avez créé dans votre répertoire principal. Il y a pour cela plusieurs méthodes :

- de votre répertoire principal, tapez `cp ~scheuer/LicMath/Algo/debut.sci Algo/TP1/` (cela me paraît le plus simple) ;
- sinon, aller dans le répertoire `Algo/TP1/` (`cd Algo/TP1/`) puis copiez le fichier dans le répertoire courant (`cp ~scheuer/LicMath/Algo/debut.sci ./`) ;
- enfin, dernière solution, allez où se trouve le fichier (`cd ~scheuer/LicMath/ Algo/`) puis copiez le fichier où il faut (`cp debut.sci ~/Algo/TP1/`).

Il existe encore beaucoup d'autres méthodes, choisissez en une et vérifiez son bon fonctionnement en comparant le contenu du fichier source et du fichier obtenu (`more ~scheuer/LicMath/Algo/debut.sci`, `more ~/Algo/TP1/debut.sci`). Essayez, à cette occasion, les commandes sous `more` (Return, Espace, `b` et `q`).

Faites une copie de ce fichier, et nommez la par exemple `copie.sci` (rappelons qu'il ne faut **jamais** utiliser d'espace dans un nom sous Linux, celui-ci serait alors considéré comme deux noms différents) : `cp debut.sci copie.sci` avec comme répertoire courant `~/Algo/TP1/`.

1. Pour ceux d'entre vous qui ne connaissent pas ces signes, ils forment ce qu'on appelle un "smiley" (sourire) tourné vers la droite, et indique une pointe d'humour. Il en existe plusieurs variantes...

Renommez ce fichier de copie (`mv copie.sci tmp.sci`), déplacez le (`mv tmp.sci ~`), puis détruisez le (`rm ~/tmp.sci`).

2.4 Pour aller plus loin

Pour utiliser la fenêtre de commandes le plus efficacement possible, on a déjà vu l'utilisation possible de la touche Tabulation. Ajoutons à cela la possibilité de se déplacer dans l'historique des commandes : on peut revenir à une commande précédemment tapée en tapant **Ctrl-p** ("Previous", précédent) ou \uparrow autant de fois que nécessaire ; **Ctrl-n** ("Next", suivant) ou \downarrow permet d'accéder aux commandes qui ont suivies celle à laquelle on est revenue, et **Ctrl-f** ("Forward", en avant) ou \rightarrow et **Ctrl-b** ("Backward", en arrière) ou \leftarrow permettent de se déplacer dans la ligne de commande pour en modifier le contenu (grâce à la touche Backspace).

On peut aussi taper **Ctrl-u** ("Undo", défaire) pour vider la ligne de commande, et **Ctrl-l** ("Clear", nettoyer) pour nettoyer la fenêtre de commandes toute entière (sans pour autant perdre l'historique des commandes).

Enfin, **Ctrl-c** permet d'interrompre toute commande (en cours de frappe ou d'exécution), alors que **Ctrl-z** permet de suspendre un commande (en cours d'exécution, nous verrons un exemple d'application dans la section suivante).

Nous allons arrêter là la description de commandes Linux (cela devrait vous suffire). Si vous souhaitez plus d'information à propos d'un commande `<commande>`, vous pouvez accéder à la section du manuel concernant cette commande en tapant `man <commande>` (attention, les explications sont en anglais et quelquefois compliquées).

Remarque : pour bien mémoriser les quelques commandes présentées dans cette section, n'hésitez pas à rédiger un petit mémento reprenant leur syntaxe et leur signification.

2.5 Pour aller moins loin

Certains d'entre vous regretteront sans doute l'interface graphique disponible sous Windows. Je les rassure : il existe une interface équivalente, qui peut être lancée par le bouton en forme de maison dans le tableau de bord, ou par l'icône de même forme sur le bureau.

Libre à vous d'utiliser cette interface. Faites cependant **très attention** à ne pas déplacer simplement un fichier, car l'action associée est rarement clairement définie. Il est nettement préférable de cliquer sur ce fichier avec le bouton droit de la souris, ce qui permet d'accéder à un *menu contextuel* vous fournissant plusieurs actions à effectuer sur le fichier choisi (copier, couper, coller, supprimer, etc.).

2.6 Transférer vos fichiers

Si vous souhaitez travailler chez vous, vous devrez transférer sur votre ordinateur personnel les fichiers nécessaires. Pour cela, vous disposez de plusieurs moyens.

Le plus simple est d'utiliser internet, en envoyant à votre adresse personnelle (par exemple grâce au service `etumail` de l'UHP) des documents attachés.

Si vous ne disposez pas d'une connexion chez vous, vous pouvez utiliser les lecteur de disquettes (standards ou zip) de ces PC. Après avoir introduit une de ces disquettes dans

l'ordinateur, cliquez le bouton droit sur le bureau, accédez au sous-menu **Volumes** du menu contextuel et sélectionnez le volume voulu (**Disquette** ou **Lecteur Zip**). La disquette correspondant est alors intégrée à la hiérarchie des fichiers (au travers du répertoire `/floppy` ou `/zip`) et peut être manipulée. Ce résultat peut aussi être obtenu par la commande `mount <répertoire>`.

Lorsque vous avez fini les transferts depuis ou vers la disquette, **ne la sortez surtout pas** tout de suite : certaines opérations que vous avez demandées n'ont pas forcément été effectuées. Il faut signaler au système qu'on va retirer la disquette, soit en la désélectionnant dans le sous-menu **Volumes** du menu contextuel du bureau, soit en sélectionnant la commande **Éjecter** du menu contextuel de l'icône de la disquette (sur le bureau), soit en tapant la commande `umount <répertoire>`. Chacune de ces actions fait disparaître du bureau l'icône de la disquette, après éventuellement quelques secondes nécessaires à la mise à jour de la disquette. Vous pouvez alors sortir votre disquette.

3 Édition sous Emacs

Emacs est un éditeur de texte très puissant, dont nous n'utiliserons sans doute pas toutes les possibilités. Pour l'invoquer, vous pouvez soit utiliser le menu **Applications** du tableau de bord, soit taper une commande du système d'exploitation.

Si vous souhaitez utiliser le menu **Applications** du tableau de bord, ouvrez celui-ci, puis parcourez ensuite les sous-menus successifs **Menu Debian**, **Apps** et **Editors**, avant de sélectionner la ligne **Emacs 21**. Cet accès étant long, il sera plus simple pour les autres séances de disposer d'un bouton dans le tableau de bord lançant cette application. Pour cela, sélectionnez le menu contextuel du tableau de bord (d'un clic avec le bouton droit). Choisissez le sous-menu **Ajouter au tableau de bord**, puis celui **Lanceur depuis le menu**, avant de parcourir les sous-menus indiqués précédemment pour atteindre la ligne **Emacs 21**.

Si vous préférez utiliser une commande, tapez `emacs21 &` dans une fenêtre de commandes. **Si vous avez oublié le & à la fin**, vous perdez la main dans la fenêtre de commandes. Plutôt que d'interrompre la commande (`Ctrl-c`) puis de la corriger (`Ctrl-p &`), vous pouvez aussi suspendre son exécution (`Ctrl-z`), ce qui vous rend la main, et la mettre en tâche de fond par la commande `bg` ("BackGround"), ce que fait normalement le `&` en fin de commande.

Cette commande crée une fenêtre d'édition, dont le titre (écrit sur la barre supérieure) est `emacs21@<machine>`. Cette fenêtre contient des menus déroulant juste sous la barre supérieure (nommés entre autres **File**, **Edit**, **Options**, **Buffers**, **Tools**, ..., **Help**), un ascenseur le long du coté gauche et deux lignes en bas de la fenêtre (une noire, une blanche).

Vous remarquerez que le curseur souris est le même sur cette fenêtre que sur toute autre fenêtre de texte (comme la fenêtre de commandes). Emacs possède cependant un second curseur, le curseur texte, qui indique où le texte tapé au clavier va apparaître. Ce curseur est un rectangle noir de la taille d'un caractère. Il est important de bien faire la différence entre ces deux curseurs.

Notations : Dans la suite de cette section, nous utiliserons quelque fois la notation **Meta- <lettre>**. Cela veut dire qu'il faut taper soit sur la touche **Escape** (**Echap.**) puis sur la lettre `<lettre>`, soit simultanément sur **Alt** et la lettre `<lettre>`.

3.1 Lecture et écriture d'un fichier

Pour charger un fichier dans l'éditeur, tapez **Ctrl-x Ctrl-f** ou utilisez l'option **Open File...** du menu **File** ("Open file", ouvrir un fichier). Il apparaît alors dans la dernière ligne (sous la ligne noire) le message **Find file:** <répertoire-courant>, suivi du curseur texte. Modifiez ou complétez le texte (avec **Backspace** et des caractères) pour indiquer le fichier **debut.sci** contenu dans votre répertoire **~/Algo/TP1/**. N'hésitez pas à utiliser la Tabulation (elle fonctionne ici aussi).

Lorsque vous tapez **Return**, Emacs charge le fichier à éditer dans ce qu'on nomme un buffer : on peut charger plusieurs fichiers, et basculer de l'un à l'autre en utilisant le premier menu déroulant (**Buffers**) ou en cliquant avec la souris tout en maintenant la touche **Control** appuyée. La ligne noire en bas de l'éditeur contient alors le texte suivant : les caractères **-:**, deux caractères (**-** si le fichier n'a pas été modifié depuis sa dernière écriture sur le disque, ****** s'il l'a été et nécessite donc d'être sauvegardé ou **%%** si le fichier ne peut pas être modifié ; il faut alors le sauvegarder sous un autre nom, voir un peu plus loin), deux caractères blancs puis le nom du fichier chargé. On trouve un peu plus loin, entre parenthèses, le mode sous lequel Emacs se trouve (le mode est un environnement spécifique à Emacs, définissant des commandes spéciales, et généralement associé à un type de fichier), deux caractères moins (**-**), le numéro de la ligne sur laquelle se trouve le curseur texte et le pourcentage de texte que représente la région entre le début du texte et le haut de la page.

Dans le cas qui nous intéresse maintenant (le fichier **debut.sci**), la ligne noire doit présenter le texte suivant : **-- debut.sci (Scilab Fill)--L1--Top-----** (les derniers caractères moins allant jusqu'à la fin de la ligne). Vous pouvez donc modifier le fichier, vous ne l'avez pas encore fait, il s'appelle **debut.sci**, correspond au mode **Scilab** avec indentation automatique et votre curseur texte est à sa première ligne, en haut du fichier ("Top").

Si les deux caractères suivant **-:** sur la ligne noire ne sont pas **%%**, vous pouvez modifier ce fichier. Vous pourrez alors, à tout moment, sauvegarder votre fichier (sous son ancien nom) en tapant **Ctrl-x Ctrl-s** ou en utilisant l'option **Save Buffer** du menu **File** ("Save buffer", sauvegarder la copie). Vous verrez alors apparaître dans la dernière ligne le message **Wrote** <nom-complet-du-fichier>.

Sinon, ou si vous souhaitez sauvegarder votre fichier sous un autre nom, tapez **Ctrl-x Ctrl-w** ("Write buffer", écrire la copie) ou utilisez l'option **Save Buffer As...** du menu **File**. Le message **Write file:** <répertoire-courant>, suivi du curseur texte, apparaît alors dans la dernière ligne ; spécifier un nom de fichier comme pour la commande de sauvegarde. Si le fichier existe déjà, Emacs vous demandera (en anglais et dans la dernière ligne) de confirmer que vous voulez effacer ce fichier. La sauvegarde effectuée, vous verrez alors apparaître le même message que celui concluant la sauvegarde précédente (**Wrote** <nom-complet-du-fichier>), et la ligne noire sera modifiée en conséquence : elle commencera alors par **-- <nom-du-fichier>**.

Remarque : Pour ces trois commandes, vous pouvez, si vous préférez, utiliser le menu déroulant "File" (en haut à gauche) plutôt que des raccourcis au clavier. La commande invoquée est alors la même (elle produit les mêmes résultats sur les deux lignes du bas, que ce soit des affichages ou des questions).

3.2 Déplacements dans un fichier

Il vous est possible de vous déplacer dans le fichier affiché à l'écran en tapant les raccourcis clavier suivants :

- **Ctrl-f** (“Forward”, en avant), ou \rightarrow , permet d'avancer d'un caractère vers la droite ou de passer à la ligne suivante si le curseur texte a atteint la fin de la ligne ;
- **Ctrl-b** (“Backward”, en arrière), ou \leftarrow , permet d'avancer d'un caractère vers la gauche ou de passer à la ligne précédente si le curseur texte a atteint le début de la ligne ;
- **Ctrl-p** (“Previous”, précédent), ou \uparrow , permet de passer à la ligne précédente si le curseur texte n'est pas sur la première ligne (“bip” sinon) ;
- **Ctrl-n** (“Next”, suivant), ou \downarrow , permet de passer à la ligne suivante (cela rajoute des lignes au fichier si on est sur la dernière).

Les raccourcis **Meta-f** et **Meta-b** étendent les deux premières fonctionnalités aux mots : avance ou recul d'un mot. Le raccourci **Ctrl-a** (pas de mnémonique) permet d'aller au début de la ligne, **Ctrl-e** (“End of line”, fin de ligne) à la fin, **Meta-a** et **Meta-e** correspondent aux début et fin du paragraphe courant.

Le raccourci **Ctrl-v** (pas de mnémonique) permet de passer à la page suivant (“bip” si on est à la fin du fichier), **Meta-v** (pas de mnémonique) permet de passer à la page précédente (“bip” si on est au début du fichier). Enfin, le raccourci **Meta-<** permet de revenir au début du fichier, alors que **Meta->** (obtenu par **Meta- \uparrow -<**) positionne le curseur texte à la fin du fichier.

Pas de panique : il est aussi possible de déplacer le curseur texte d'un clic de la souris (le curseur texte se place sous le curseur souris). On peut aussi utiliser l'ascenseur (situé près du cadre gauche) pour monter et descendre dans le texte : un clic à gauche dans la zone de l'ascenseur fait descendre le texte (d'une hauteur proportionnelle à la distance du haut de l'ascenseur jusqu'au curseur souris), un clic à droite ayant un effet similaire pour la montée, alors qu'un clic au milieu permet un mouvement plus fin et continu)...

Application : Parcourez le fichier `debut.sci` d'un bout à l'autre en utilisant chacune des méthodes précédentes. **Si vous modifiez ce fichier**, empressez-vous de passer à la section suivante, pour corriger cette modification.

3.3 Modifications et recherches

Le mode d'édition par défaut d'Emacs est l'insertion : taper du texte a pour effet de l'insérer à la place du curseur texte. On peut évidemment effacer du texte, en utilisant \leftarrow (Delete) pour revenir en arrière (efface le caractère **précédant** le curseur texte), ou **Ctrl-d** (“Delete”, détruire) pour effacer en avant (le caractère **suisant** le curseur texte). Il est possible d'effacer les retours à la ligne (qui sont invisibles), ce qui fusionne les deux lignes qui entourent ce retour à la ligne.

Il est possible de sélectionner du texte avec la souris, et de le copier ailleurs : un clic à gauche déplace le curseur texte et marque le début, un clic à droite sélectionne la zone reliant le curseur texte au curseur souris et copie cette zone, un clic au milieu colle la zone précédemment sélectionnée où se trouve le curseur souris et positionne le curseur texte à la fin du texte ajouté. On peut aussi copier du texte uniquement avec le curseur texte : on choisit

le début de la zone en tapant **Ctrl-Espace** (le message **Mark set**, position repérée, s'affiche sur la dernière ligne), **Ctrl-w** coupe la zone reliant cette position à la nouvelle position du curseur texte, et **Ctrl-y** la colle à la nouvelle position du curseur texte.

Il est heureusement possible de **défaire des erreurs**, en utilisant au choix **Ctrl-x Ctrl-u** (“Undo”, défaire; peu recommandé, trop compliqué), **Ctrl-_** (très bien pour les claviers AZERTY) ou bien **Ctrl-/** (préférable sur les claviers QWERTY). On peut répéter plusieurs fois cette opération, revenant en arrière dans l'édition du fichier. Ce “retour en arrière” est très pratique. Attention, cependant : il s'arrête dès qu'on fait autre chose qu'annuler les actions précédentes, et annuler les actions précédentes revient alors à annuler les annulations que l'on vient de faire (jusqu'au point maximum de l'avancement, où on repart en arrière). Vous pourrez essayer, plutôt un autre jour et surtout sur la copie d'un fichier (ou du moins sur un fichier sans aucune importance : les risques de raté sont grands lorsqu'on tente ce genre de manipulation pour la première fois).

Pour faire des modifications plus intelligemment, il est possible de chercher un texte dans le fichier. Utilisez pour cela **Ctrl-s** (“Search”, chercher) si vous cherchez le texte dans la suite du fichier (la recherche reprendra du début après vous avoir signalé qu'elle a atteint la fin du fichier), ou **Ctrl-r** (r précède s dans l'alphabet) si vous préférez une recherche à rebours. Tout deux déclenchent un affichage dans la ligne sous la ligne noire : **I-search** : dans le cas de **Ctrl-s**, **I-search backward** : pour **Ctrl-r** ; puis attendent que vous tapiez le texte recherché et effectuent la recherche au fur et à mesure de votre frappe. Si l'instance trouvée du texte (la première ou la dernière) ne vous convient pas, tapez à nouveau **Ctrl-s** ou **Ctrl-r** (vous pouvez passer de l'un à l'autre sans problème). Tapez deux fois de suite **Ctrl-s** ou **Ctrl-r** (sans entrer de texte à la demande) relance la précédente recherche effectuée.

Il est aussi possible de chercher et de remplacer une expression, en tapant **Meta-%**, ce qui fait apparaître le message **Query replace** : (remplacement avec vérification) dans la dernière ligne. Tapez ensuite l'expression à remplacer, puis Return. Le message de la dernière ligne devient alors **Query replace** : <expression-à-remplacer> **with**:. Tapez l'expression remplaçante, puis Return. Pour chaque occurrence de l'expression à remplacer entre le curseur texte et la fin du fichier (pas de retour au début du fichier), Emacs vous demande de confirmer la substitution (par le message **Query replacing** <expression-à-remplacer> **with** <expression-remplaçante> : (? for help) : répondez **y** pour oui, **n** pour n ou **q** pour terminer.

Application : Modifiez à votre guise le fichier **debut.sci** en utilisant chacune des méthodes précédentes. Vous pourrez le récupérer sans problème dans son répertoire d'origine **~scheuer/LicMath/Algo/**.

3.4 Sortir d'Emacs

Si vous souhaitez en savoir plus sur les commandes Emacs, n'hésitez pas à utiliser l'aide en ligne : **Ctrl-h** propose plusieurs aides, dont **Ctrl-h i** (“Help Informations”) qui décrit de nombreux utilitaires et **Ctrl-h t** (“Help Tutorial”) qui propose un tutoriel.

Sinon, vous pouvez quitter Emacs en tapant **Ctrl-x Ctrl-c** (“Close”, fermer), ou en utilisant la dernière option du menu déroulant **File** (en haut à gauche)... **Ne le faites pas tout de suite**, nous avons encore un peu besoin de cet éditeur.

4 Utilisation du logiciel SciLab

Vous allez utiliser pour chaque TP, soit pour neuf séances, le logiciel SciLab. Plutôt que le lancer depuis le menu **Applications** (*cf.* § 1.3), son accès étant long (nous allons voir que la liste des menus à traverser est longue), il sera plus simple de disposer d'un bouton dans le tableau de bord.

Pour obtenir ce bouton, sélectionnez le menu contextuel du tableau de bord (d'un clic avec le bouton droit). Choisissez le sous-menu **Ajouter au tableau de bord**. Parcourez ensuite les sous-menus successifs **Lanceur depuis le menu**, **Menu Debian**, **Apps** et **Math**, pour finir par sélectionner l'élément **Scilab** de ce dernier menu. Ce bouton étant créé, vous pouvez grâce à son menu contextuel le déplacer sur le tableau de bord ou le personnaliser.

Vous pouvez alors utiliser le bouton nouvellement créé pour lancer SciLab. Cependant, l'interface de SciLab manquant de quelques fonctionnalités, nous allons préférer utiliser SciLab sous Emacs.

5 Lancer SciLab sous Emacs

Pour cela, il faut d'abord modifier deux fichiers configurant le fonctionnement de vos applications. Tout d'abord, utilisez Emacs pour copier les lignes 49 et 50 (modifiant les chemins de recherche des macros Lisp), 82 à 84 (pour prendre en compte la molette de la souris) et 171 à 181 (configurant SciLab sous Emacs) du fichier `~scheuer/.emacs`, définissant la configuration Emacs de l'utilisateur `scheuer`, aux mêmes positions de votre fichier de configuration Emacs (`~/.emacs`). Pour vérifier que cette modification ne pose pas problème, lancez un autre Emacs **sans quitter celui-ci** : une erreur dans le fichier de configuration pouvant empêcher de lancer Emacs, il vaut mieux garder celui qui est ouvert pour pouvoir corriger cette erreur.

Si tout fonctionne bien, fermez le second Emacs. Ajouter maintenant, en fin de votre fichier `~/.cshrc` (configuration votre fenêtre de commande) les lignes :

```
### Pour SciLab sous Emacs
setenv SCI "/usr/lib/scilab"
```

Pour que cette modification soit prise en compte, il faut ensuite taper `source ~/.cshrc` dans une fenêtre de commandes. La modification n'est cependant prise en compte **que** dans cette fenêtre de commandes, et dans les applications en dépendant. Cela étant nécessaire au lancement de SciLab sous Emacs, Emacs doit donc être lancé depuis cette fenêtre de commandes (*cf.* § 3).

Chargez ensuite un des fichiers `debut.sci`. Vous constaterez qu'Emacs fait de la reconnaissance syntaxique : il analyse le fichier selon son type (indiqué par son extension), reconnaît les mots-clés et les met en couleur, et finalement offre des menus et de commandes spécifiques à ce type. Ainsi, Vous constaterez l'apparition de trois menus : **Scilab**, **Navigate** et **Insert**. Notez aussi la possibilité d'utiliser la molette de la souris pour parcourir verticalement le fichier.

Démarrer ensuite Scilab en utilisant l'option **Start Scilab** du menu **Scilab**. Cette commande ouvrant un autre buffer, il est préférable de visualiser les deux. Pour cela, choisissez l'option **Split Window** du menu **File**, ou tapez `Ctrl-x 2`. La fenêtre se coupe alors en deux. Dans la première partie, revenez au fichier `debut.sci`, en utilisant l'option correspondante du menu **Buffers** ou en tapant `Ctrl-x b`.

Vous pouvez alors évaluer le résultat de la totalité du fichier en utilisant l'option **Save and exec** du menu **Scilab**, ou en tapant **Ctrl-c Ctrl-l**. Vous pouvez aussi sélectionner une région, dont le début est marqué d'un clic (du bouton gauche) de la souris et la fin d'un clic droit, la mettant ainsi en couleur, avant d'évaluer uniquement le résultat de cette région en utilisant l'option **Run Region** du menu **Scilab**, ou en tapant **Ctrl-c r**.

Manipulez un peu ce fichier, pour vous familiariser avec l'utilisation conjointe de SciLab et d'Emacs. **Évitez** cependant d'utiliser les commandes graphiques de SciLab (**plot2d**, etc.), celles-ci semblant ne pas se terminer correctement : SciLab, et donc Emacs, ne peuvent pas terminer normalement. Pour ces commandes, préférez le logiciel SciLab, hors d'Emacs. De même, dans le buffer SciLab sous Emacs, n'utilisez pas le menu **Graphic 0**.

Enfin, l'aide n'apparaît pas très lisiblement sous Emacs, quoi que celui-ci mette en couleur certains termes : sa présentation est coupée (page par page). Si le déroulement est plus lisible sous SciLab, le manque de couleur nuit à la lisibilité. De plus, à l'heure de l'hypertexte, cette aide en ligne paraît vraiment simpliste.

Avant de quitter Emacs, pensez à terminer la session SciLab en tapant **exit**, ou Emacs signalera qu'il doit terminer un processus tournant encore (pas d'inquiétude à avoir).

6 Quitter Gnome

Lorsque vous avez terminé, vous pouvez vous déconnecter de la machine en sélectionnant l'option **Terminer la session** du menu **Actions** du tableau de bord.