

Installation et utilisation de SSH

.et

.soumission de travail

SSH est un protocole de communication qui permet de communiquer à distance entre deux ordinateurs. Pour cela, il faut qu'un des deux ordinateurs fasse tourner un serveur et que l'autre possède un client. L'ordinateur que vous utilisez est le client. L'ordinateur du Département d'informatique est le serveur. Pour pouvoir donc vous connecter au serveur, il faut que vous installiez sur votre ordinateur un client SSH.

Installation d'un client SSH

Vous pouvez obtenir un client SSH:

- 1) **pour windows:** vous pouvez obtenir un client dans la section prive du site du cours (IFT159). Les instructions qui suivent indiquent comment l'utiliser.
- 2) **pour Linux:** vous pouvez en obtenir un (openSSH) avec votre distribution. La section outils de la page web du cours ift159 donne des liens vers d'autres outils SSH.

Pour l'utiliser, il suffit d'entrer au terminal la commande

ssh utilisateur@rigel.dmi.usherb.ca.

- 3) **pour MacOS:** un client SSH est intégré dans le terminal (ouvrir le terminal et utiliser la commande ou

ssh utilisateur@rigel.dmi.usherb.ca

le principe est le même après que ce qui est indiqué ici. Il existe aussi un client SFTP libre :

<http://www.columbia.edu/acis/software/fugu/>

Les commandes en ligne de SSH et le principe de fugu sont similaires à ce qui se trouve dans ce document.

Utilisation de SSH

Notes :

- Le document qui suit introduit l'utilisation de SSH en se servant de l'outil recommandé pour Windows. Les commandes pour les autres environnements sont toutefois similaires.
- Les exemples dans ce document supposent que l'étudiant suit ift159. Il suffit de remplacer ift159 par le cours suivi si ce n'est pas votre cas.
- Je suppose aussi que vous avez déjà activé votre compte au département d'informatique :
http://www.usherbrooke.ca/informatique/ressources/documentation/compte_dmi.htm.
- Les commandes en ligne de SSH et le principe de fugu est après similaire à ce qui se trouve dans ce document.

Supposons maintenant que vous avez écrit votre programme et que vous l'avez testé. Celui-ci marche chez vous ou sur votre poste. Vous devez le déposer sur le serveur du département (rigel) pour pouvoir ensuite le soumettre. Nous allons voir dans un premier temps comment configurer SSH pour que vous puissiez effectuer la soumission par la suite.

1. Étape 1 : Démarrer SSH

Le client SSH vient en fait avec deux programmes : le premier permet de vous connecter sur le serveur comme si vous étiez au laboratoire D7-1021-1 et l'autre permet de transférer des fichiers comme si vous étiez sur un répertoire local. On peut passer aisément de l'un à l'autre.

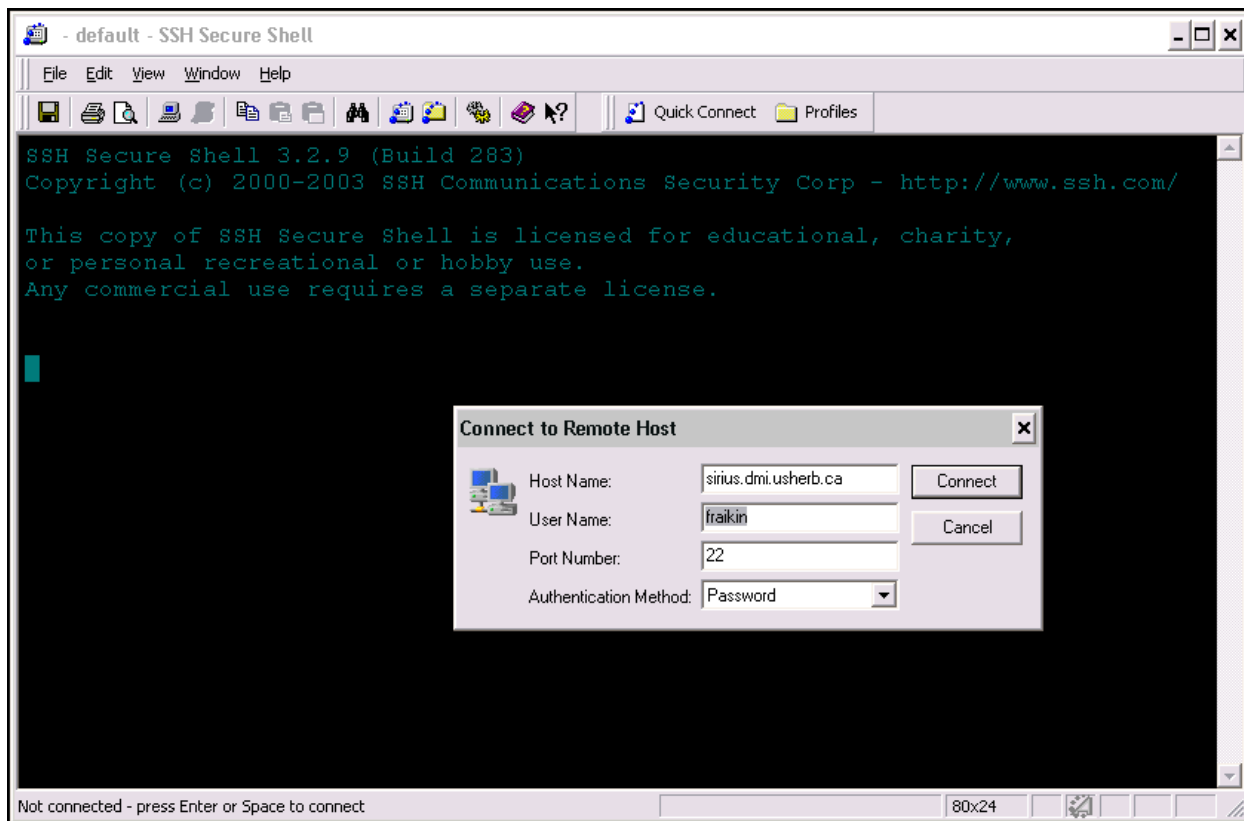


2. Étape 2 : Établir la connexion

Ouvrez le Client SSH avec le premier des deux icônes. Puis sélectionnez l'icône *Quick Connect* pour créer rapidement une première connexion.

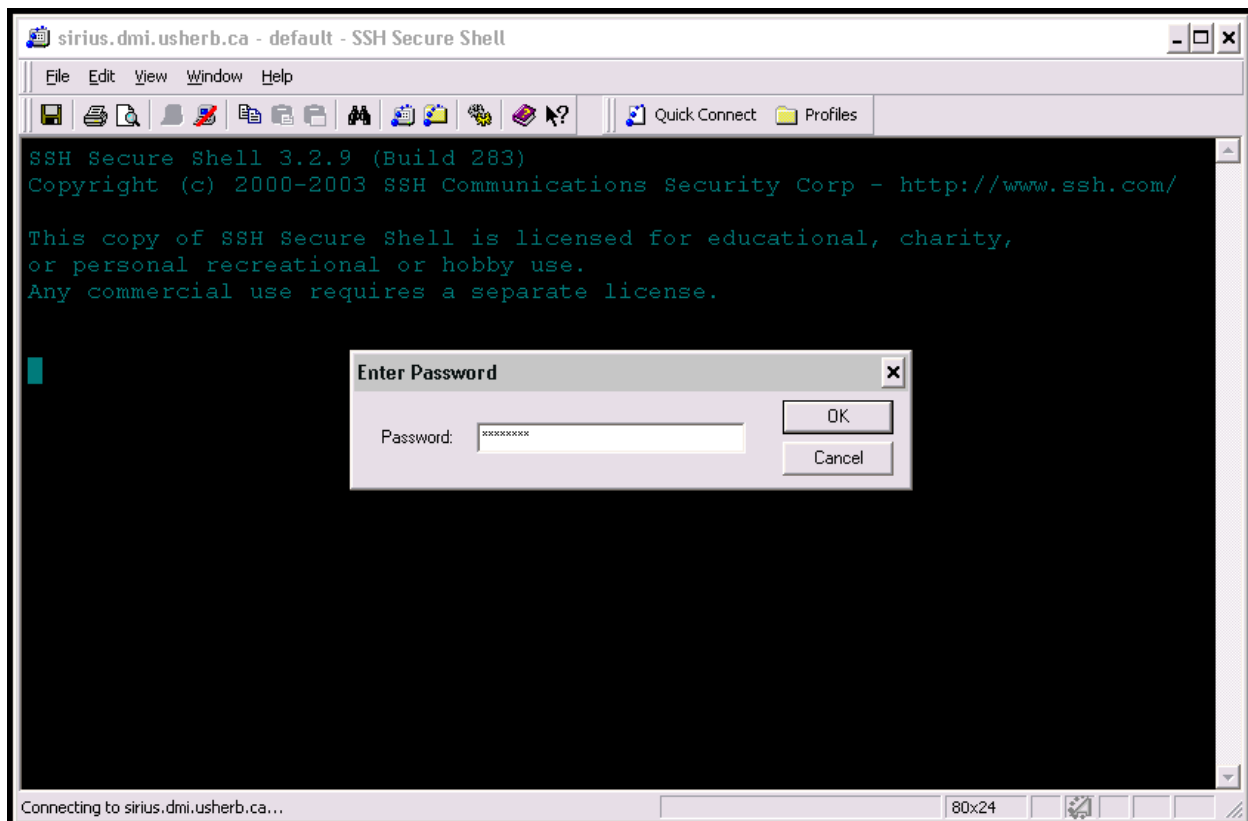


Le nom de l'hôte (*host*) est ***rigel***.dmi.usherb.ca. Le nom d'utilisateur (*User name*) est le code qui vous est attribué pour vous connecter au département. Ce n'est pas votre CIP. Il est normalement composé des 4 premières lettres de votre nom de famille, de la première lettre de votre prénom et de deux chiffres. Appuyez ensuite sur *Connect*.



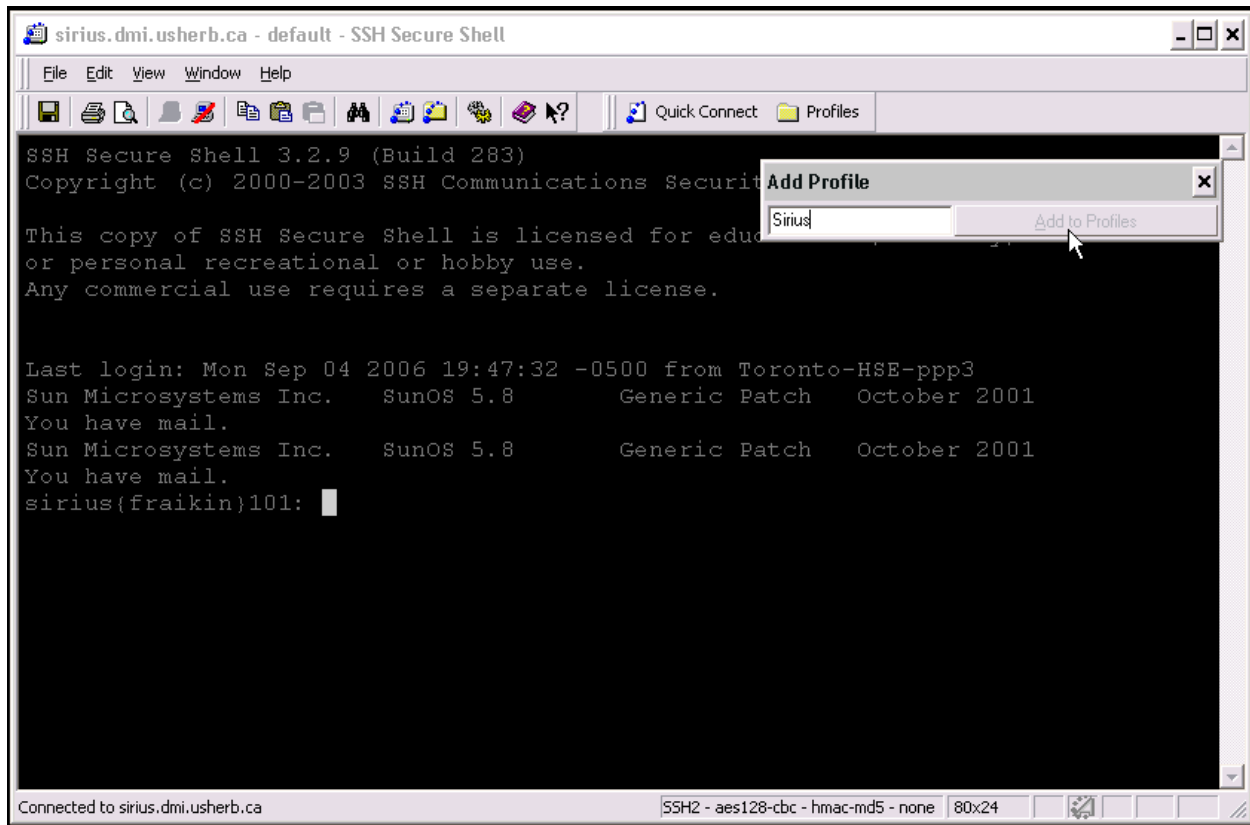
3. Étape 3 : Saisie du mot de passe

Le programme vous demande votre mot de passe. Encore une fois, c'est celui utilisé pour vous connecter au D7-1021-1.

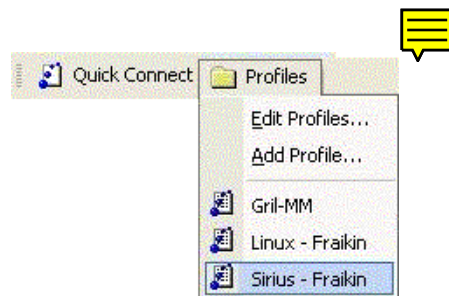


4. Étape 4 : Mémorisation de la connexion

Une fenêtre s'ouvre pour créer un nouveau profil avec les informations rentrées afin de ne pas avoir à tout retaper la prochaine fois. Donnez un nom au profil (j'ai choisi simplement rigel) et appuyez sur *Add to Profiles*.

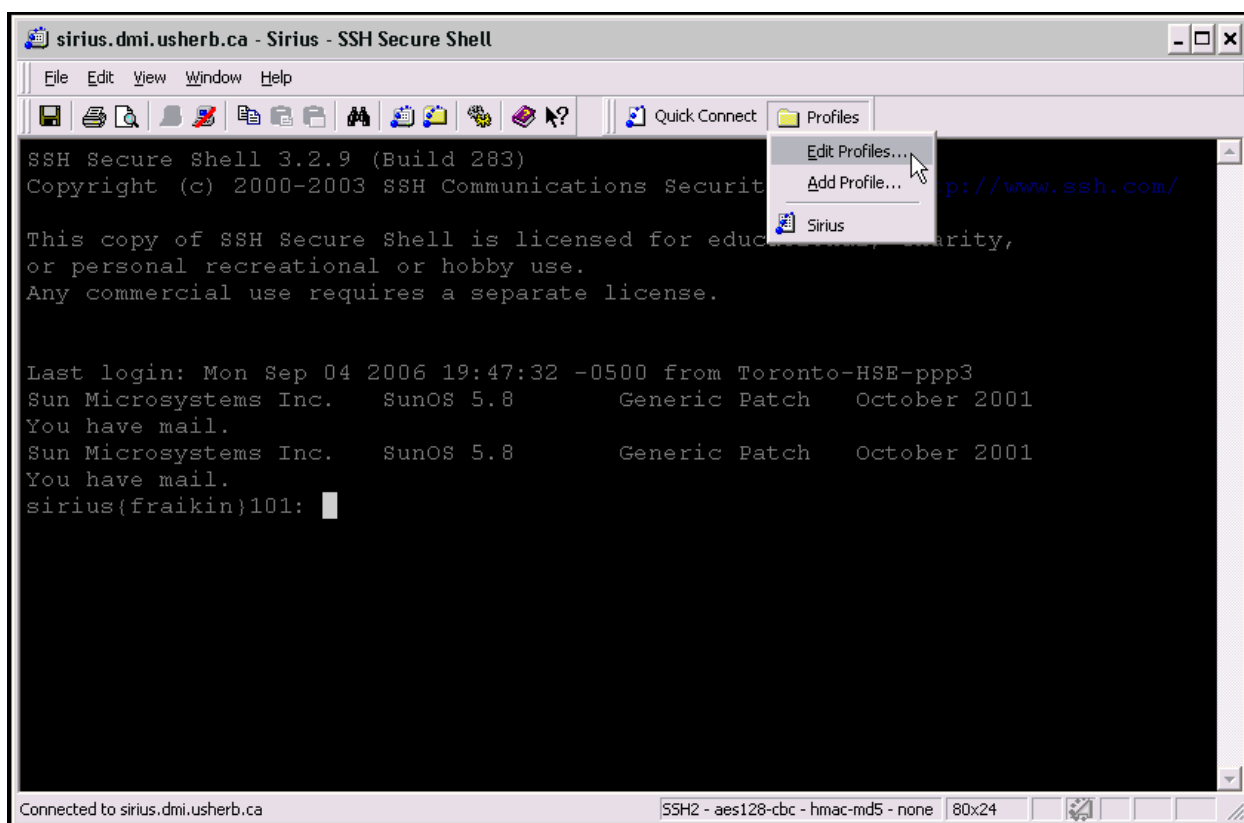


Par la suite, vous pourrez aisément le récupérer



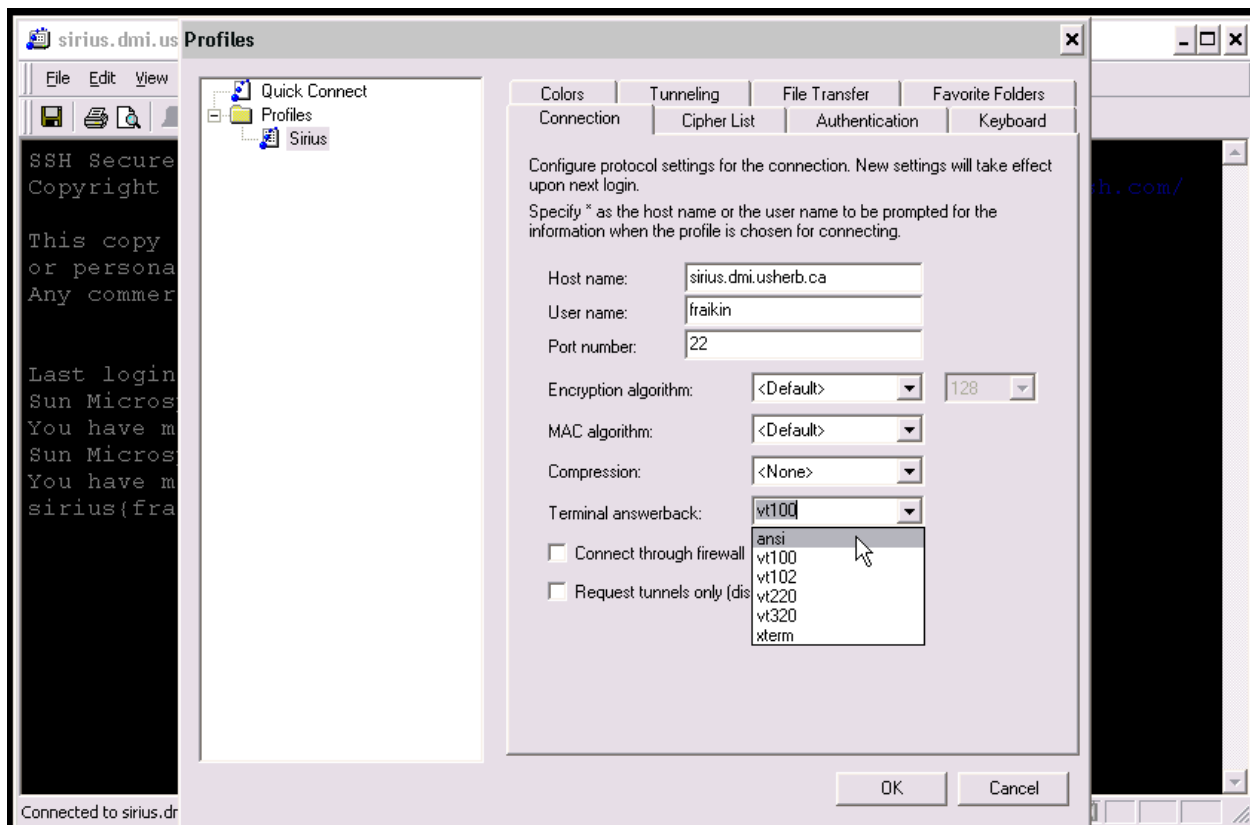
5. Étape 5

Dans le menu *Profiles*, vous voyez maintenant apparaître le profil que vous venez de créer. La prochaine fois, il suffira de le sélectionner pour arriver à l'écran de demande du mot de passe. Pour l'instant, sélectionner *Edit Profiles*.



6. Étape 6

Nous obtenons l'écran qui contient les informations des profils. Nous allons changer un paramètre qui assouplit l'utilisation du terminal. Dans le menu *Terminal answerback*, sélectionner *ansi*. Ceci permettra à votre terminal de mieux répondre et évitera certain affichage ennuyeux.



7. Étape 7 : Après la première connexion

Si cela n'est pas déjà fait, Vous allez devoir commencer à créer un répertoire pour les devoirs de ift159, puis vous déplacer dans ce répertoire. De là, vous pouvez alors créer des répertoires pour les six (6) devoirs à remettre. Pour cela, à l'invite qui apparaît dans votre fenêtre, vous allez taper les commandes suivantes :

mkdir ift159

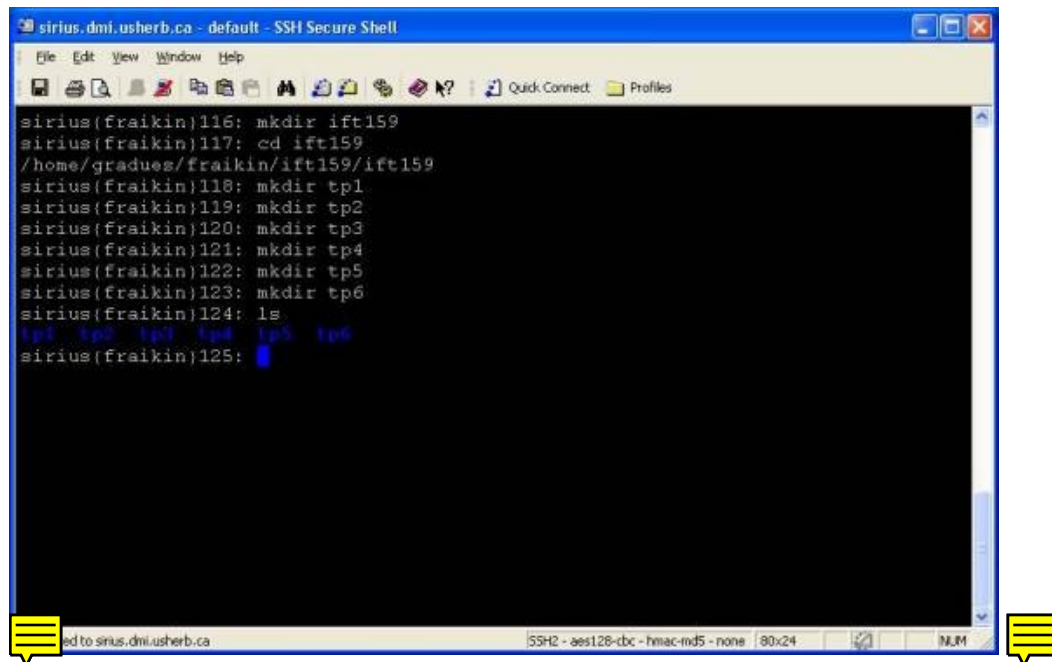
créer le répertoire ift159

cd ift159

se déplacer dans le répertoire ift159

mkdir tp1	créer le répertoire tp1
mkdir tp2	créer le répertoire tp2
mkdir tp3	créer le répertoire tp3
mkdir tp4	créer le répertoire tp4
mkdir tp5	créer le répertoire tp5
mkdir tp6	créer le répertoire tp6

La commande **ls** (*list*) vous permet de voir l'affichage de ce que vous avez fait.



```

sirius.dmi.usherb.ca - default - SSH Secure Shell
File Edit View Window Help
sirius(fraikin)116: mkdir ift159
sirius(fraikin)117: cd ift159
/home/gradues/fraikin/ift159/ift159
sirius(fraikin)118: mkdir tp1
sirius(fraikin)119: mkdir tp2
sirius(fraikin)120: mkdir tp3
sirius(fraikin)121: mkdir tp4
sirius(fraikin)122: mkdir tp5
sirius(fraikin)123: mkdir tp6
sirius(fraikin)124: ls
tp1 tp2 tp3 tp4 tp5 tp6
sirius(fraikin)125:

```

8. Étape 8 : Connexion ultérieure

Maintenant, quand vous vous reconnecterez, vous pourrez taper la commande

cd ift159/tp2

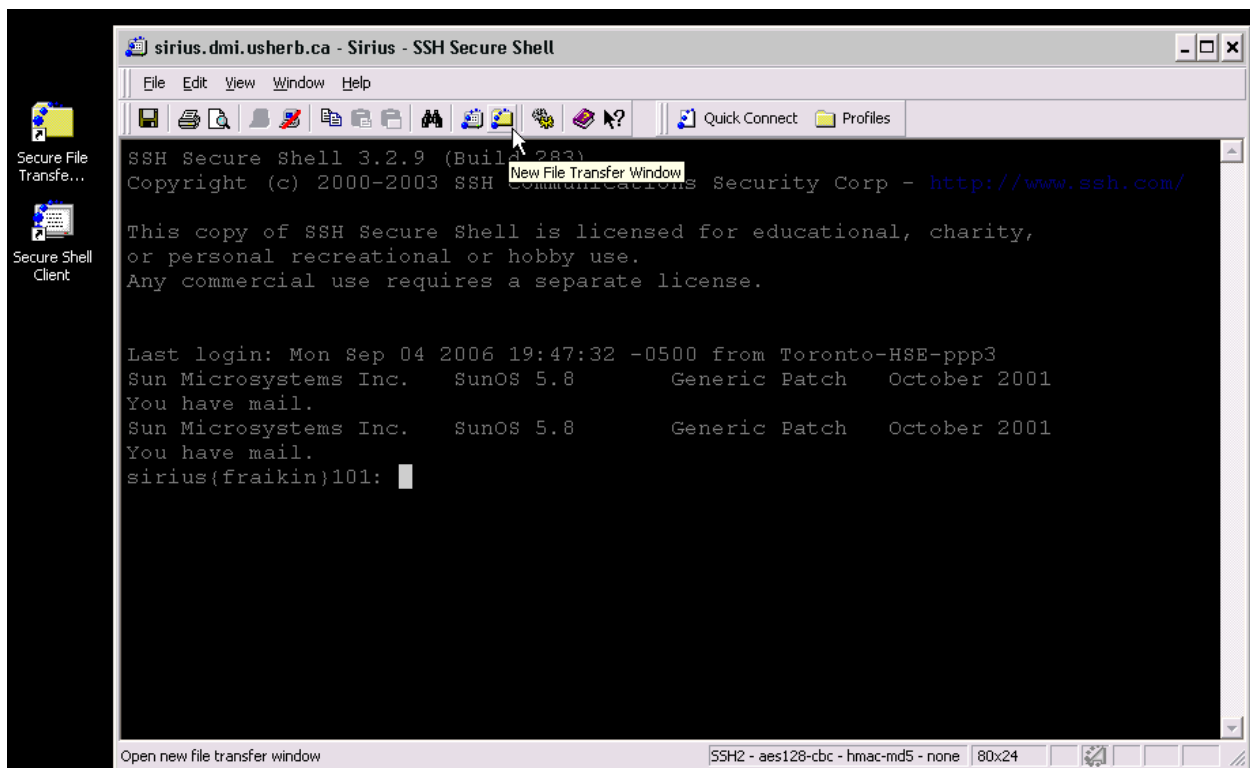
pour accéder directement dans le répertoire du tp2.

Attention sous UNIX, les majuscules et les minuscules sont différenciées. Il faut aussi utiliser un *slash* (/) et non un *backslash* comme sous windows (\).

9. Étape 7

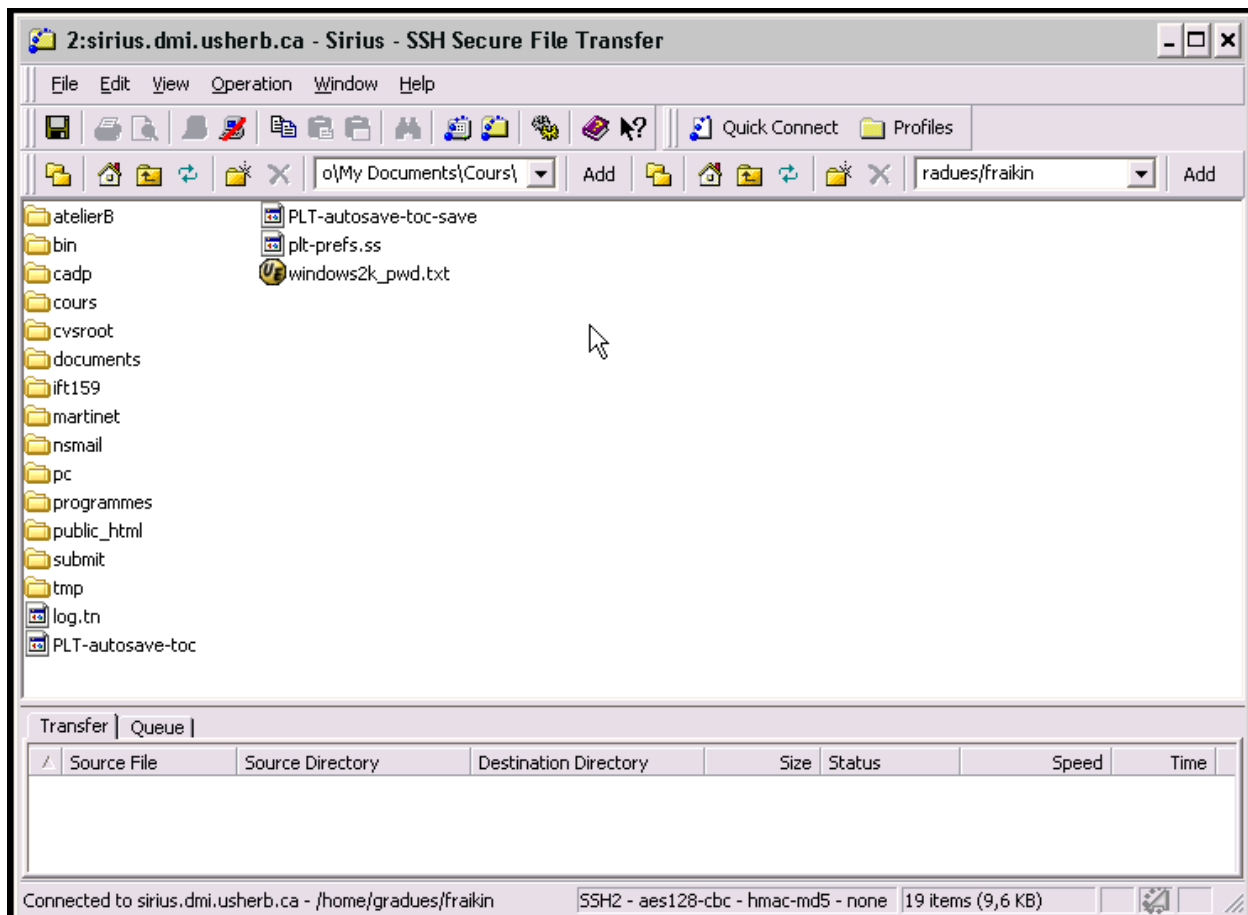
A partir de cette étape, votre terminal est configuré.

Nous allons basculer du terminal au gestionnaire de transfert de fichiers (qui utilise SFTP, *Secure File Transfert Protocol*). La procédure décrite ici est illustrée avec le fichier tp1.cpp. Il est évident que la même chose est faisable pour un autre fichier comme tp1.tex. Il suffit pour cela d'appuyer sur l'icône du gestionnaire présent dans votre fenêtre.



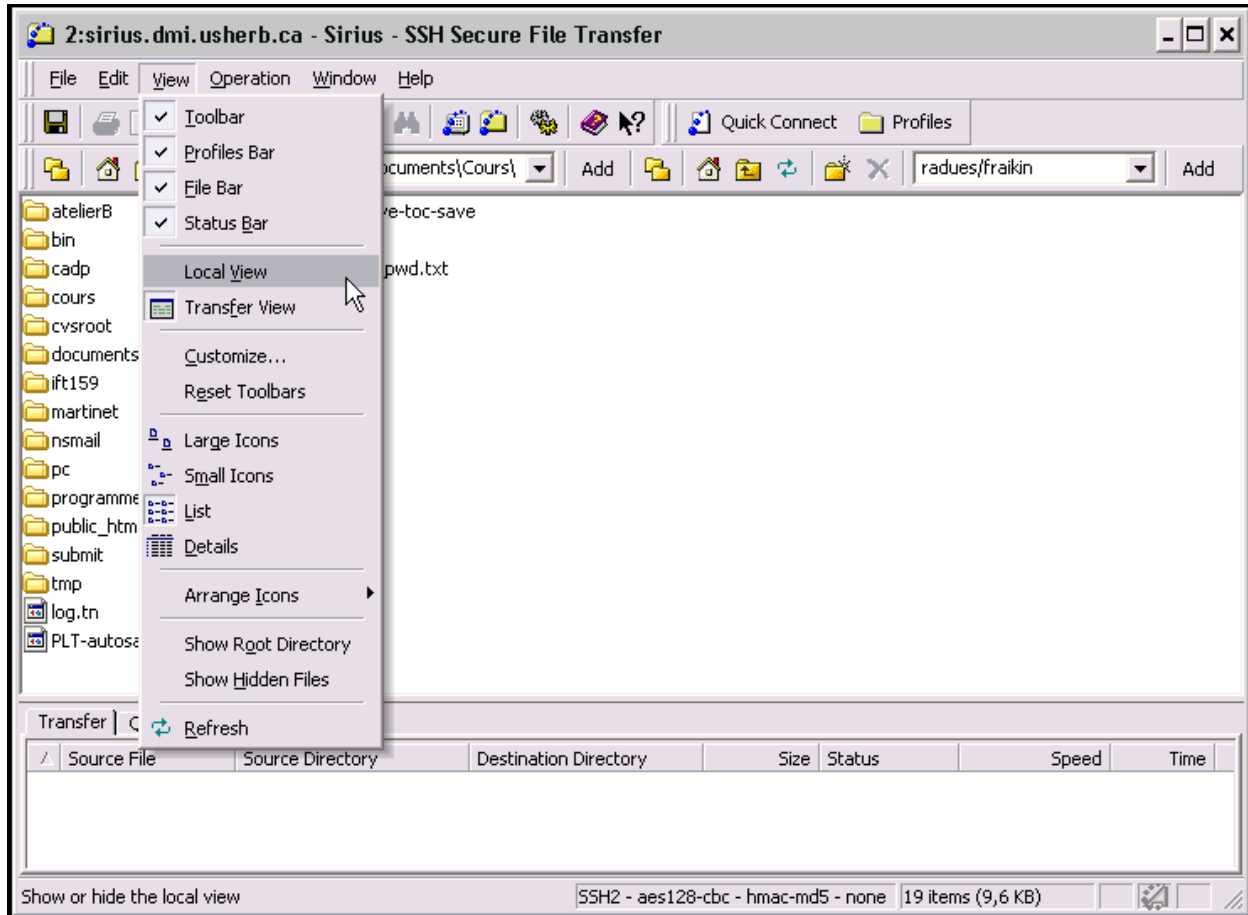
10.Étape 8

La fenêtre du gestionnaire s'ouvre. La première fois vous ne devriez voir apparaître uniquement que les fichiers du serveur Rigel (c'est-à-dire que vous ne voyez pas les fichiers sur votre ordinateur). Ceci suffit à travailler puisque cette fenêtre se comporte presque comme une fenêtre de répertoire de l'*explorer* de Windows. Avec un clic droit vous pouvez faire apparaître des menus pour créer des répertoires, pour effacer des fichiers ou des répertoires ou encore pour faire des copies par exemple. On peut faire des déplacements de fichiers à l'aide de la souris (*drag and drop*) sur les fichiers de cette fenêtre vers votre système, ou de votre système vers cette fenêtre.



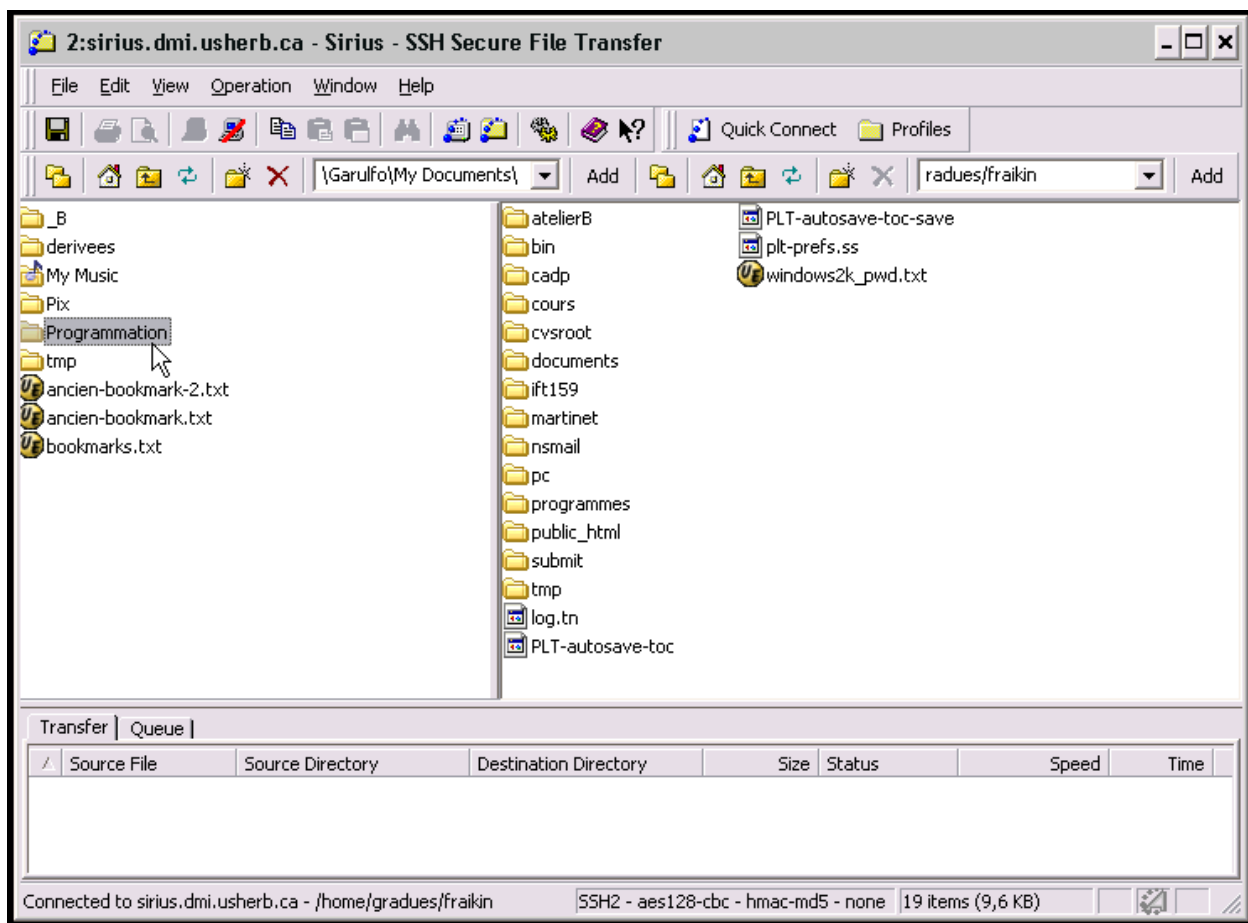
11.Étape 9

Nous allons cependant procéder pour que vous ayez la vue des fichiers locaux sur votre gauche et des fichiers distants sur votre droite. Pour cela, ouvrez le menu *View* et sélectionnez *Local View*.



12.Étape 10

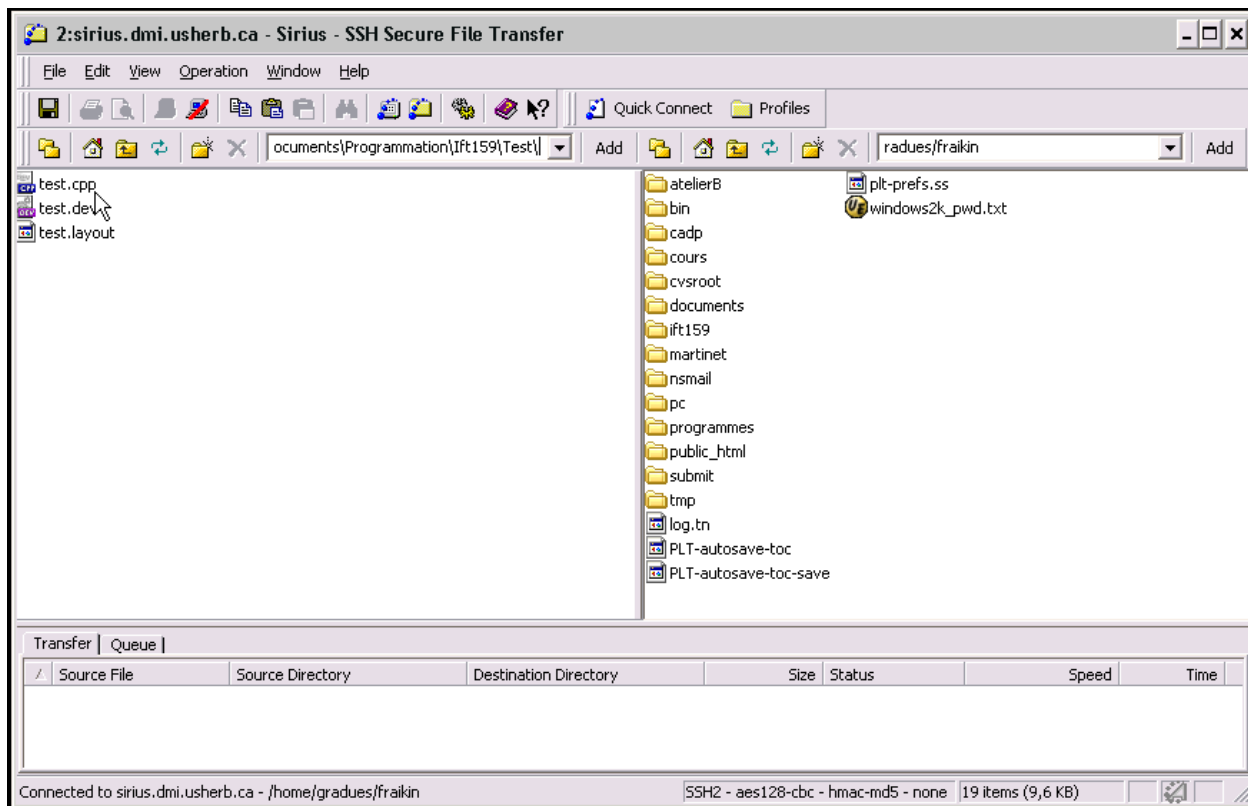
Vous disposez maintenant des deux vues si vous le préférez. Par exemple, dans l'image qui suit, le répertoire Programmation est celui qui se trouve sur mon ordinateur local. La partie de droite affiche toujours les fichiers sur mon compte sous Rigel.



Vous pouvez naviguer facilement et transférer des documents d'une fenêtre à l'autre ou même directement d'une autre fenêtre (de l'explorateur de Windows par exemple) dans un répertoire du service distant.

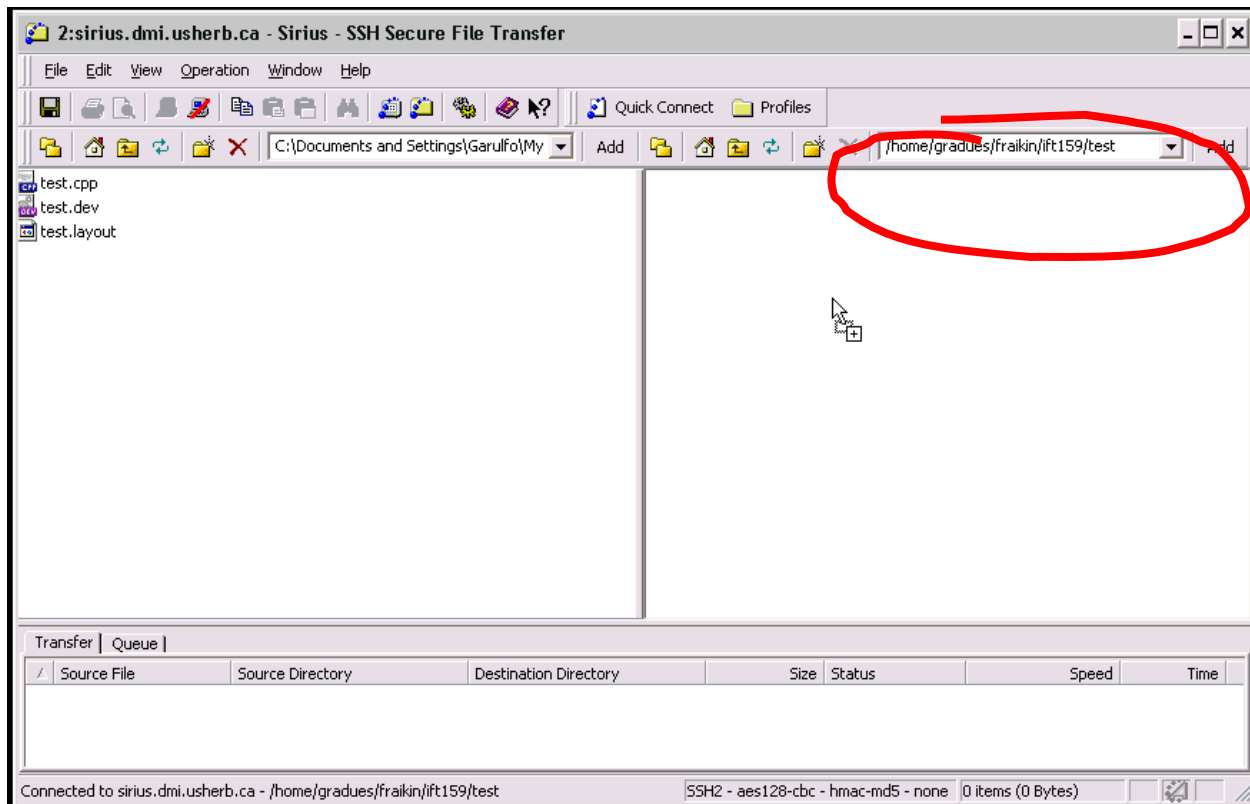
13.Étape 11

Il ne vous reste maintenant qu'à aller chercher le fichier pertinent (dans cet exemple c'est le fichier test.cpp) et le placer ensuite sur le serveur Rigel.



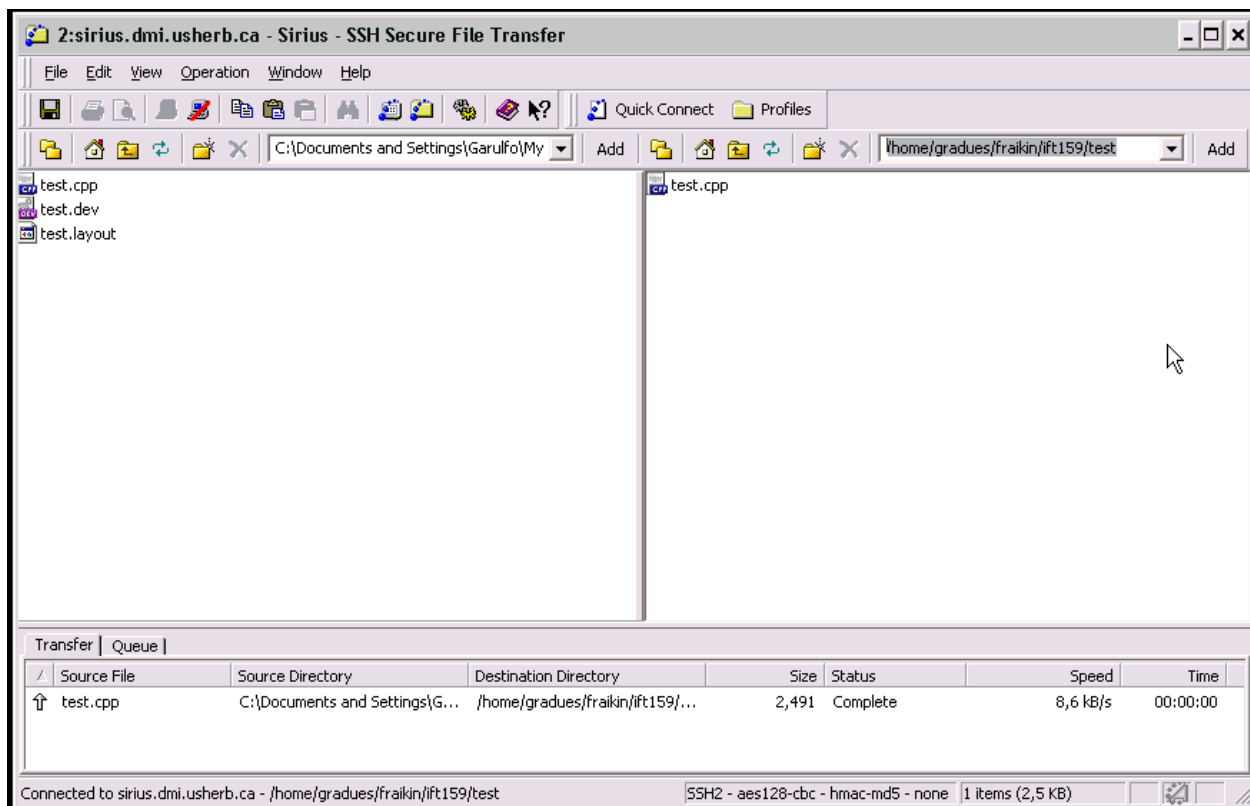
14.Étape 12

Vous pouvez remarquer ici que j'ai, à partir de mon compte qui est /home/gradues/fraikin et qu'on voyait dans la partie droite de la fenêtre précédente, créé un répertoire test dans le répertoire ift159. Vous n'êtes pas obligé de respecter cela. Mais rappelez-vous cependant bien le chemin (c'est-à-dire la position dans la partie entouré de rouge) où vous placez votre fichier.



15.Étape 13

Un fois le fichier déposé avec la souris à droite, votre fichier est sur le serveur Sirius et peut être soumis par la procédure à venir. Encore une fois, veuillez à vous rappeler le chemin où il a été déposé. Ici le chemin entier est /home/gradues/fraikin/ift159/test/test.cpp. Ceci sera nécessaire pour la suite.



Soumission de travaux

Supposons donc que vous voulez envoyer votre premier TP pour le cours ift159. On peut vous demander, a priori, deux choses différentes :

1. envoyer un répertoire contenant le programme
2. envoyer les fichiers (éventuellement un seul) représentant le programme

Nous allons voir les deux cas en exemple. Il faut procéder en plusieurs étape.

Soumission grâce à la commande *turnin*.

1. Étape 1 : Transfert et positionnement

Vous devez transférer le document grâce à SFTP. Une fois le document placé dans le bon répertoire à l'aide de SFTP (disons dans le répertoire **~/ift159/tp1**), à l'aide du client SSH vous allez vous y rendre avec la commande **cd ~/ift159/tp1**.

2. Étape 2 : Compilation et tests

Une fois dans le répertoire où est situé votre programme, vous pouvez lancer la compilation à l'aide de la commande

g++ -o nomDeLExecutable nomDuFichierSource.cc

Si vous n'ajoutez pas l'option **-o nomDeLExecutable**, l'exécutable produit sera disponible sous le nom **a.out**, au lieu de **nomDeLExecutable**.

Pour lancer un exécutable il suffit de taper son nom dans la fenêtre du client SSH, en étant dans le répertoire dans lequel il se trouve :

nomDeLExecutable

Si votre programme ne s'exécute pas :

- il peut exister une protection qui empêche l'exécution d'un programme situé dans le répertoire courant (pour des raisons de sécurité);
- il se peut aussi que l'exécutable ne possède pas les droits d'exécution (a priori ce serait étonnant sous **sirius** si vous n'avez rien touché du

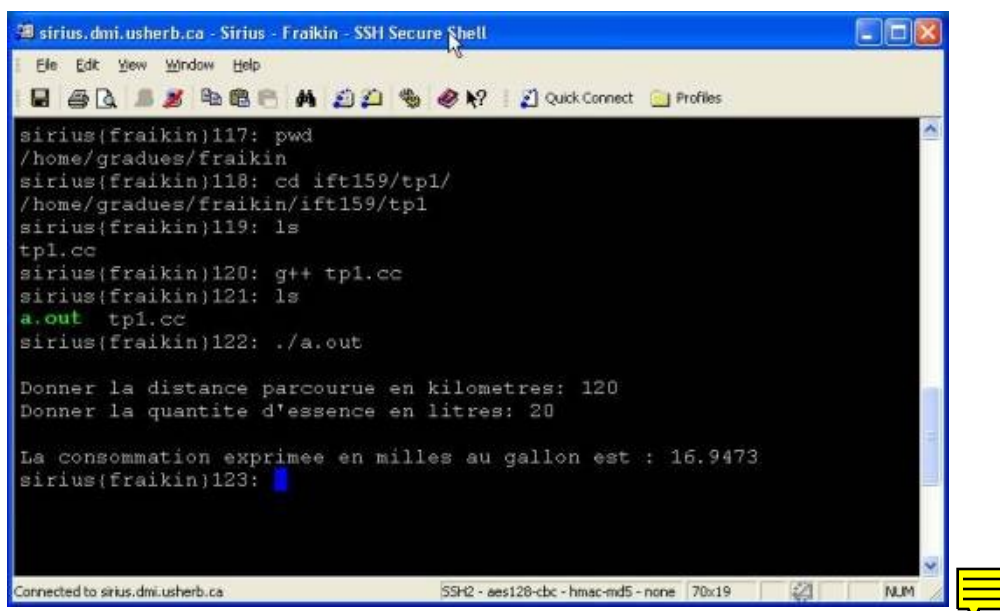
masque des droits par défaut);

- enfin, il se peut que vous n'ayez pas respecté les minuscules et les majuscules dans le nom.

En supposant que vous avez vérifié la casse du nom du programme, voici les solutions aux deux autres problèmes :

- Essayez de lancer votre programme en rajoutant le répertoire courant à l'avant du nom : **./nomDeLExecutable**
- Ajouter les droits d'exécution à votre programme en tapant la commande suivante : **chmod u+x nomDeLExecutable**

L'image suivante montre une compilation et une exécution



```
sirius.dmi.usherb.ca - Sirius - Fraikin - SSH Secure Shell
File Edit View Window Help
sirius(fraikin)117: pwd
/home/gradues/fraikin
sirius(fraikin)118: cd ift159/tp1/
/home/gradues/fraikin/ift159/tp1
sirius(fraikin)119: ls
tp1.cc
sirius(fraikin)120: g++ tp1.cc
sirius(fraikin)121: ls
a.out tp1.cc
sirius(fraikin)122: ./a.out

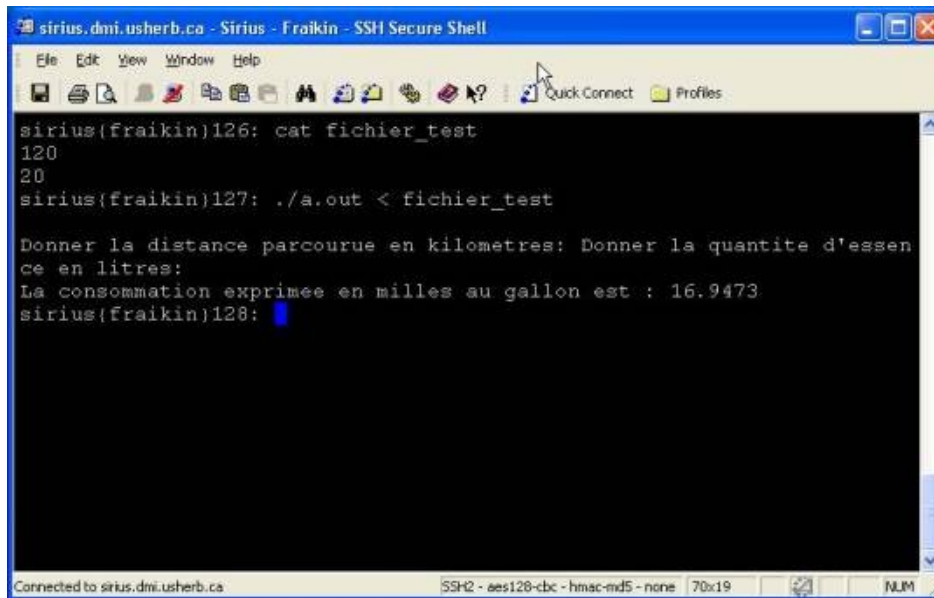
Donner la distance parcourue en kilometres: 120
Donner la quantite d'essence en litres: 20

La consommation exprimee en milles au gallon est : 16.9473
sirius(fraikin)123: 
```

Il peut être rapidement fatigant de devoir rentrer plusieurs fois les mêmes données. Il existe cependant une façon simple de faire rentrer rapidement une série de données lors de l'exécution. Il faut construire un fichier contenant les entrées qui seront placées à la suite, séparées d'une espace ou d'un retour chariot. Supposons que ce fichier s'appelle **fichier_test**. Il suffit ensuite d'appeler l'exécution du programme en redirigeant le fichier vers l'entrée du programme :

./nomDeLExecutable < fichier_test

L'image suivante reprend l'exemple de l'image précédente avec un fichier de test.



```
sirius(dmi.usherb.ca) - Sirius - Fraikin - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles

sirius(fraikin)126: cat fichier_test
120
20
sirius(fraikin)127: ./a.out < fichier_test

Donner la distance parcourue en kilometres: Donner la quantite d'essence en litres:
La consommation exprimee en milles au gallon est : 16.9473
sirius(fraikin)128:
Connected to sirius.dmi.usherb.ca
SSH2 - aes128-cbc - hmac-md5 - none 70x19 NLM
```

La commande **cat** permet d'afficher le contenu d'un fichier donné en argument.

3. Étape 3 : Conformité

Le département a récemment mis en place une série de normes à respecter. Celles-ci devraient vous être précisées à l'intérieur du cours par le professeur. Il existe aussi un document accessible par le *web* qui les résume :

<http://www.dmi.usherb.ca/~girard/ift159/>

Il peut cependant y avoir aussi des contraintes particulières à votre cours (nom du fichier, nom du répertoire, etc.).

Il faut s'assurer que le fichier est conforme à ce qui a été demandé, notamment le nom du fichier et le nom du répertoire.

4. Étape 4 : Soumission

Pour soumettre une liste de fichiers, par exemple **fich1.cc** et **fich2.cc**, il faut effectuer le **turnin** (projet **devoir1**, cours **ift159**) en tapant :

turnin -cift159 -pdevoir1 fich1.cc fich2.cc

Pour soumettre un répertoire, il faut remonter dans l'arborescence :

cd ..

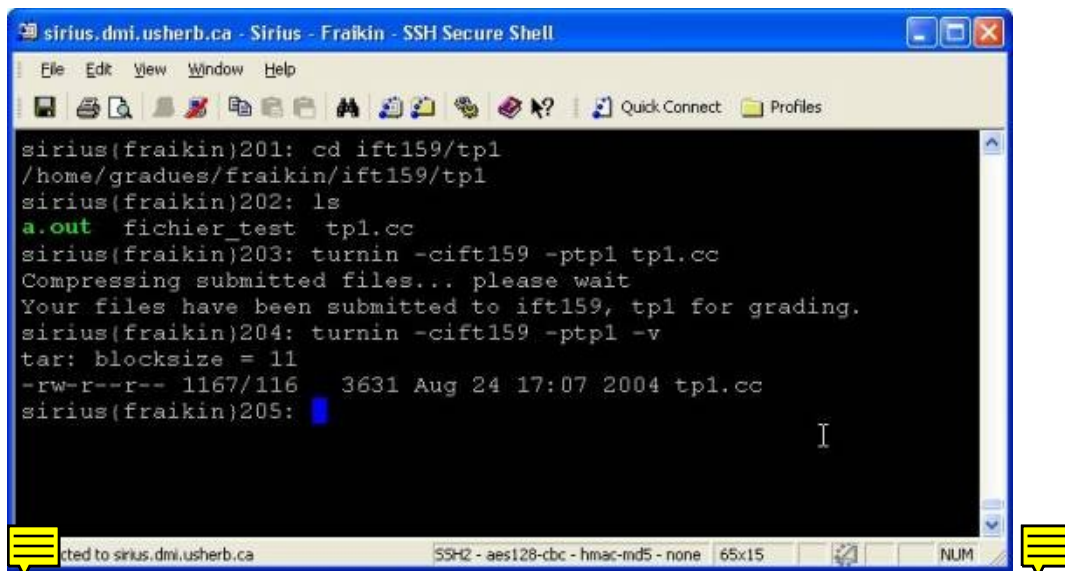
puis effectuer le **turnin** (projet **devoir1**, cours **ift159**), en tapant :

turnin -cift159 -pdevoir1 tp1

Normalement le professeur vous dira laquelle de ces deux options il choisit, et vous indiquera même éventuellement la commande exacte.

L'image qui suit est un exemple de soumission d'un fichier nommé **tp1.cc** pour le cours **ift159** et pour le projet **tp1**. Il est à noter que le répertoire dans lequel se trouve le fichier **tp1.cc** est **ift159/tp1** à partir du répertoire de l'utilisateur (celui dans lequel on arrive à l'ouverture de SSH). Le chemin complet serait donc, dans mon cas, serait **/home/gradues/fraikin/ift159/tp1**.

Vous pouvez noter que la dernière commande **turnin -cift159 -ptp1 -v** permet la vérification de la réussite de la soumission.



```
sirius.dmi.usherb.ca - Sirius - Fraikin - SSH Secure Shell
File Edit View Window Help
[Icons] Quick Connect Profiles

sirius(fraikin)201: cd ift159/tp1
/home/gradues/fraikin/ift159/tp1
sirius(fraikin)202: ls
a.out fichier_test tp1.cc
sirius(fraikin)203: turnin -cift159 -ptp1 tp1.cc
Compressing submitted files... please wait
Your files have been submitted to ift159, tp1 for grading.
sirius(fraikin)204: turnin -cift159 -ptp1 -v
tar: blocksize = 11
-rw-r--r-- 1167/116 3631 Aug 24 17:07 2004 tp1.cc
sirius(fraikin)205: 
```

5. Étape 5 : Erreur ou modification

En cas d'erreur dans votre soumission ou si vous voulez la modifier, vous pouvez toujours effectuer à nouveau toutes ces manipulations. L'ancienne soumission est alors écrasée. Faites donc attention à fournir **tous** les fichiers nécessaires et non seulement celui que vous avez modifié.

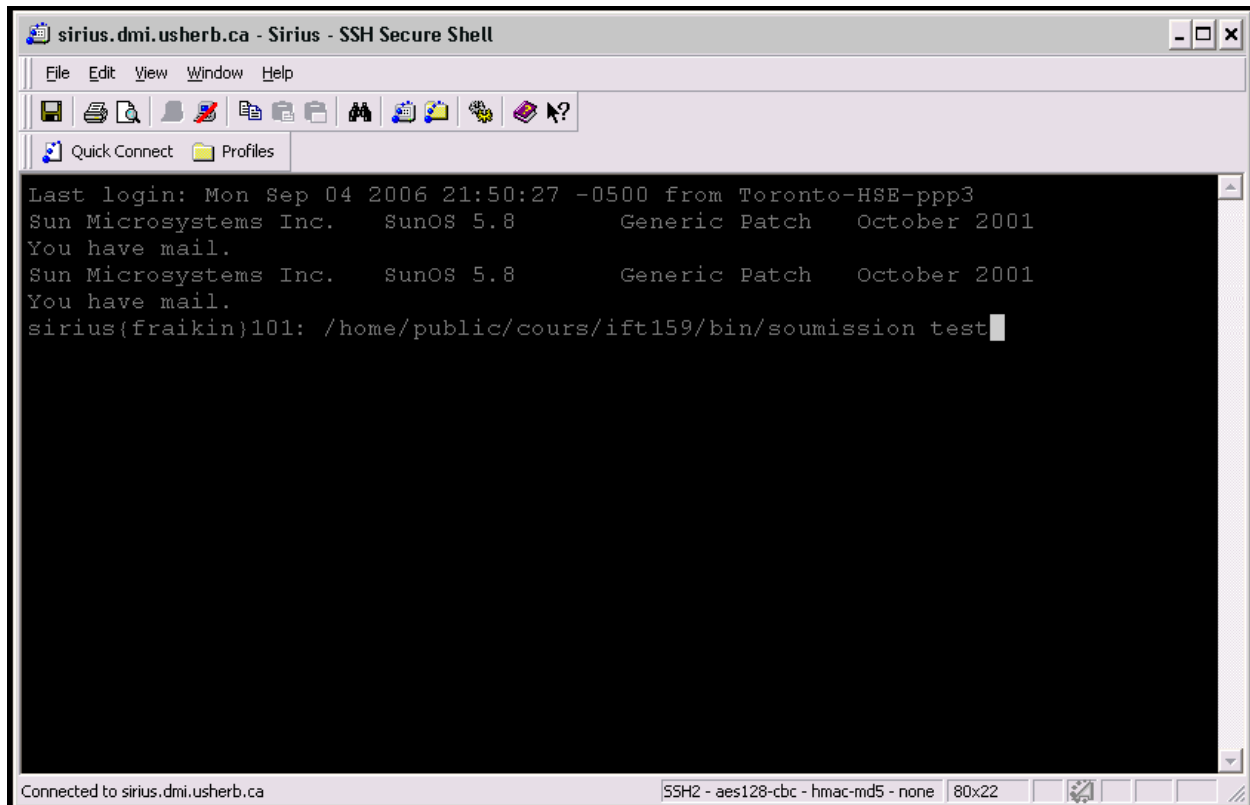
Soumission grâce à la commande *soumission*.

Vous devez suivre les mêmes étapes que pour la soumission avec turnin. Toutefois vous

utilisez une commande différente.

Revenez sur la fenêtre du terminal et taper le nom du programme de soumission suivi du nom du projet (qui est aussi le nom du fichier sans le .cpp) :

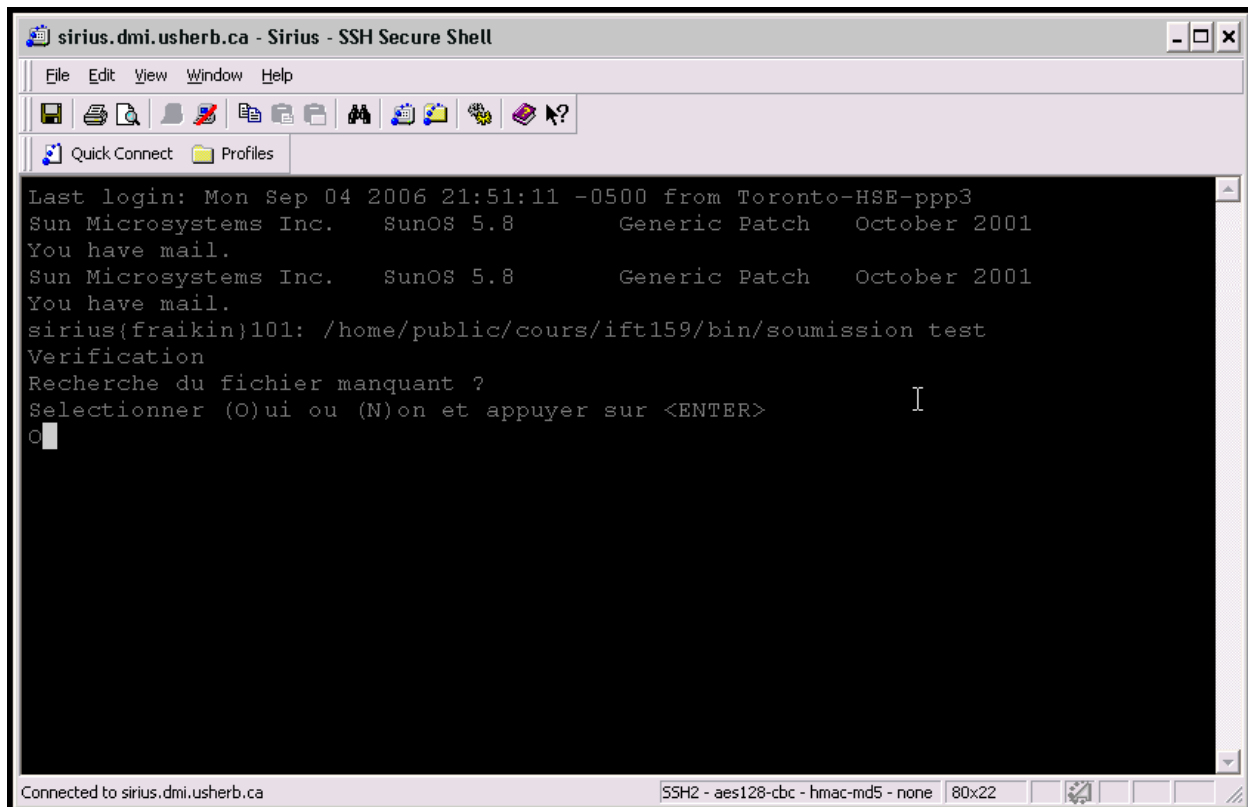
/home/public/cours/ift159/bin/soumission test.



```
sirius.dmi.usherb.ca - Sirius - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
Last login: Mon Sep 04 2006 21:50:27 -0500 from Toronto-HSE-ppp3
Sun Microsystems Inc. SunOS 5.8 Generic Patch October 2001
You have mail.
Sun Microsystems Inc. SunOS 5.8 Generic Patch October 2001
You have mail.
sirius(fraikin)101: /home/public/cours/ift159/bin/soumission test
```

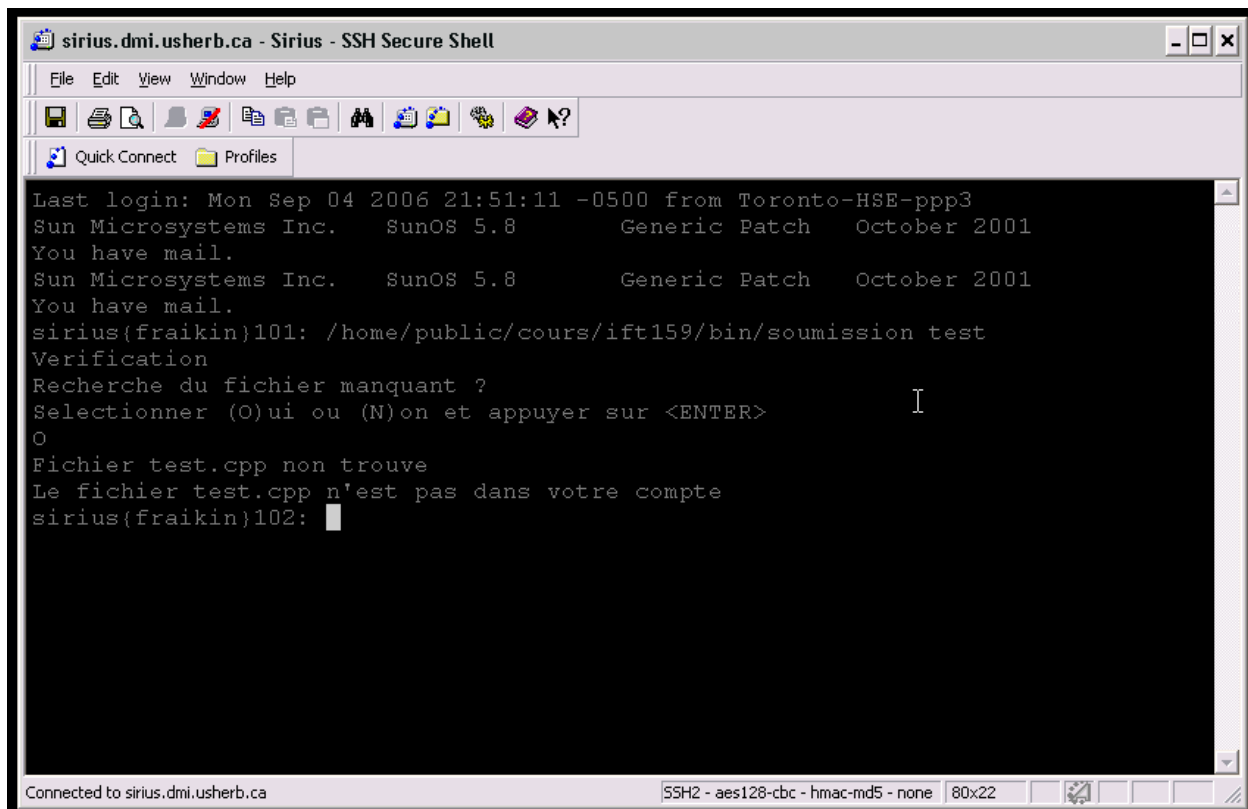
Connected to sirius.dmi.usherb.ca SSH2 - aes128-cbc - hmac-md5 - none 80x22

Il est important de se rappeler qu'un serveur UNIX (comme l'est Rigels) fait la différence entre les majuscules et les minuscules. Veuillez donc à respecter les noms exacts demandés. C'est une erreur courante de débutant de se tromper sur ce détail. Le fichier test.cpp n'est pas situé à la racine de votre compte (qui pour moi est /home/gradues/fraikin) et le programme ne va donc pas le trouver tout de suite. Il vous demandera s'il faut effectuer une recherche pour le retrouver. Dites oui en tapant un O (la lettre et non pas un zéro) et en appuyant sur <Enter>.



```
sirius.dmi.usherb.ca - Sirius - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
Last login: Mon Sep 04 2006 21:51:11 -0500 from Toronto-HSE-ppp3
Sun Microsystems Inc. SunOS 5.8 Generic Patch October 2001
You have mail.
Sun Microsystems Inc. SunOS 5.8 Generic Patch October 2001
You have mail.
sirius{fraikin}101: /home/public/cours/ift159/bin/soumission test
Verification
Recherche du fichier manquant ?
Selectionner (O)ui ou (N)on et appuyer sur <ENTER>
O
```

Si vous avez effectué une erreur (comme par exemple avoir mis une majuscule à votre nom de fichier ou lui avoir donné un autre nom) le programme ne pourra détecter de fichier correct et il se terminera.



```
sirius.dmi.usherb.ca - Sirius - SSH Secure Shell
File Edit View Window Help
[Icons]
Quick Connect Profiles

Last login: Mon Sep 04 2006 21:51:11 -0500 from Toronto-HSE-ppp3
Sun Microsystems Inc. SunOS 5.8 Generic Patch October 2001
You have mail.
Sun Microsystems Inc. SunOS 5.8 Generic Patch October 2001
You have mail.
sirius(fraikin)101: /home/public/cours/ift159/bin/soumission test
Verification
Recherche du fichier manquant ?
Selectionner (O)ui ou (N)on et appuyer sur <ENTER>
O
Fichier test.cpp non trouve
Le fichier test.cpp n'est pas dans votre compte
sirius(fraikin)102: 
```

Connected to sirius.dmi.usherb.ca SSH2 - aes128-cbc - hmac-md5 - none 80x22