KUTZTOWN
UNIVERSITY

Department of Computer Science

Introduction to Unix Tutorial

Authored by Nathaniel Moyer

Edited by Dr. Lisa Frye, Dr. Dylan Schwesinger, Henry Morales, Patrick Perrin and Reiley Walther

Table of Contents

1.0 Introduction

Here at Kutztown University, many courses in the department of Computer Science and Information Technology use on-campus Unix servers as a platform for coursework. These servers provide a Command Line Interface (CLI) rather than a Graphical User Interface (GUI), which may seem intimidating to those without prior experience. However, the objective of this introduction is to make the process of learning how to use them easy. Since they are a standard, it is necessary to learn how to access these servers, how to navigate them, and how to complete coursework on them.

Note: If at any point you want further elaboration, you can refer to the detailed instructions in section 5.0 Additional Resources.

2.0 Connecting to the Kutztown University Unix Servers

The simplest approach to accessing the Kutztown University Unix servers is by using a command line interface provided by the operating system of your computer. If you would prefer an interface that can be personalized, and is faster to open after setup, then using PuTTY is an alternative approach.

2.1 Using an OS provided command line Interface

As an example, I will use cmd.exe, or Command Prompt, which is an application provided by Windows 10. Other operating systems such as ChromeOS and macOS have applicable interfaces called Terminal.

Step 1: Open the command line interface provided by your computer.

Step 2: Use the SSH protocol to connect to acad.kutztown.edu using your KU User ID (ex. jsmit123). In other words, enter the command:

“ssh -l [Your KU User ID] acad.kutztown.edu”

```
Ex:     ssh -l  jsmit123  acad.kutztown.edu
```

Note: If using an operating system other than Windows, this command might be different.

Step 3: If it is your first time logging in from your device, you should be prompted with a question. This question is a security measure to verify the identity of the server. Enter “yes” as a response, unless you don’t trust that this is the correct server, in which case you should enter “no”, but you will then be unable to connect to the server.

Step 4: Then, enter the same password you use to connect to the Kutztown University student portal.

Note: Your input will not be visible for the sake of password protection.

2.2 Using PuTTY

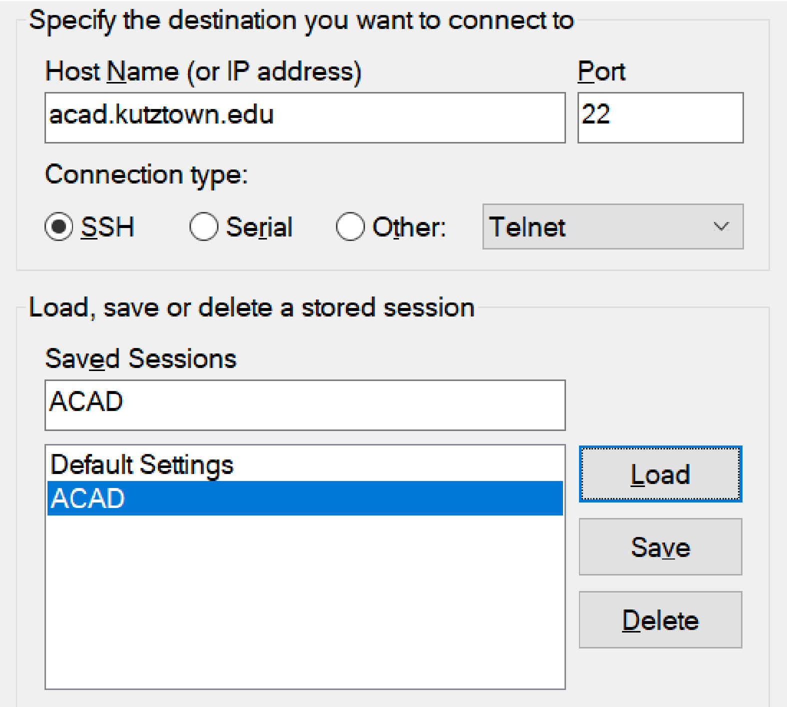
Step 1: Download PuTTY. You can download it through the following link:

https://download.kutztown.edu/Download/ComputerScience

The login credentials are the same ones you use to log in to the Kutztown University Student Portal.

Step 2: Open PuTTy and configure the settings in the “Session” category as follows:

A. Enter “acad.kutztown.edu” as the **Host Name (or IP address).**
B. Set the **Port** to “22”.
C. Set the **Connection type** to “SSH”.
D. Under **Saved Sessions**, enter a name of your choosing (ex. ACAD) to call these settings.



Specify the destination you want to connect to

Host Name (or IP address): acad.kutztown.edu

Port: 22

Connection type:

● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions: ACAD

Default Settings
ACAD

Step 3: Click save.

Step 4: Select the saved session that you named from the list, click “Load”, and then click “Open”.

Step 5: Enter your Kutztown University User ID (ex. jsmit123).

Step 6: If it is your first time logging in from your device, you should be prompted with a question. This question is a security measure to verify the identity of the server. Enter “yes” as a response, unless you don’t trust that this is the correct server, in which case you should enter “no”, but you will then be unable to connect to the server.

Step 7: Then, enter the same password you use to connect to the Kutztown University student portal.

Note: Your input will not be visible for the sake of password protection.

3.0 Navigating the Unix Servers

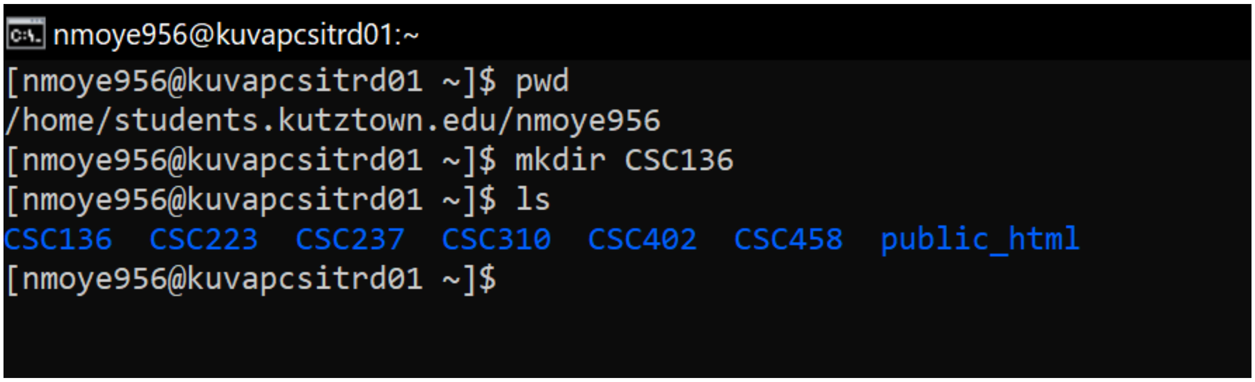
For the purpose of attaining the skills necessary to complete coursework, we will go through a demonstration on how to add assignments to your workspace while taking some extra steps along the way. As an example, I will be adding an assignment from the course "Computer Science II" to my workspace.

Step 1: Enter the command **"pwd".** This command will Print your Working Directory, and will be useful in ensuring you are in your desired directory.

Step 2: Enter **"mkdir [directory name]"** to make a directory for your course. This is an optional step, but it is useful in keeping your home directory organized. Naming it by the course number keeps the name short, and consequently makes navigating to this directory faster. When naming your directory, you should avoid using spaces in the directory name. For example, if you are using this server for CPSC135, then you may want to create a directory for that class using the command

mkdir cpsc135

Step 3: Enter **"ls"** to see the newly created directory in your current directory. The ls command stands for "list" and will list everything in your current working directory. You may want to also use the -l option to see a "long" listing of the directory contents, which will include some detailed information about each directory / file in that current working directory. If using the -l option, the command would be "**ls -l**".

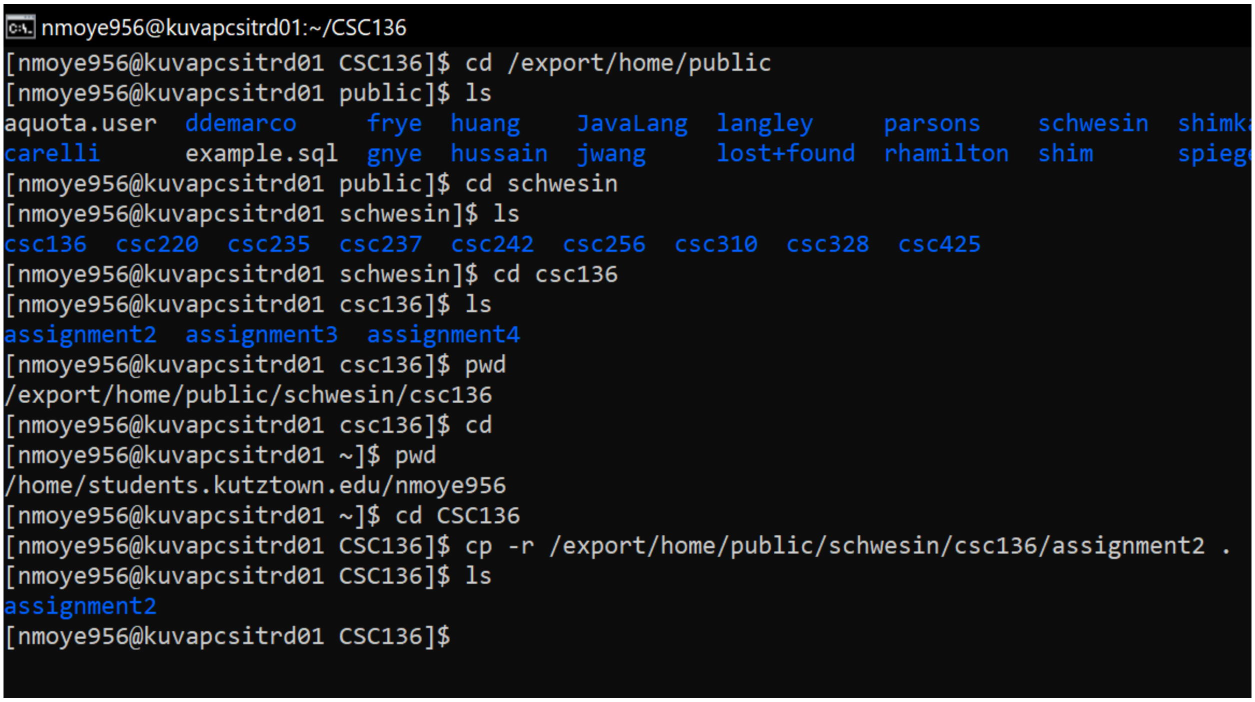


nmoye956@kuvapcsitrd01:~

```
[nmoye956@kuvapcsitrd01 ~]$ pwd
/home/students.kutztown.edu/nmoye956
[nmoye956@kuvapcsitrd01 ~]$ mkdir CSC136
[nmoye956@kuvapcsitrd01 ~]$ ls
CSC136  CSC223  CSC237  CSC310  CSC402  CSC458  public_html
[nmoye956@kuvapcsitrd01 ~]$
```

Step 4: Enter the command **“cd [directory name]”** to change directory to the specified directory. You can enter the command **“cd”** to return to your home directory at any time, or **“cd ..”** to the parent directory. Note that **.** represents the current working directory and **..** represents the parent directory.

Step 5: Once you know the path to your assignment, which should be supplied by the instructor, enter **“cp -r [source directory] .”** to copy the source of your assignment to your current directory if your assignment is contained within a directory, or **“cp [source directory/filename] .”** if it is contained within a single file. Be sure not to miss the **.** at the end of the cp command. The **.** in this command represents the current directory and will be the destination for the file being copied.



```
nmoye956@kuvapcsitrd01:~/CSC136
[nmoye956@kuvapcsitrd01 CSC136]$ cd /export/home/public
[nmoye956@kuvapcsitrd01 public]$ ls
aquota.user  ddemarco     frye  huang    JavaLang  langley     parsons    schwesin  shimk
carelli      example.sql  gnye  hussain  jwang     lost+found  rhamilton  shim      spieg
[nmoye956@kuvapcsitrd01 public]$ cd schwesin
[nmoye956@kuvapcsitrd01 schwesin]$ ls
csc136  csc220  csc235  csc237  csc242  csc256  csc310  csc328  csc425
[nmoye956@kuvapcsitrd01 schwesin]$ cd csc136
[nmoye956@kuvapcsitrd01 csc136]$ ls
assignment2  assignment3  assignment4
[nmoye956@kuvapcsitrd01 csc136]$ pwd
/export/home/public/schwesin/csc136
[nmoye956@kuvapcsitrd01 csc136]$ cd
[nmoye956@kuvapcsitrd01 ~]$ pwd
/home/students.kutztown.edu/nmoye956
[nmoye956@kuvapcsitrd01 ~]$ cd CSC136
[nmoye956@kuvapcsitrd01 CSC136]$ cp -r /export/home/public/schwesin/csc136/assignment2 .
[nmoye956@kuvapcsitrd01 CSC136]$ ls
assignment2
[nmoye956@kuvapcsitrd01 CSC136]$
```

Step 6 (Optional): At this point, you should be set to start learning how to work on your assignments. If you want to reorganize your home directory, you can use **“rm [target filename]”** or **“rm -r [target directory]”** to remove files/directories. Note that this command will delete the file or directory.

Use the “rm -r” command VERY carefully. You may want to always use the -i option for the rm command, which will confirm you want to remove the file or directory before deleting it. Once deleted, there is no easy way to retrieve the deleted file or directory.

Similarly, you can use **“mv [source directory] [target directory]”** to move a file, or to move a directory. There is no rename command, so the mv command can be used to rename a file or directory.

4.0 Coding on the Unix Servers

An intuitive way to write code on the Kutztown University Unix servers is by using Notepad++. This requires an initial setup, but once completed, you will have a modern source code editor available to you. This additionally makes transferring files from your computer to the Unix servers a simple process. If you would prefer to, or must use a different source code editor, then Emacs is a highly compatible alternative.

4.1 Using Notepad++

Step 1: Download Notepad++. You can download it through the following link: https://notepad-plus-plus.org/downloads/

Step 2: Open Notepad++. If it is your first time using Notepad++, opt for the default settings during installation.

Step 3: Go to the “Plugins” tab and click on “Plugins Admin”. Then select “NppFTP” on the Plugins Admin window, install it, and click “Yes” on the pop-up to restart Notepad++.

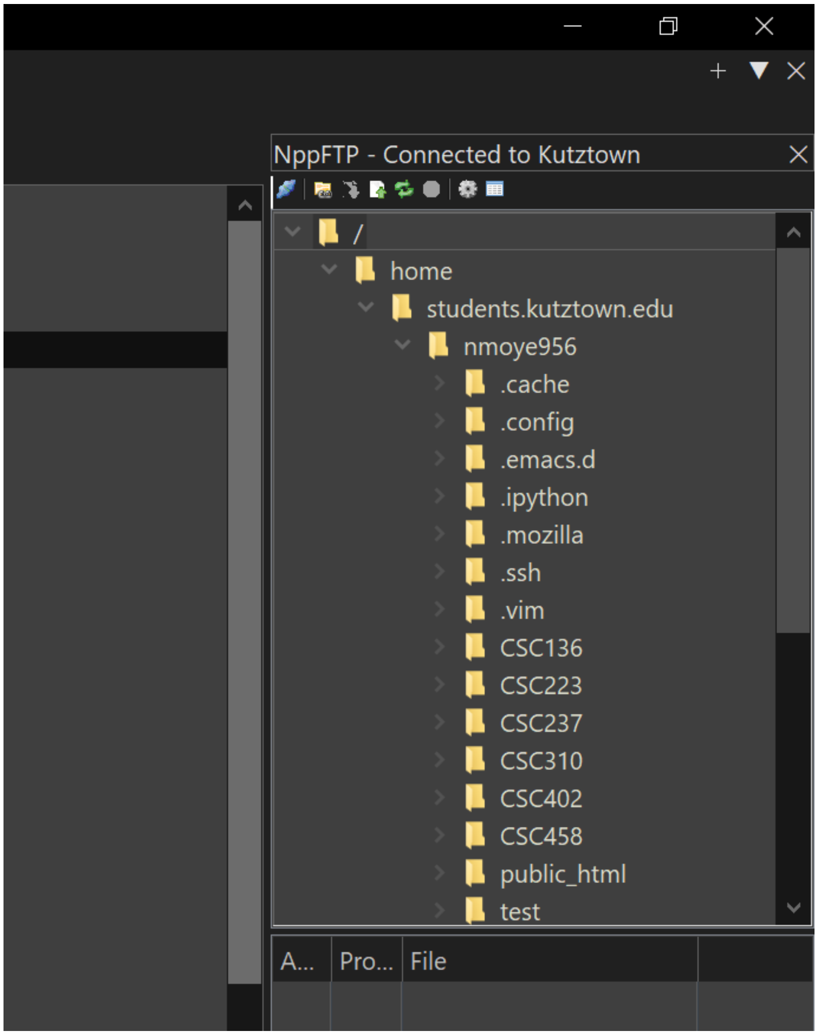
Step 4: If you do not see the NppFTP window on the side after restarting, go to the “Plugins” tab, hover over “NppFTP”, and click “Show NppFTP Window”.

Step 5: Click on the gear on the top right of that window, and then click “Profile settings”.

Step 6: Click “Add new” on the Profile settings window, and configure the settings as follows:

A. Enter “acad.kutztown.edu” as the **Host Name.**
B. Set the **Connection type** to “SFTP”.
C. Set the **Port** to “22”.
D. Set the **Username** as your Kutztown University username, not the full email address. (ex. jsmit123)
E. Either set the **Password** to the password of your Kutztown University email account, OR check the box labeled “Ask for password”.

Step 7: Double-click on your newly created profile in the NppFTP window, and then you will have access to the files within your home directory on acad. <u>Be sure you are still connected to acad while updating your files, otherwise changes to your work will not be saved</u>.



Additional Tips:

A. To transfer files from your computer to the Unix servers, simply drag the files from the file explorer of your computer into one of the folders in the NppFTP window, like those seen above.

B. You can change the tab settings by first going to the “Settings” tab of the main window and clicking on “Preferences”. Then, click on the “Language” tab of the Preferences window. “Tab Settings” can then be found on the right side of the Preferences window.

C. When you exit Notepad++ or close your laptop, you will be disconnected from the Unix server and will have to reconnect to the server when you want to continue your work. Be sure to save your work before exiting Notepad++ or closing your laptop.

4.2 UNIX Editors

There are a variety of editors available on the Unix servers for editing files using the Command Line Interface (CLI). Common editors are Nano, Emacs, and vi. The vi editor is the default text editor on most Unix systems, but it is not intuitive and not recommended for beginners.

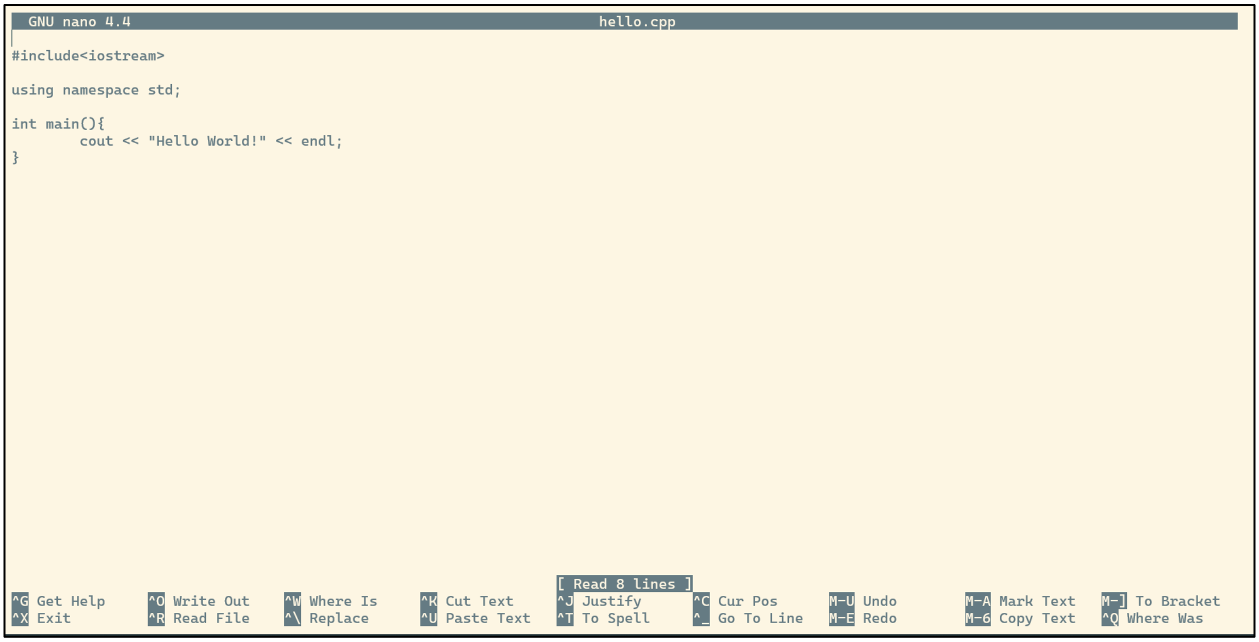
Using Nano

Nano is a simple editor with a built-in menu system at the bottom of the screen. This is highly recommended for new users. In the menu system, the ^ represents the Control (Ctrl) key on the keyboard. For example, the menu states “^X Exit”, which means you need to press the Ctrl key and the X key at the same time to exit this editor.

To use the Nano text editor, simple type the command **“nano [filename]”** to open the file in Nano. If the file already exists, it will be opened with Nano and you will see any source code that exists in the file. If the file does **not** exist, then the Nano text editor will open a new file with the name that you give it. This will show a blank file with the built-in menu at the bottom of the screen, as mentioned above.

An example can be seen on the next page:

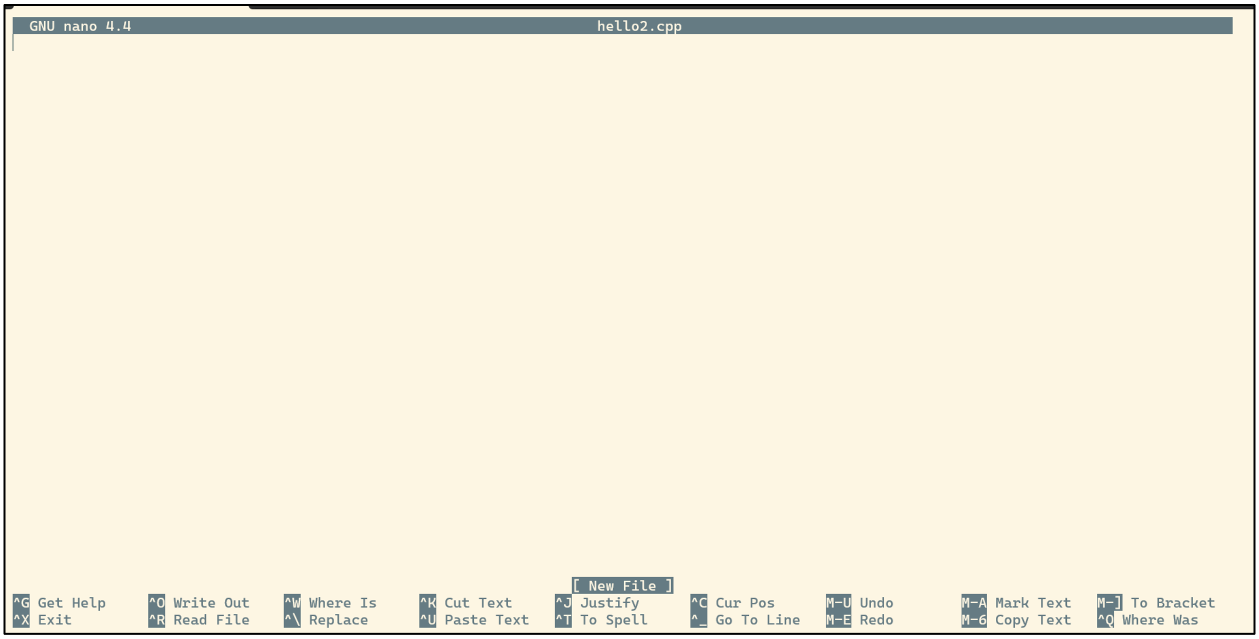
Existing file: *After entering the command:* ***nano hello.cpp***



```
  GNU nano 4.4                                          hello.cpp

#include<iostream>

using namespace std;

int main(){
        cout << "Hello World!" << endl;
}
```

```
[ Read 8 lines ]
```

New file: *After entering the command:* ***nano hello2.cpp***



```
  GNU nano 4.4                                          hello2.cpp
```

```
[ New File ]
```

Using Emacs

Emacs is another editor on many Unix systems, including those at KU. It is a powerful text editor, but is not intuitive and not recommended for beginners.

Step 1: Navigate to the directory containing the file you would like to edit using the cd command to change directory. Refer to section 3.0 Navigating the Unix Servers if you are inexperienced with navigating Unix, or see the common Unix commands in section 5.0 Additional Resources if you need a refresher.

Step 2: Enter the command: **"emacs [filename]"** to open the file in Emacs.

Once completing these two steps, you will be in the text editor, Emacs. Here you can use the arrow keys to navigate and make changes to the file you opened. Additionally, you will need the following commands to close Emacs:

ctrl+x ctrl+s	saves changes to the file
ctrl+x ctrl+c	closes emacs

These commands involve using two hotkeys for each command. For example, holding the **control key** while pressing **x**, and then sequentially holding the **control key** while pressing **s** will save the file. Emacs is home to many commands such as these, and if you are interested in them, they can be found by following the link below: https://csit.kutztown.edu/UnixWorkshop/emacs_commands_list.html

Using Vim

Vim is the final text editor on many Unix systems, including those at KU. It is also an incredibly powerful text editor with multiple modes to use, but the most popular two are its command-line mode and its insert mode. Vim is not intuitive and only recommended for *advanced users* who know how to navigate a text editor and/or have experience with Vi or Vim already.

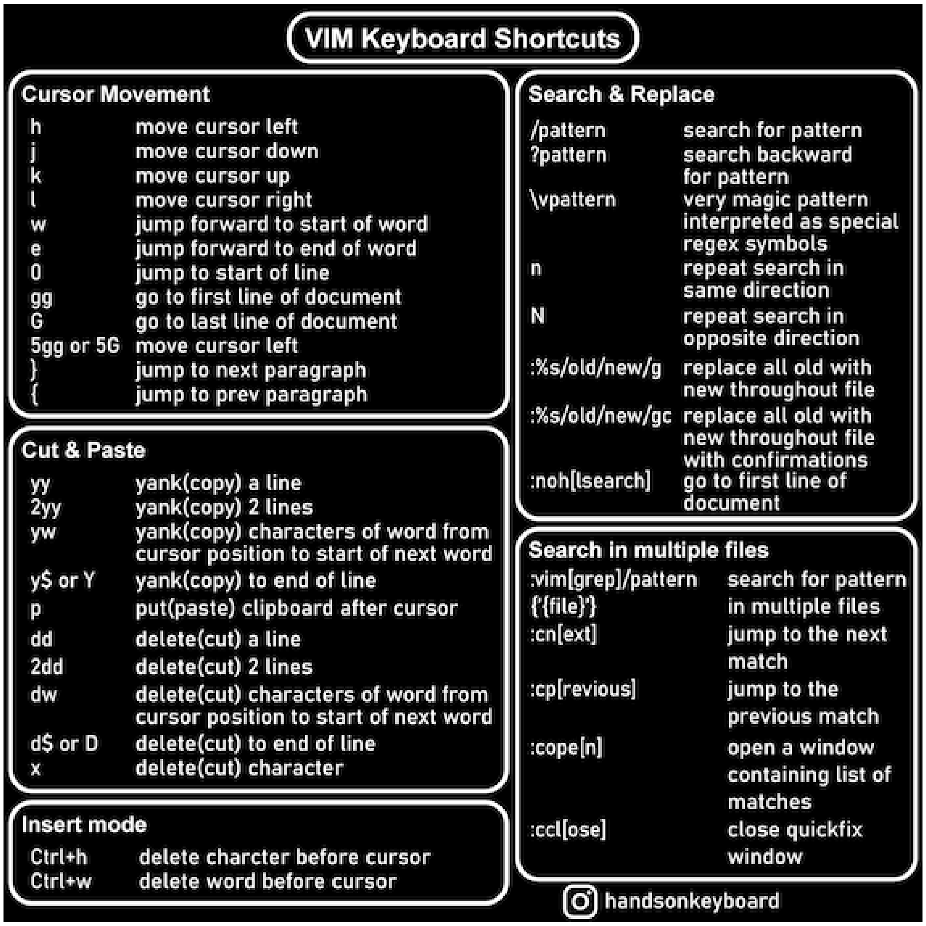
To use the Vim text editor, simple type the command **"vim [filename]"** to open the file in Vim. If the file already exists, similarly to nano and emacs, it will be opened with Vim and you will see any source code that exists in the file. If the file does **not** exist, then the Vim text editor will open a new file with the name that you give it.

.

CAUTION: If you enter Vim, it can be challenging to get out of this text editor if you are not already familiar with the exit procedure.
The procedure can be found below:

1. If you are in Vim, you will need to get into command-line mode, or normal mode. To do this, simply press the Esc key on your keyboard. If there is a prompt at the bottom of the screen, by pressing this, the prompt will disappear.
2. Now that you are in command-line mode, or normal mode, there are two basic ways to exit Vim: exit and save, or exit without saving:
   a. To exit and save, enter "**:wq**" (colon-q-w). Press and hold shift and press the colon key, then enter 'w' and 'q'. Press enter. You will be taken out of Vim and your changes will be saved. The 'w' key will 'write' any changes to the file, and the 'q' key will quit Vim.
   b. To exit without saving, enter "**:q!**" (colon-q-!). Press and hold shift and press the colon key, then enter 'q' and '!'. Press enter. You will be taken out of Vim BUT your changes will **not** be saved.

Below is a quick reference guide on some commands to navigate Vim:



VIM Keyboard Shortcuts

Cursor Movement

h	move cursor left
j	move cursor down
k	move cursor up
l	move cursor right
w	jump forward to start of word
e	jump forward to end of word
0	jump to start of line
gg	go to first line of document
G	go to last line of document
5gg or 5G	move cursor left
}	jump to next paragraph
{	jump to prev paragraph

Cut & Paste

yy	yank(copy) a line
2yy	yank(copy) 2 lines
yw	yank(copy) characters of word from cursor position to start of next word
y$ or Y	yank(copy) to end of line
p	put(paste) clipboard after cursor
dd	delete(cut) a line
2dd	delete(cut) 2 lines
dw	delete(cut) characters of word from cursor position to start of next word
d$ or D	delete(cut) to end of line
x	delete(cut) character

Insert mode

Ctrl+h	delete charcter before cursor
Ctrl+w	delete word before cursor

Search & Replace

/pattern	search for pattern
?pattern	search backward for pattern
\vpattern	very magic pattern interpreted as special regex symbols
n	repeat search in same direction
N	repeat search in opposite direction
:%s/old/new/g	replace all old with new throughout file
:%s/old/new/gc	replace all old with new throughout file with confirmations
:noh[lsearch]	go to first line of document

Search in multiple files

:vim[grep]/pattern {'{file}'}	search for pattern in multiple files
:cn[ext]	jump to the next match
:cp[revious]	jump to the previous match
:cope[n]	open a window containing list of matches
:ccl[ose]	close quickfix window

handsonkeyboard

A More detailed “cheat sheet” can be found here: https://vim.rtorr.com/

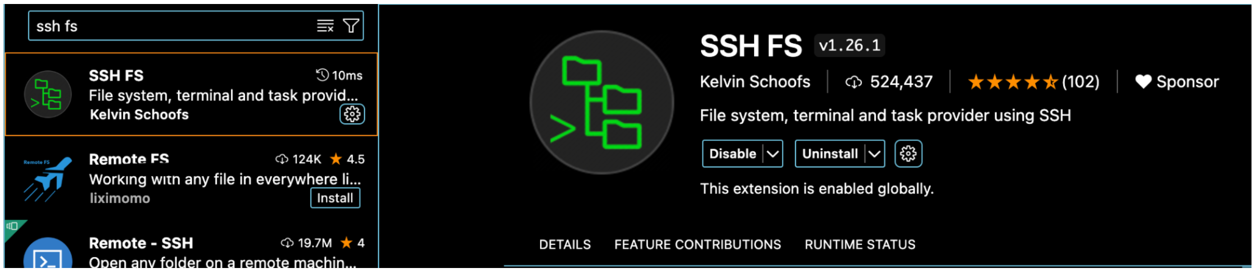
Visual Studio Code (VS Code)

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

It is possible to connect VS Code to the ACAD server at Kutztown University. There, you will be able to use a terminal to navigate ACAD, and you will be able to use the GUI file system layout in VS Code as well. Below are instructions on how to set this up:

VS Code for ACAD Instructions:

1. Download VS Code for your computer/operating system.
2. Open VS Code and navigate through your setup steps.
3. On the far-left vertical navigation pane, click on the "**extensions**" icon. This will look like building blocks (if your mouse is hovered over it, it will show the word 'extensions').
4. Search for an extension named "**SSH FS**". This extension is by Kelvin Schoofs and has a green and black icon. IMPORTANT - do **NOT** use "Remote SSH" since this will cause problems with the ACAD server.



ssh fs

SSH FS 10ms
File system, terminal and task provid...
Kelvin Schoofs

Remote FS 124K ★ 4.5
Working with any file in everywhere li...
liximomo

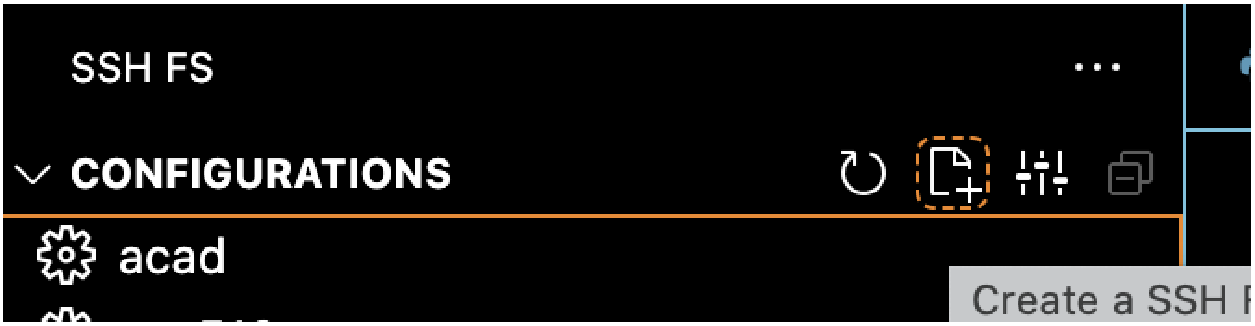
Remote - SSH 19.7M ★ 4
Open any folder on a remote machin...

SSH FS v1.26.1
Kelvin Schoofs | 524,437 | ★★★★★ (102) | ♥ Sponsor
File system, terminal and task provider using SSH

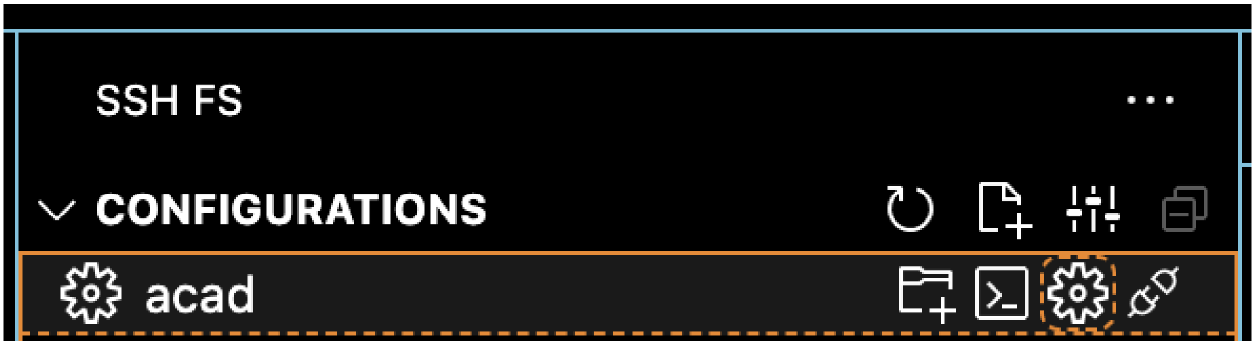
This extension is enabled globally.

5. Install the extension and click the new icon in the far-left navigation pane. This will look like a folder.

6. Under the configurations pane, create a new SSH FS configuration as shown below:



7. Give the configuration a name, for example “acad” or ‘KU Acad” Save this and close out of the edit config page.
8. Click on the cog wheel to open the settings pane



For some reason, you must scroll to the bottom and click the Cancel button. Then refresh the configurations (the circle with the arrow in the above image)

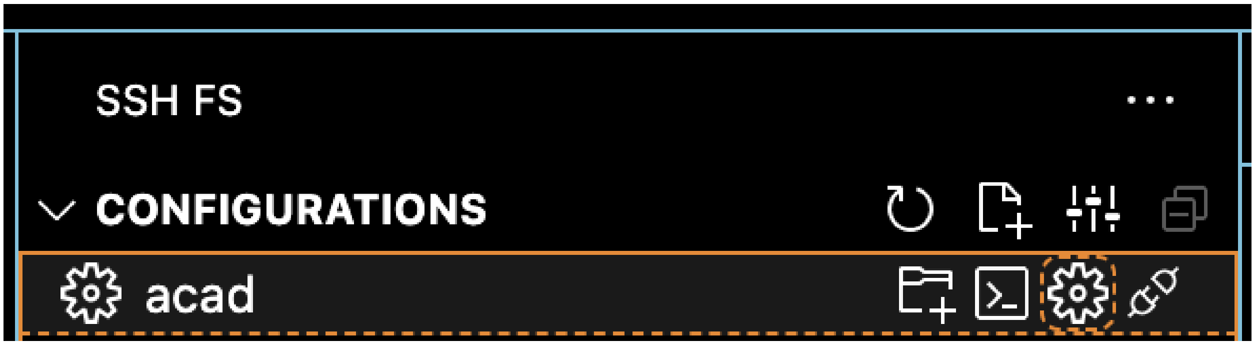
9. Click on the cog wheel again to be able to open and save the following settings

○ Host:	“acad.kutztown.edu”
○ Port:	“22”
○ Root:	”~”
○ Username:	“your first part of your kutztown email” e.g. jsmit123

IMPORTANT when it comes to your PASSWORD, it is **strongly recommended that you DO NOT** enter this into the configuration page for multiple reasons. This is in plaintext and not hidden, and when you inevitably change your password, it is likely that you will not remember to change it in the configuration page since it does not automatically update.

INSTEAD, as tedious as it may be, entering your password each time when you are prompted is recommended.

10. Scroll to the bottom and click the "**SAVE**" button.
11. To open your files, click on the folder with the plus next to it (the first option)
12. To open a terminal window (the same as you get using Putty or ssh) click the square next to the folder.



5.0 Additional Resources

Common Unix Commands

pwd	prints path for the present working directory
ls	lists files in the current directory
mkdir [directory name]	makes a new directory
cd [target directory]	changes directory to the target directory
cd	change directory to your home directory
cd ..	change directory to the parent directory
cp [source directory/filename] **.**	copies file from source directory to current directory, represented by the **.**
cp [source directory/filename] [target directory]	copies file from source directory to target directory
cp -r [source directory] [target directory]	copies the entire source directory to target directory
rm [target filename]	removes the target file
rm -r [target directory]	removes the entire target directory
mv [source directory/filename] [target directory]	moves file from source directory to target directory (rename file if source and target directory are the same)
nano [filename]	opens file in nano for editing
emacs [filename]	opens file in emacs for editing
vim [filename]	opens file in vim for editing

Detailed Instructions

Setting Up PuTTY on a Windows Machine: https://www.kutztown.edu/Departments-Offices/A-F/ComputerScienceInformationTechnology/Documents/Student%20Resources/Putty_config_UNIX.pdf

More Unix Commands:
https://csit.kutztown.edu/UnixWorkshop/basic-unix.html

More Emacs Commands:
https://csit.kutztown.edu/UnixWorkshop/emacs_commands_list.html

Setting Up Notepad++ on a Windows Machine:
https://csit.kutztown.edu/UnixWorkshop/manuals/NotepadPlusPlusManual.pdf

The Original Kutztown University Introduction to Unix:
https://csit.kutztown.edu/UnixWorkshop/