

## Session 2: File Operations



P. S. Langeslag

25 October 2018

## Correction

In next week's homework, please replace the address  
`http://www.gutenberg.org/38334/38334-0.txt` with  
`http://langeslag.uni-goettingen.de/38334-0.txt`.

# Logging into the Course Terminal

1. You need to be on the campus network, physically or by VPN.
2. You'll need an ssh client:
  - ▶ Included in all Linux distributions, all versions of OS X;
  - ▶ On Windows, install [PuTTY](#).  
(stock ssh client trickier to set up for X11 forwarding)
3. To make use of graphical applications on Windows, you'll also need [Xming](#) or [Cygwin](#).
4. Log onto [langeslag.uni-goettingen.de](https://langeslag.uni-goettingen.de) on port 22 with the credentials I have sent you.

# PuTTY Settings (Basic)

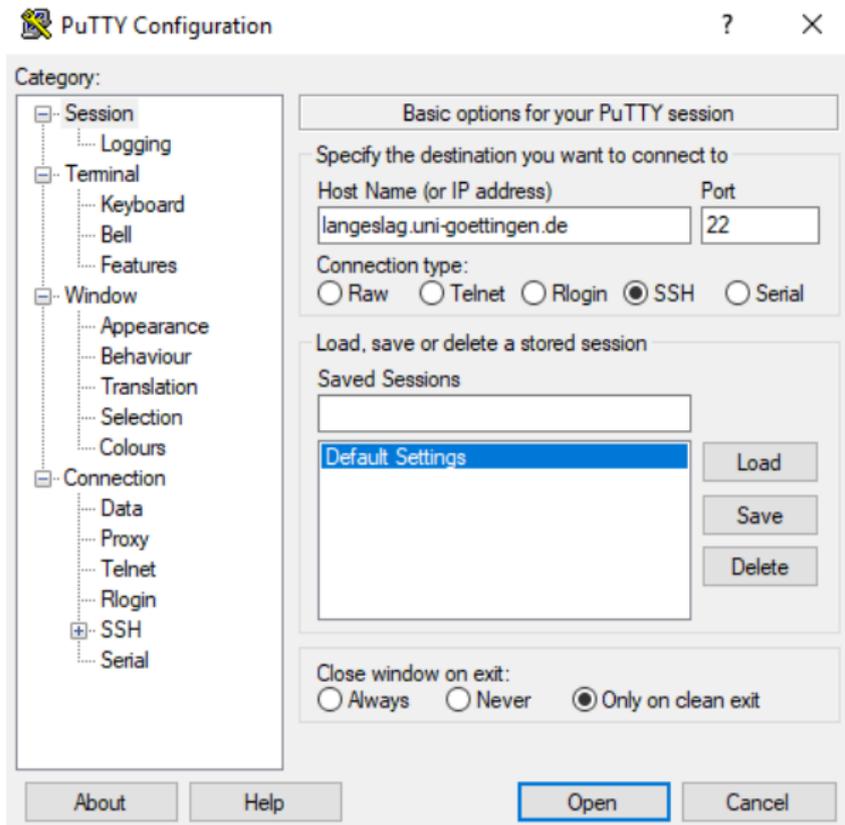


Figure: PuTTY: connection settings

# PuTTY Settings (X11 Forwarding)

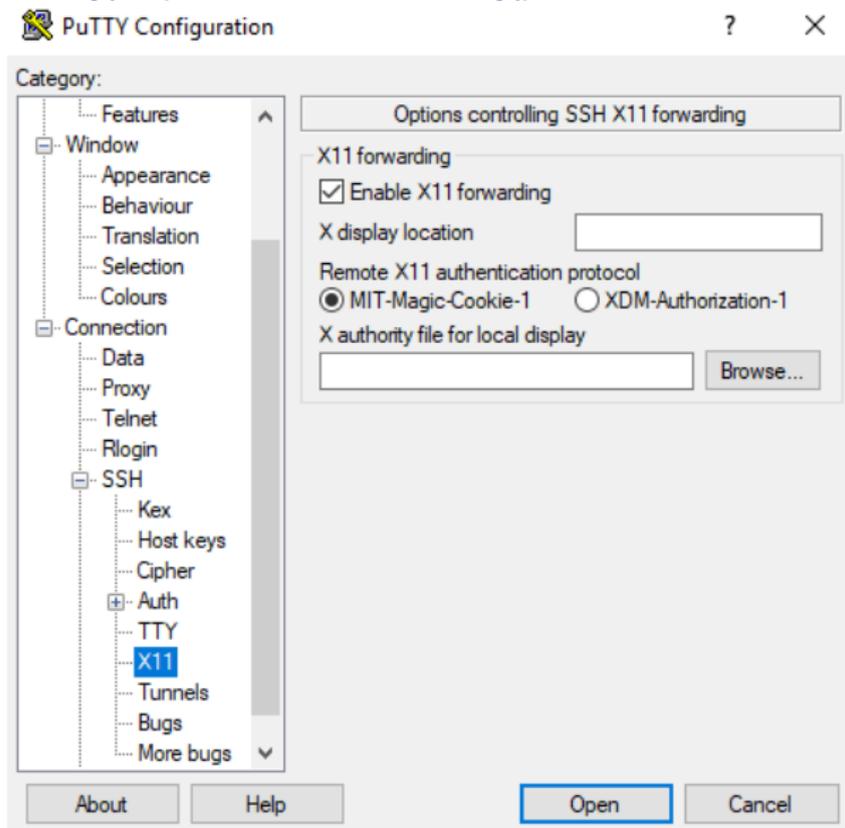


Figure: PuTTY: X11 forwarding

# PuTTY With Xming

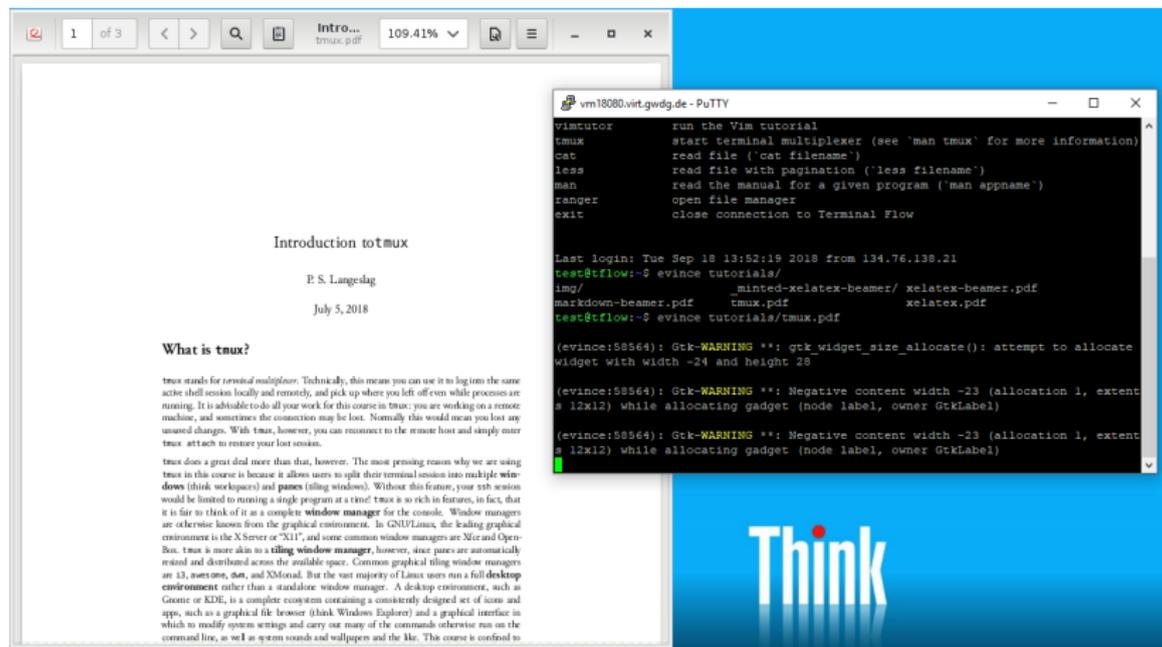


Figure: PuTTY with Xming

# PuTTY Settings (Appearance)

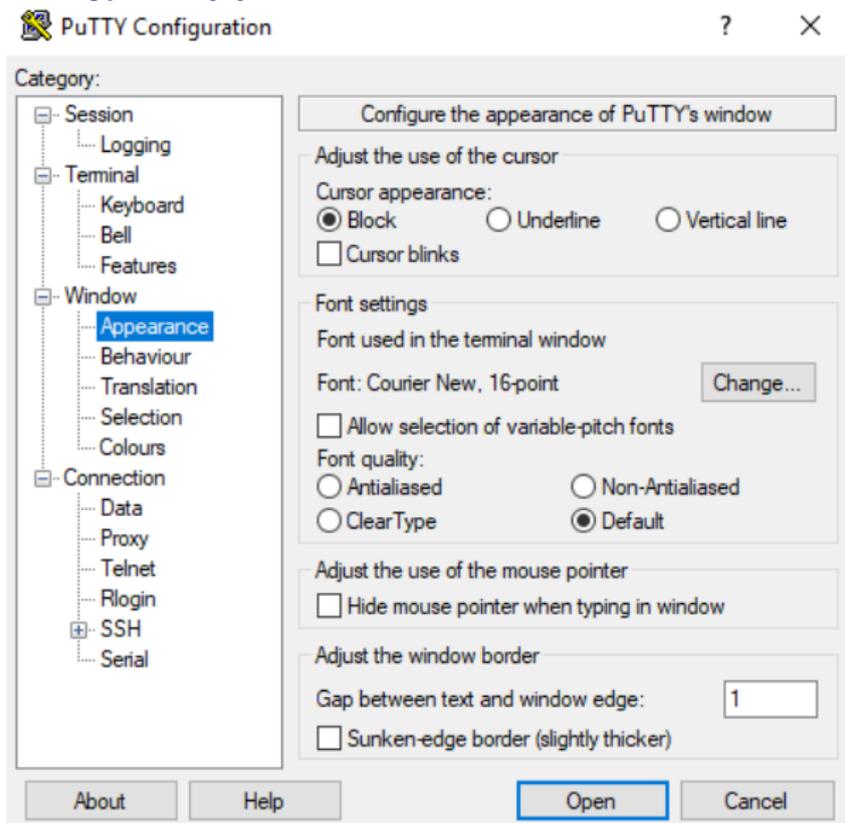


Figure: PuTTY: X11 forwarding

## ssh on Linux or Mac

```
$ ssh -Y username@langeslag.uni-goettingen.de
```

## If You Encounter X11 Forwarding Issues

1. Disable X11 forwarding (e.g. drop the `-Y` argument)
2. Modify your `~/.latexmkrc` as follows:

```
< $pdf_previewer = 'evince';  
> $pdf_previewer = 'less';
```

3. Create a file `~/.bashrc` with the following content:

```
export PDFVIEWER_texdoc="less"
```

- ▶ Issues will remain because I haven't configured an explicit fallback option.
- ▶ If you get X11 warning messages in spite of a correct setup, try to ignore them.

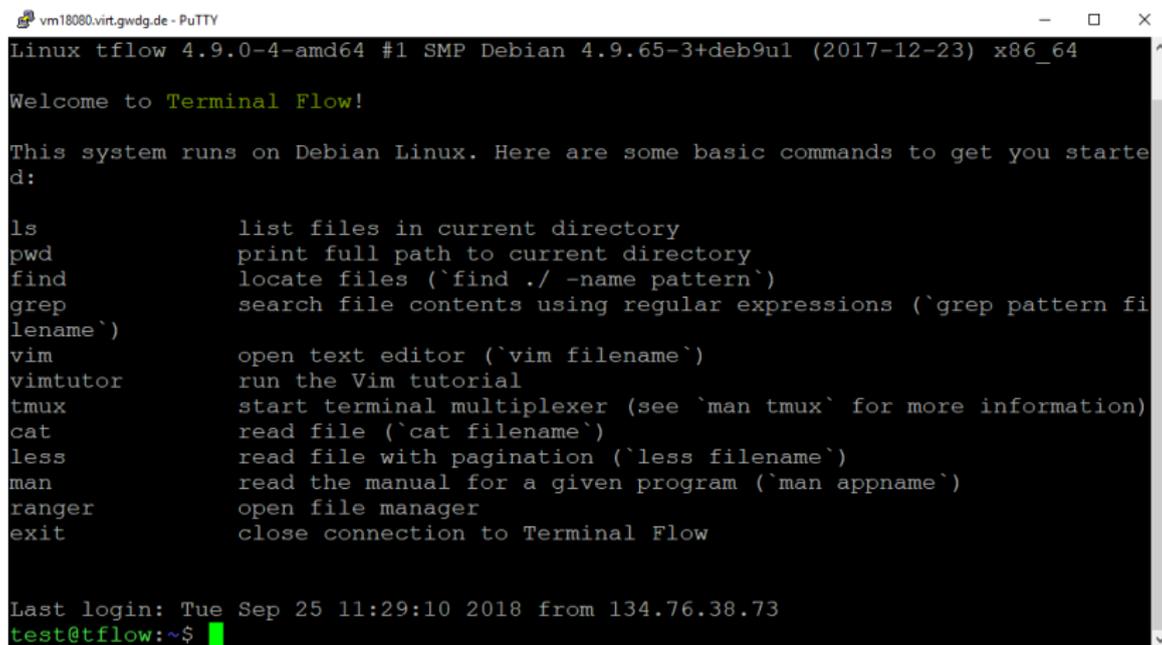
# Virtual Private Network (VPN)

See [https://info.gwdg.de/docs/doku.php?id=en:services:network\\_services:vpn:start](https://info.gwdg.de/docs/doku.php?id=en:services:network_services:vpn:start)

## Linux

```
sudo ip tuntap add mode tun tun0  
sudo ip link set dev tun0 up  
sudo openconnect 134.76.22.1
```

# The Shell in PuTTY

A screenshot of a PuTTY terminal window. The title bar at the top reads "vm18080.virt.gwdg.de - PuTTY". The terminal content shows the system information: "Linux tflow 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x86\_64". Below this is a green "Welcome to Terminal Flow!" message. A block of text lists basic Linux commands and their uses. At the bottom, it shows the last login time and the current shell prompt "test@tflow:~\$".

```
vm18080.virt.gwdg.de - PuTTY
Linux tflow 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x86_64

Welcome to Terminal Flow!

This system runs on Debian Linux. Here are some basic commands to get you started:

ls                list files in current directory
pwd              print full path to current directory
find            locate files (`find ./ -name pattern`)
grep            search file contents using regular expressions (`grep pattern filename`)
vim             open text editor (`vim filename`)
vimtutor        run the Vim tutorial
tmux            start terminal multiplexer (see `man tmux` for more information)
cat             read file (`cat filename`)
less           read file with pagination (`less filename`)
man            read the manual for a given program (`man appname`)
ranger         open file manager
exit           close connection to Terminal Flow

Last login: Tue Sep 25 11:29:10 2018 from 134.76.38.73
test@tflow:~$
```

Figure: The Terminal Flow welcome prompt (“MOTD”)

# Your Home Directory

---

/home/username/

experimental/

public\_html/

tutorials/

work/

.config/

.vim/

.vimrc\_background

.latexmkrc

.Xauthority

---

## Command Syntax

Program Name	Options	Arguments
true		
false		
pwd		
ls		
ls	-a	
date	+%A\ %d\ %B	
cd		~/experimental
ls	-lh	~/tutorials
rm	-rf	file1 dir1 dir2 file2

- ▶ Options are usually available in long form and shorthand;
- ▶ Shorthand options may be stacked except where they require arguments.

## Shell Basics

- ▶ The shell is case sensitive
- ▶ Directories are delimited by /
- ▶ ~ is a shorthand for /home/username/
- ▶ Cursor keys up and down navigate your command history
- ▶ Highlighting with the left mouse button copies to paste buffer
- ▶ Shift + Insert (or the middle mouse button) pastes from buffer (Linux has more than one past buffer; we'll discuss this for Vim)
- ▶ Space delimits between program names, options, and arguments, but can be escaped with \ or by quoting: `thesis\ final.docx` or `"thesis final.docx"`
- ▶ CTRL + L clears the screen (in most terminal emulators)
- ▶ Shift + PgUp/PgDn allows scrollbar (in most terminal emulators)

## In-Terminal Aid with Programs

- ▶ Tab completion
- ▶ `which`
- ▶ `man`

# Manual Sections

---

Section	Description
1	General commands
2	System calls
3	Library functions
4	Special files
5	File formats
6	Games and screensavers
7	Miscellaneous
8	System administration commands; daemons

---

## Navigating less

---

Official key	Action	Also permitted
j / ENTER	One line down	Cursor down
k	One line up	Cursor up
d	Half-screen down	
u	Half-screen back	
f / SPACE	Page down	PgDown
b	Page up	PgUp
g	Back to top	
G	Down to end	
/	Find	
n	Show next hit	
N	Show previous hit	
q	Quit	

---

. . . and there is more! See man less.

## grep Options

- ▶ `-i` for case insensitive searches
- ▶ `-P` for PCRE (Perl Compatible Regular Expression) searches

## PCRE Syntax

---

.	matches any one character
?	matches 0 or 1 of the previous character
*	matches 0 or more of the previous character
+	matches 1 or more of the previous character
{n}	matches the previous character exactly n times
{n,m}	matches the previous character between n and m times
[abc]	matches any of the bracketed characters
[^abc]	matches strings containing none of the bracketed characters
[a-zA-Z0-9]	matches characters from the bracketed ranges
	OR operator
^	matches the beginning of a string (i.e. line)
\$	matches the end of a string (i.e. line)
(abc)	group the sequence abc for further processing (e.g. (abc)+ matches abc, abcabc, etc.)

---

## PCRE Lookaround

---

Positive lookahead	<code>foo(?=bar)</code>
Negative lookahead	<code>foo(?!bar)</code>
Positive lookbehind	<code>(?&lt;=bar)foo</code>
Negative lookbehind	<code>(?&lt;!=bar)foo</code>

---

## sed Replacement (Risky! Use Vim Where Possible)

---

Operation	Effect
<code>sed s/He/She/ file</code>	Replace all instances of He in file with She (-s for “substitute”); print result to stdout, leaving file untouched
<code>sed -n s/He/She/p file</code>	Idem, but print affected lines only (-n for “no output”; p for “print”)
<code>sed -n s/He/She/pI file</code>	Idem, but case insensitive (I for “insensitive”); note unwanted effects
<code>sed -n s/He/She/gpI file</code>	Idem, replace beyond the first match in the string (sentence) (g for “global”)
<code>sed -i.bak s/He/She/ file</code>	Idem, but save results to original file and copy original file to backup file file.bak (-i for “insert”); no output
<code>sed -i s/He/She/ file</code>	Idem, but without backup file! Risky!
<code>sed -i s/He/She/ *</code>	Idem, but for every file in working directory. Highly risky!

---

## ls -l Long Listing Format

```
test@tflow:~/tutorials$ ls -l
total 2372
-rwxr-xr-x 1 test tflow2018    6519 Oct 12 09:50 biblatex.bib
-rwxr-xr-x 1 test tflow2018    1437 Oct 12 09:50 csquotes.tex
drwxr-xr-x 2 test tflow2018    4096 Oct 12 09:50 img
. . .
```

- ▶ directory yes/no
- ▶ file permissions user/group/others, read/write/execute
- ▶ number of hard links
- ▶ owner
- ▶ group
- ▶ filesize
- ▶ date
- ▶ filename

## chmod

Syntax: `chmod nnn filename`, where

- ▶ the numerical values concern owner (“user”), group, and others respectively
- ▶ each numerical value is an addition of the following values:

Value	Meaning
4	read
2	write
1	execute

Shorthands are e.g. `chmod +x filename` to make a file executable by all.