Statistical Machine Translation
Linux Command-line Cheat Sheet

Here are a few Linux commands which may come in handy when using the command-line to run Moses:

command	description
cd folder_name	moves, or **c**hanges **d**irectory, into the specified folder (folder_name)
ls folder_name	lists the contents of a given folder (folder_name)
mkdir folder_name	create a new folder (**m**ake **dir**ectory) with a given name (folder_name)
wc file_name	shows how many characters, words and lines are in a specified file (file_name). Useful for seeing if your file is empty, or if it has the correct number of lines
head file_name	displays the first 10 lines of a specified file (file_name)
tail file_name	displays the last 10 lines of a specified file (file_name)
pwd	prints out the full pathname of the current folder location (**p**rint **w**orking **d**irectory)
export Name="user/folder_name"	assign a name (Name) to a folder path (user/folder) so that it may be used quickly in other commands e.g. *ls Name* will output the files which are present in user/folder_name
echo Name	writes its arguments to standard output e.g. *echo Name* will output user/folder_name Useful for making sure that your exported variables are correctly assigned
*	0 or more (i.e. everything) - use in conjunction with other command e.g. *head ** will print out the first 10 lines of all files in the current folder
--help	displays a help screen - use in conjunction with another command e.g. *wc --help* will show the usage and options for the wc command

cp file_name folder_name	**cop**y a file (file_name) from original place to a new place (folder_name)
cp -r folder_name_1 folder_name_2	**cop**y a folder containing all the files and subfolders under this path (folder_name_1) a new place (folder_name_2)
nohup nice commands **>&**file_name **&**	"nohup" is a POSIX command to ig**no**re the **HUP** (hangup) signal. ">&file_name &" means to save the outputs of running commands to a file (file_name).
.	short for current path
> file_name	write something into the file (file_name)
< file_name	read the file (file_name)
rm file_name	**rem**ove (delete) file(s)
rm -r folder_name	**rem**ove (delete) files under the directory
ssh user@machine	remotely log in to another Linux machine, over the network. Leave an ssh session by typing **exit**.

Once you find an unknown Linux command, you could also google it and learn to understand it. There are also some useful websites on Linux command tutorial. You could learn it by yourself.

1, http://linuxcommand.org/index.php

2, http://ryanstutorials.net/linuxtutorial/