**UNIX commands**

Author: Luong Minh Thang

These are my random collection of UNIX stuff. I’ll arrange them once I collected enough things here !!!

You could find here several interesting stuff about UNIX commands like find, xargs, grep…., CVS, git, …

! Ubuntu

Problem: “W: Failed to fetch …/ubuntu/dists/maverick-updates/Release”

Fix:

(Ref: <https://help.ubuntu.com/community/Repositories/CommandLine> )

+ sudo vim /etc/apt/sources.list

+ Add the following two lines at the end:

deb http://ubuntu.oss.eznetsols.org/ubuntu/ maverick main

deb-src http://ubuntu.oss.eznetsols.org/ubuntu/ maverick main

(the alternative mirror links could be found here <https://launchpad.net/ubuntu/+archivemirrors> )

! Text editing

less TEXTFILE | **awk** '{print $1, "\t", $1}'

! GIT

\* Resolve binary file

<http://www.lostechies.com/blogs/joshuaflanagan/archive/2010/01/28/how-to-resolve-a-binary-file-conflict-with-git.aspx>

\*\* Resolve using mine: git checkout --ours path

\*\* Resolve using theirs: git checkout --theirs path

! VIM

\* Indent code block

<http://vim.wikia.com/wiki/Indent_a_code_block>

\* Indent source code

<http://vim.wikia.com/wiki/Indenting_source_code>

\* Sample vim settings file

\* Macro: <http://vim.wikia.com/wiki/Macros>

qq start recording to register q

... your complex series of commands

q stop recording

@q execute your macro

@@ execute your macro again

**Text Deletion Commands**

x Delete character

dw Delete word from cursor on

db Delete word backward

dd Delete line

d$ Delete to end of line

d^ (d caret, not CTRL d) Delete to beginning of line

# ! MySQL

\* Operators & functions: <http://dev.mysql.com/doc/refman/5.0/en/func-op-summary-ref.html>

\* Efficient join: <http://stackoverflow.com/questions/1725856/which-mysql-join-query-is-more-efficient>

! Latex

\* Vertical lines in tables

\usepackage{multirow} \usepackage{rotating}

Vertical text in the table in Latex

\multirow{4}{15mm}{\begin{sideways}\parbox{15mm}{text}\end{sideways}}

http://en.wikibooks.org/wiki/LaTeX/Packages/Rotating

\* Installing rotating package

\*\* http://www.ctan.org/tex-archive/macros/latex/contrib/rotating/

\*\* download rotating.ins and rotating.dtx files

\*\* cygwin/unix, run: latex rotating.ins

--> proce rotating.sty file

\*\* copy .cls or .sty files into C:\Program Files\MiKTeX 2.6\tex\latex\thang

\*\* Open the Start menu and select "MikTex/Settings". In the dialog box that now appears, go to the "General" tab and click the the "Refresh FNDB"

\*\* You're done! One the file name database is refreshed, check to make sure that it works.

# ! CVS

\* 21/05/10

\*\* Add new module CVS

cvs –d :ext:lmthang@cte.comp.nus.edu.sg:/home/cvs/Repositories/MarkusCollab/ import -m "Thang test" testDir vendor release

\* Before

[**http://users.piuha.net/martti/comp/cvs/cvs.html**](http://users.piuha.net/martti/comp/cvs/cvs.html)

\*\* Initial checkout

export CVS\_RSH=/usr/bin/ssh

export CVSROOT=":ext:rpnlpir@cte.comp.nus.edu.sg:/home/rpnlpir/CVSdir

cvs checkout -d project sg/edu/nus/comp/wing/starChallenge

\*\* Update

The cvs update command can be used to update the working copies. The following command checks for new directories (-d), removed empty directories (-P) and gets rid of so-called sticky tags (-A). It is also possible to get a certain version by using the -r ver option.

# cvs update -dPA

or

# cvs update -dPA -r name-of-the-branch

\*\* Add multiple files

The following commands can become handy if you are still stuck with cvs and cannot use the excellent eclipse cvs client.

\*\*\* Add normal file at top level

find . -maxdepth 1 -type f | xargs cvs add

\*\*\* Add all directories

find . -type d -print | grep -v CVS | xargs cvs add

\*\*\* Add all files

find . -type f -print | grep -v CVS | xargs cvs add

-n1 parameter for xargs:

find . -type d -print | grep -v CVS | xargs -n1 cvs add  
find . -type f -print | grep -v CVS | xargs -n1 cvs add

the -n1 ensures that cvs add is called only for a single file each time.

\* 05/12/09

Ignore the first line and output the remaining line to a file

sed -n '2,$p' file > file2

\* convert pdf file to text: pdftotext

\* compute the size:

du –hsc dir\_name

-h: human-readable

-s: summary

-c: total

\* find files

find . -name "rc.conf" –print

find . -name "rc.conf" -exec chmod o+r '{}' \;

This command will search in the current directory and all sub directories. All files named rc.conf will be processed by the chmod -o+r command. The argument '{}' inserts each found file into the chmod command line. The \; argument indicates the exec command line has ended.

<http://www.athabascau.ca/html/depts/compserv/webunit/HOWTO/find.htm>

\* scp [lmthang@aye.comp.nus.edu.sg:public\_html/problem\_list.pdf](mailto:lmthang@aye.comp.nus.edu.sg:public_html/problem_list.pdf) . : download file to local host (could recursively download a folder with –r option)

\* Decompressing files

\*\* tar -xzf archive.tar.gz -C ~/dest/ :extract to a specific folder

\*\* tar -tzf archive.tar.gz: list files in an archive

\*\* tar xvf file.tar

\*\* tar zxvf file.tar.gz

\*\* tar zxvf file.tgz

\*\* gzip –d file.gz

\*\* bzip2 –d file.bz2

c: output to standard out

d: decompress

gunzip -c -d openssl-0.9.8g.tar.gz | tar -xvf –

\* Compressing

\*\* tar -cvf archive.tar -C mydir

\*\* tar -cvf archive.tar file1 file2

\*\* tar -zcvf archive.tar.gz [compress dir or files]

z: for gzip

j: for bzip2

v - displays the name of each file being archived

f -  for using the files name for achieving

C - option is used to archive the directory using tar command.

**CYGWIN**

\* <http://web.gccaz.edu/~medgar/puttycyg/#releases>: replace Window console by Putty console

\*\* open putty.exe

\*\* in connection, type: choose Cygterm

\*\* in command, type: -

make sure that PATH has the string "C:\cygwin\bin" in it.

**Emacs**

**\* <http://www-h.eng.cam.ac.uk/help/tpl/emacs/node22.html>**

To save a macro into ~/.emacs so that you will always be able to use it, make ~/.emacs the current file and do

M-x insert-kbd-macro RET <macroname> RET

This will convert the macro into Lisp for you. Doing

M-x 4 insert-kbd-macro RET <macroname> RET

will save the relevent key bindings too.

\* Use macro

<http://www.emacswiki.org/cgi-bin/wiki/KeyboardMacrosTricks>

macro riffle is: C-x e C-x z z z z z z z (keep your finger on the z key)

<http://www.csm.uwe.ac.uk/~lrlang/java_html/emacsref.html>

\* Set a mark: ^ - spacebar

\* Delete: ^ - W

\* Copy: M - W

\* Paste: ^ - Y

\* Queried search and replace: M - %

|  |  |
| --- | --- |
| Spacebar | Replace text and find the next occurrence |
| Del | Leave text as is and find the next occurrence |
| . (period) | Replace text, then stop looking for occurrences |
| ! (exclamation point) | Replace all occurrences without asking |
| ^ (caret) | Return the cursor to previously replaced text |

\* Go to a specific line: M-x goto-line

\* Delete line: ^ - k

\* Begin line: ^ - a

\* End line: ^ - e

\* Top of file: M - <

\* Bottom of file: M - >

\* Split the emacs window into 2 parts: ^ - X 2

\* Unsplit the window: ^ - X 1

\* Switch to buffer in the other window: ^ - X O

\* List buffers: ^ - x ^ - b

\* Find files: C-x C-f

**Saving Arbitrary Keystrokes**

You can, with Emacs' macro facility, create, replay, name and save keyboard macros comprised of almost any arbitrary key sequence. You start the definition of a keyboard macro with "C-x(" and terminate it with "C-x)". After a macro is defined, you can do the following things:

* "C-x e" runs the command "call-last-kbd-macro" to execute it
* "M-x name-last-kbd-macro" to give your macro a name
* "insert-kbd-macro" inserts the Lisp code from a named macro into the current buffer
* "C-x C-w" runs the "write-file" command to write the file to disk
* "M-x load-file" loads Lisp code from disk and interprets it

**Repating Multiple Macros**

Once you create a macro, chances are pretty good you will want to run it more than once...that's why you created it to begin with. "C-u some-number your-command" will execute "your-command" exactly "some-number" times. This can be used to run the command "call-last-kbd-macro", or any command, for that matter, bunches o' times.

**CGI**

\* Configure Apache httpd.conf to run cgi script

\*\* AddHandler cgi-script .cgi

\*\* Options FollowSymLinks +ExecCGI

**LaTex**

latex caption.ins: to create sty file (work in window command line)

 Copy your new .cls or .sty file into this latex folder. (If you have a lot of style files, you may organize them in subfolders of this folder.)

 Open the Start menu and select "MikTex/MikTeX configuration". In the dialog box that now appears, go to the "General" tab and click the the "Refresh now" button under "File name database".

 You're done! One the file name database is refreshed, check to make sure that it works.

**VIM**

<http://www.apmaths.uwo.ca/~xli/vim/quickstart.html>

**Git**

**Steps to synchronize remote and local git repositories**

\* Initialize remote repository, in remoteDir

git init, git add ., git commit –a –m “Remote initial commit”

!!! if git is installed locally (on the server), please modify .bash\_profile to add

export GIT\_EXEC\_PATH=/home/l/luongmin/git/bin

export PATH=$PATH:$GIT\_EXEC\_PATH

\* Initialize local repository

git clone ssh://lmthang@cte.comp.nus.edu.sg/home/lmthang/remoteDir

(clone will create remote dir)

(git clone ssh://luongmin@tembusu2.comp.nus.edu.sg/home/l/luongmin/HYP)

!!!! If git is installed remotely by user (not root), there’s a need for some hack here

Create

In ~/.profile

# For login shells (interactive or not).

# For interactive non-login shells see ~/.bashrc.

# For non-interactive non-login ssh shells see ~/.bashrc.

# For other non-interactive non-login shells see $BASH\_ENV.

# At present this is sourced by ~/.bashrc for the ssh case, so

# all ssh sessions act like login sessions.

**export GIT\_EXEC\_PATH=/home/l/luongmin/git/bin**

**export PATH=$PATH:$GIT\_EXEC\_PATH**

# Source ~/.bashrc for interactive shells.

if [ -n "$PS1" -a -r ~/.bashrc ]; then source ~/.bashrc ; fi

... stuff for login shells (more environment mostly)

In ~/.bashrc

# For interactive non-login shells

# and non-interactive non-login ssh shells.

# For login shells (interactive or not) see first existent of

# ~/.bash\_profile, ~/.bash\_login and ~/.profile.

# For other non-interactive non-login shells see $BASH\_ENV.

# At present, this is sourced by ~/.profile so runs for all

# interactive shells (login or not).

# If the shell isn't actually interactive it is an ssh session, and

# we want to source /etc/profile and ~/.profile instead. We can't

# use a simple test of $PS1 and must test $- because /etc/bashrc

# (also sourced when bash detects ssh) sets $PS1. (We empty it

# here.)

if echo $- | grep -q i ; then : ; else

[ -r /etc/profile ] && source /etc/profile

PS1=

source ~/.profile

return

fi

Modify the bold part accordingly to where you set it

\* The common add and commit command is

git add .

git commit –a –m “msg”

\* After modify at local repository, to update to the remote repository:

\*\* First, replace the file **.git/hooks/post-update** at **remote repo** by the file here

<http://utsl.gen.nz/git/post-update>

then chmod 755 post-update

(this is to be done only one time)

\*\* git push origin

(origin represents [lmthang@cte.comp.nus.edu.sg](mailto:lmthang@cte.comp.nus.edu.sg), could check in .git/config)

\* To get update from remote:

git pull origin

\* If any problem with old version of git in CYGWIN, check this:

\*\* carriage-return/line-feed problem of Git under Windows/cygwin (<http://www.dont-panic.cc/capi/tag/git/> )

edit **.git/hooks/pre-commit** at **local repo** and comment out the following lines:

if (/\s$/) {  
 bad\_line("trailing whitespace", $\_);  
}

if (/^\s\* /) {

bad\_line("indent SP followed by a TAB", $\_);

}

**Other commands**

<http://git.or.cz/gitwiki/GitFaq#head-b96f48bc9c925074be9f95c0fce69bcece5f6e73>

\* git remote add bob /home/bob/myrepo

\* git pull /home/bob/myrepo master

\* <http://code.google.com/p/msysgit/> : git on Window

\* git show-branch –more=10 master

\* git ls-files: list files in the working directory

-d: deleted files

-s: stage files

--others: show other files

-m: modified files

## [“Project description file” error in git](http://stackoverflow.com/questions/200213/project-description-file-error-in-git)

http://stackoverflow.com/questions/200213/project-description-file-error-in-git

A colleague of mine experienced a similar issue here where push was not working. You could not push to a local or remote public repository. He was getting a project description file hasn't been set error thrown by .git/hooks/update. This error was not happening for the same project on a linux or Windows box, and seemed to be happening only on Windows Vista. From my research hooks/update is not by default executed, but in windows vista the file permissions meant that it was. Deletion of hooks/update resolved these issues.

-----

Deleting hooks/update is not what you want to do.

Just change .git/description to whatever you want - for instance "FOOBAR repository", or, if you are French, "Le depot de FOOBAR".

i got this error as well when pushing from macosx (git version 1.6.1, compiled with macports) to an ubuntu remote repository (git version 1.5.4.3)

-----

i've added the name of repository in the .git/description file on both local and remote repository and that fixed it

**Install rar**

[**http://www.cyberciti.biz/faq/open-rar-file-or-extract-rar-files-under-linux-or-unix/**](http://www.cyberciti.biz/faq/open-rar-file-or-extract-rar-files-under-linux-or-unix/)

**\* extract a line from a file using the line number: eg. extract line 4**

head -4 filename | tail -1

sed 4!d filename

awk 'NR==4{print $0}' filename

\* compile java, e.g. compile file Main.java

javac -d bin -extdirs lib -sourcepath src src/Main.java

lib: containing the jar lib files

bin: will receive .class outputs

src: contains the .java src files

Run:

java -classpath bin:lib/sphinx4.jar:lib/lucene-core-2.3.1.jar:lib/HUB4\_8gau\_13dCep\_16k\_40mel\_133Hz\_6855Hz.jar:lib/jsapi.jar Main

need to list out all jar file libs

\* Transfer files

sftp [sadm@luongmin-z.comp.nus.edu.sg](mailto:sadm@luongmin-z.comp.nus.edu.sg)

put local\_file

\* **sed**

sed -i 's/HelloWorld/UNIX/g' tmp/test1.txt: replace all occurrence of “HelloWorld” in file test1.txt by “UNIX”

http://drupal.org/node/128513

* Replacing *foo with foo\_bar* in a single file.

sed -i 's/foo/foo\_bar/g' somefile.module

* + -i = tell sed to edit the file(s)
  + s = substitute the following text
  + foo = what you want to substitute
  + foo\_bar = what you want to replace
  + g = global, match all occurrences in the line

 Replacing *foo with foo\_bar* in a multiple files.

sed -i 's/foo/foo\_bar/g'  \*.module

\* **grep, sed, find, xargs**

find ./tmp -type f | xargs grep -l 'HelloWorld' | xargs sed -i 's/HelloWorld/UNIX/g'

grep –l: print only filename

find –type f: find all ordinary files

xargs: generate args to input to another tool

The script find all files in tmp and replace all occurrence of “HellowWorld” by “UNIX”

\* [use](http://perldoc.perl.org/functions/use.html) Getopt::Long;

[my](http://perldoc.perl.org/functions/my.html) $data = "file.dat";

[my](http://perldoc.perl.org/functions/my.html) $length = 24;

[my](http://perldoc.perl.org/functions/my.html) $verbose;

$result = GetOptions ("length=i" => \$length, # numeric

"file=s" => \$data, # string

"verbose" => \$verbose); # flag

http://perldoc.perl.org/Getopt/Long.html

# Autotools Tutorial for Beginners

http://vindaci.members.sonic.net/cbreak/projects/autotools/index.php3

1. Create your sources. Start with [Makefile](http://vindaci.members.sonic.net/cbreak/projects/autotools/files/complete/Makefile), [Makefile.am](http://vindaci.members.sonic.net/cbreak/projects/autotools/files/complete/Makefile.am)
2. Run "autoscan" to generate "configure.scan".
3. Rename "configure.scan" to "configure.ac".
4. Run "autoheader" to generate "config.h.in".
5. Make your source portable by looking at "config.h.in". (We previously did this at a later step by reading "config.h", but we can do it in this step by referring to "config.h.in" instead.)
6. Run "automake".
7. Fix errors and run "automake" again.

Add AM\_INIT\_AUTOMAKE to “configure.ac”

Run "aclocal".

`automake –add-missing –copy

create missing files if still errors

1. Run "autoconf".
2. Configure, make, and run!
3. Create a file named "Makefile.am".
4. "Makefile.am" will be used by Automake to create a new "Makefile.in".
5. "Makefile.in" will be used by "configure" to create "Makefile".

If you modify your source...

1) Run `autoscan` again

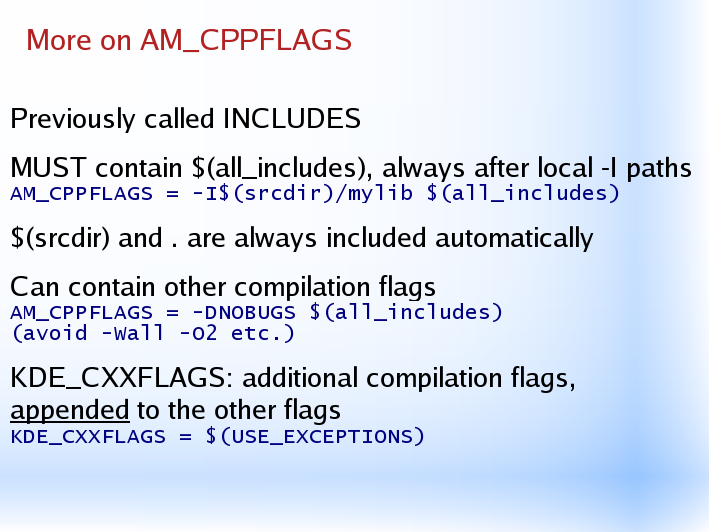
2) Compare configure.scan with configure.ac

● Update configure.ac

3) Run `autoreconf`

# David Faure's talks at Nove Hrady

<http://www.kdab.net/~dfaure/n7y/> How to write a Makefile.am: [HTML version](http://www.kdab.net/%7Edfaure/n7y/Makefile.am/index.html)



*Makefile.am example*

bin\_PROGRAMS = processPhraseTable processLexicalTable queryLexicalTable

processPhraseTable\_SOURCES = GenerateTuples.cpp processPhraseTable.cpp

processLexicalTable\_SOURCES = processLexicalTable.cpp

queryLexicalTable\_SOURCES = queryLexicalTable.cpp

AM\_CPPFLAGS = -W -Wall -ffor-scope -D\_FILE\_OFFSET\_BITS=64 -D\_LARGE\_FILES -I$(top\_srcdir)/moses/src

processPhraseTable\_LDADD = -L$(top\_srcdir)/moses/src -lmoses

processPhraseTable\_DEPENDENCIES = $(top\_srcdir)/moses/src/libmoses.a

processLexicalTable\_LDADD = -L$(top\_srcdir)/moses/src -lmoses

processLexicalTable\_DEPENDENCIES = $(top\_srcdir)/moses/src/libmoses.a

queryLexicalTable\_LDADD = -L$(top\_srcdir)/moses/src -lmoses

queryLexicalTable\_DEPENDENCIES = $(top\_srcdir)/moses/src/libmoses.a

---------------------

bin\_PROGRAMS = moses

moses\_SOURCES = Main.cpp mbr.cpp IOStream.cpp TranslationAnalysis.cpp

AM\_CPPFLAGS = -W -Wall -ffor-scope -D\_FILE\_OFFSET\_BITS=64 -D\_LARGE\_FILES -DUSE\_H

YPO\_POOL -I$(top\_srcdir)/moses/src

moses\_LDADD = -L$(top\_srcdir)/moses/src -lmoses

moses\_DEPENDENCIES = $(top\_srcdir)/moses/src/libmoses.a

-------------

lib\_LIBRARIES = libmoses.a

AM\_CPPFLAGS = -W -Wall -ffor-scope -D\_FILE\_OFFSET\_BITS=64 -D\_LARGE\_FILES

libmoses\_a\_SOURCES = \

ConfusionNet.cpp \

XmlOption.cpp

if SRI\_LM

libmoses\_a\_SOURCES += LanguageModelSRI.cpp

endif

if IRST\_LM

libmoses\_a\_SOURCES += LanguageModelIRST.cpp

endif

if INTERNAL\_LM

libmoses\_a\_SOURCES += LanguageModelInternal.cpp \

NGramCollection.cpp \

NGramNode.cpp

endif

**perl -pi -e 's/Windoze/UsefulNIX/g' filename**

**find A -type f | xargs grep -l html | xargs perl -pi -e 's/Windoze/UsefulNIX/g'**

**ls | xargs –I xxx chmod 755 xxx : apply for each file listed chmod 755**

date '+%Y%m%d-%H%M': print date acroding to the format

date -> Sat Sep 13 10:38:51 SGT 2008

date '+%Y%m%d-%H%M' -> 20080913-1039

* Get the number of lines of a file

wc -l fileName | grep -oP "^\d+"

wc -l fileName | awk '{ print $1 }'

wc -l < fileName

* Count the number of lines a file has, then for the number-of-line times

N=`wc -l $1 | awk '{ print $1 }'`

echo "$N"

for i in `seq 1 $N`;

do

echo "item: $i"

done

find . –type f | xargs –I xxx echo “\”xxx\”” | xargs grep pattern : escape the filename with “” before grep since several files might have space in their names

\* Create temporary file in shell

**$$** which is the *process id*

temp\_file="/tmp/"`date '+%Y%m%d-%H%M'`"-"$$".tmp"

\* discard output || redirect to a tmp file that we won’t need

uses /tmp/null

./runProgram.sh 2>/tmp/null: discard all stderr print out

\* Determine if a file exists

if [ -f testfile ]  
then  
echo testfile exists!  
fi

* Add public key to remote server so that we don’t need to type password when ssh or scp ..

<http://www.hostingrails.com/forums/wiki_thread/27>

\* check if we have the file ~/.ssh/id\_dsa.pub

otherwise generate using

ssh-keygen -t dsa

(just keep entering)

\* next, copy the id to the remote server using

ssh-copy-id -i ~/.ssh/id\_dsa.pub user@'servername'

(for CYGWIN, copyt the the script here to file ssh-copy-id <http://www.experts-exchange.com/Software/CYGWIN/Q_23768288.html>

Run ./ssh-copy-id username@domain.com

)

After setting up this authentication, we could do other commands without the need for authentication, e.g:

ssh lmthang@cte.comp.nus.edu.sg mkdir /home/lmthang/HYP/result/acl05

* handle decimal in bash

LINE=`wc -l < acl05\_filtered-$corpus.en`

echo "$LINE"

SIZE=`echo "$LINE\*2.5/100" | bc`

echo "$SIZE"

SVN

1119 mkdir -p svn/HYP

1120 svnadmin create svn/HYP

1121 svn import HYP/scripts file:///home/l/luongmin/svn/HYP -m "Init repo"

1122 svn import HYP/src file:///home/l/luongmin/svn/HYP -m "Init repo"

svn list svn+ssh://host.example.com/repos/project

**sort**

grep ' n/SUF' $RS/acl05\_full-m/result/model/lex.0-0.n2f | sort -nrk 3 | head

-n numeric sort

-r reverse the result of comparison

-k to specify the key position to sort

cat $RS/acl05\_40K-m/result/model/lex.0-0.n2f | sort -k 2 > tmp.txt

grep ' n/SUF' $RS/acl05\_full-m/result/model/lex.0-0.n2f | sort -nrk 3 | head -10 | awk '{print $1}'

bash integer comparison

<http://www.faqs.org/docs/abs/HTML/comparison-ops.html>

-eq

is equal to

**if [ "$a" -eq "$b" ]**

-ne

is not equal to

**if [ "$a" -ne "$b" ]**

-gt

is greater than

**if ["$a" -gt "$b" ]**

-ge

is greater than or equal to

**if [ "$a" -ge "$b" ]**

-lt

is less than

**if [ "$a" -lt "$b" ]**

-le

is less than or equal to

**if [ "$a" -le "$b" ]**

bash: file test

-e

file exists

<http://tldp.org/LDP/abs/html/fto.html>

* multi-line comments

#!/bin/bash

echo "Say Something"

<<COMMENT1

your comment 1

comment 2

blah

COMMENT1

echo "Do something else"

* bash string matching

if [[ "$FROMINFO" =~ "morpheme0" ]]

then

echo "match"

else

echo "not match"

fi

* bash multiple conditions
* if [ $count -gt 0 ] && [ $somevar != $var ]; then
* ...do something
* fi

if [[ $count -gt 0 && $someVar != $var ]]  
then  
  
...  
  
fi

* LC\_ALL=C sort -T ./ $outPhraseTable.unsort > $outPhraseTable.sort

**Installing libxslt**

http://jamesclarke.info/notes/libxml2/

By default configure will build against the default libxml included with OS X. And will typically give the error:

checking for libxml libraries >= 2.6.8... configure: error:

Version 2.6.7 found. You need at least libxml2 2.6.8 for this

version of libxslt

This can be corrected by telling configure to use the libxml2 use the new libxml2.

$ ./configure

--with-python=/System/Library/Frameworks/Python.framework/Versions/2.3/

--prefix=/usr/local

--with-libxml-prefix=/usr/local

--with-libxml-include-prefix=/usr/local/include

--with-libxml-libs-prefix=/usr/local/lib

$ make

$ sudo make install

./configure --prefix=/home/lmthang/usr --with-libxml-prefix=/home/lmthang/usr --with-libxml-include-prefix=/home/lmthang/usr/include --with-libxml-libs-prefix=/home/lmthang/usr/lib

Installing Tomcat with distribution to different directory

* You need to download several dependent packages. First, add the following line into build.properties.default

base.path=/research/wing/luongmin/nutch/archive/tomcat-download

Then, run

ant download

* Run

ant

* Run

ant –f extras.xml

* Before running ant distribution, add the following line into build.properties.default

tomcat.dist=/research/wing/luongmin/nutch/tomcat

* Run

ant –f dist.xml

* If you encounter any problem with Window installer, remove “installer” in depends specified in dist.xml <target name="release" depends=" (just a tmp work-around, as I’m not expert with ant)