

# Shells

tsaimh (2022, CC BY-SA)

lctseng (2019-2021, CC BY-SA)

? (1996-2018)

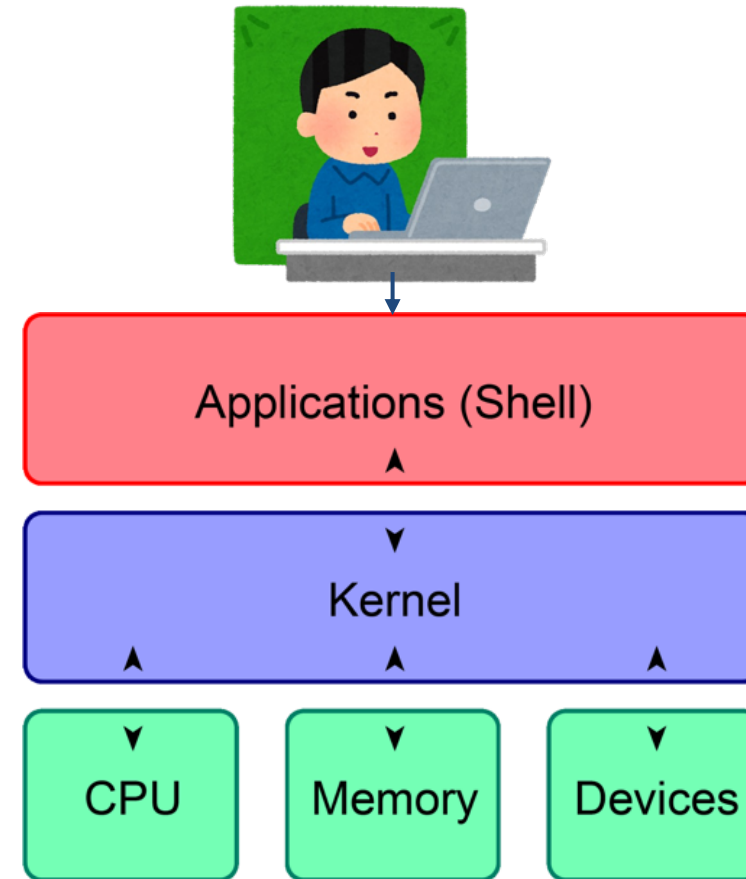
國立陽明交通大學資工系資訊中心

Computer Center, Department of Computer Science, NYCU

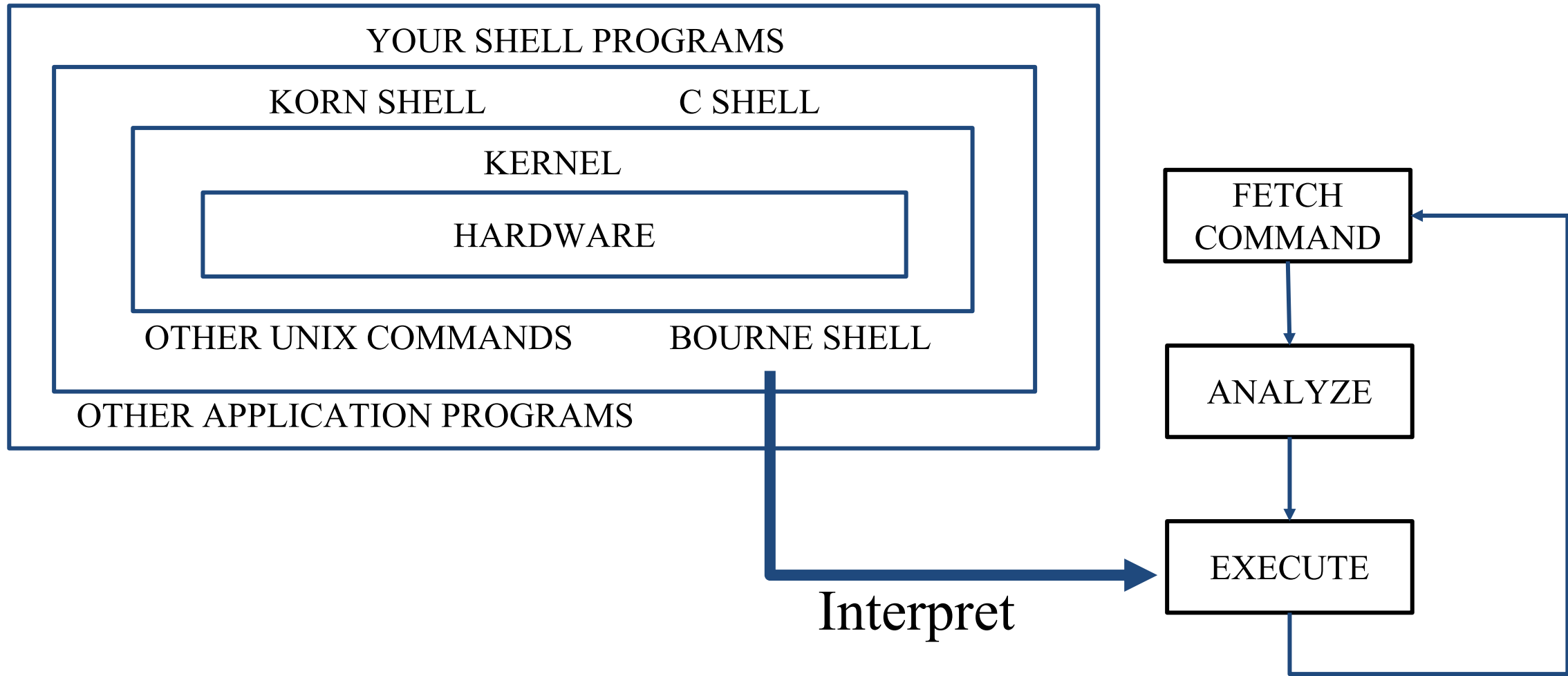
# UNIX Kernel and Shell

- Interface to communicate with kernel
- Where you type commands

```
[Meng-Hsun Tsai@pts/37(alumni)][~] > date
Tue Aug 16 08:55:19 CST 2022
[Meng-Hsun Tsai@pts/37(alumni)][~] > pwd
/net/dcs/93/9317807
[Meng-Hsun Tsai@pts/37(alumni)][~] > whoami
tsaimh
[Meng-Hsun Tsai@pts/37(alumni)][~] > |
```



# UNIX Kernel and Shell (2)



# The UNIX Shells

Shell	Originator	System Name	Prompt
Bourne Shell <b>(In FreeBSD base)</b>	S. R. Bourne	/bin/sh	\$
C Shell <b>(In FreeBSD base, Default for root)</b>	Bill Joy	/bin/csh	%
TENEX C Shell <b>(In FreeBSD base)</b>	Ken Greer	/bin/tcsh	>
Korn Shell	David Korn	(shells/ksh93)	\$
Bourne-Again Shell <b>(Widely used)</b>	Brian J. Fox	(shells/bash)	\$
Z Shell <b>(macOS default)</b>	Paul Falstad	(shells/zsh)	%

# Windows Shell

- cmd.exe
  - First released in 1987
  - For Windows NT/Windows CE
  - Still used in modern Windows
- PowerShell
  - First released in 2006
  - To provide the same functionality as UNIX shells
  - Also has [Linux](#)/[MacOS](#) releases

# Shell Startup Files

sh	/etc/profile	login shell, system wide
	~/.profile	login shell
	ENV	
csh	/etc/csh.cshrc	always, system wide
	/etc/csh.login	login shell, system wide
	~/.cshrc	always
	~/.login	login shell
	~/.logout	logout shell
	/etc/csh.logout	logout shell, system wide

# Shell Startup Files (2)

tcsh	~/.tcshrc	login shell
	(csh startup files)	backward compatibility for csh
bash	/etc/profile → ~/.bash_profile → ~/.bash_login → ~/.profile	login shell
	~/.bashrc	login shell
	BASH_ENV	

*Bash Startup Files : [https://www.gnu.org/software/bash/manual/html\\_node/Bash-Startup-Files.html](https://www.gnu.org/software/bash/manual/html_node/Bash-Startup-Files.html)*

# Shell Startup Files (3)

- A sample tcshrc for you to change your prompt
- Simplest install steps
  - Take a look at the content before running it

```
$ fetch https://nasa.cs.nctu.edu.tw/sa/sample/.tcshrc.color -o  
~/.tcshrc  
$ source ~/.tcshrc
```

```
tsaimh@bsd1:~ % fetch https://nasa.cs.nctu.edu.tw/sa/sample/.tcshrc.color -o ~/.  
tcshrc  
/home/tsaimh/.tcshrc 861 B 8477 kBps 00s  
tsaimh@bsd1:~ % source ~/.tcshrc  
[tsaimh@bsd1 ~ ]
```



# Shell Environment Variables (1)

- Controlling shell behaviors
  - There are many environment variables that control the shell behavior
- To dump them:

```
$ env
```

- To get value:

```
$ echo $VARIABLE_NAME  
$ echo ${VARIABLE_NAME}  
$ echo "$PATH"
```

# Shell Environment Variables (2)

- Useful Environment Variables

Variables	Description
HOME	User's home directory
MAIL	User's mailbox
PATH	Command search path

# Variables and Strings Quotes

Char.		Purpose
sh	var=value	Assign value to variable
csh	set var=value	
\$var, \${var}		Get shell variable
`cmd`		Substitution stdout
'string'		Quote character without substitution
"string"		Quote character with substitution

# Variables and Strings Quotes (2)

Shell	sh	Csh
Commands	<pre>\$ varname=`/bin/date` \$ echo \$varname \$ echo 'Now is \$varname' \$ echo "Now is \$varname"</pre>	<pre>\$ set varname2=`/bin/date` \$ echo \$varname2 \$ echo 'Now is varname2' \$ echo "Now is \$varname2"</pre>
Result	<pre>Mon Aug 15 14:22:19 CST 2022 Now is \$varname Now is Mon Aug 15 14:22:19 CST 2022</pre>	

# Global Variables

- Use "env" command to display global variables
- Assignment

	Bourne Shell	C Shell
Local variable	<pre>my=test current_month=`date +%m`</pre>	<pre>set my=test set current_month=`date +%m`</pre>
Global variable	<pre>export my=test export EDITOR=/usr/bin/ee</pre>	<pre>setenv my test setenv EDITOR /usr/bin/ee</pre>

# Shell Special Characters

- Reduce typing as much as possible

	Characters	Description
sh	*	Match any string of characters
	?	Match any single alphanumeric character
	[...]	Match any single character within []
	[!...]	Match any single character not in []
	~	Home directory

# Shell Special Characters (2)

- Example: There are some files in current directory
  - test1, test2, test3, test4, test-5, testmess

	Command	Result
sh	\$ ls test*	test1 test2 test3 test4 test-5 testmess
	\$ ls test?	test1 test2 test3 test4
	\$ ls test[123]	test1 test2 test3
	\$ ls test[!345]*	test1 test2 test-5 testmess
	\$ ls ~	List files under your home

# Shell Special Characters (3)

Char.	Purpose	Example
#	Start a shell comment	# this is a comment
;	Command separator	\$ ls test*; ls test?
&&	Executes the first command, and then executes the second if first command success (exit code=0)	\$ cd foo/bar && make install
	Executes the first command, and then executes the second if first command fail (exit code≠0)	\$ cp x y    touch y



# Shell Special Characters (4)

Char.	Purpose	Example
\	(1)Escape character (2)Command continuation indicator	<pre>\$ touch test\*; ls test\ test* \$ ls \ &gt; test*</pre>
&	Background execution	<pre>\$ make buildworld &amp; \$ sleep 5 &amp;</pre>

# Common Built-in Commands

SH	CSH	Description
	set/unset	Set/Unset shell options and positional parameters
<i>(empty)</i> /unset	set/unset	Set/Unset a local variable
export	setenv/unsetenv	Set/Unset a global variable
	set	Display shell variables (sh: local + global, csh: local)
	env	Display global (environment) variables
<i>(N/A)</i>	login, logout	Login / Logout
	exit	exit shell

# Common Built-in Commands (2)

SH	CSH	Description
(N/A)	dirs	print directory stack
(N/A)	popd, pushd	Pop/push directory stack
	echo	write arguments on stdout
	alias/unalias	command aliases
	fg, bg	Bring a process to foreground/background (e.g. sleep 5 &)
	jobs	List active jobs (with job numbers)
	%[job no.]	Bring a process to foreground (e.g. %1)

# Built-in Shell Commands (3)

SH	CSH	Description
	kill	Send a signal to a job ( <b>kill %job</b> or <b>kill pid</b> )
(N/A)	stop	Suspend a background process (%job   pid)
	exec	execute arguments
	nice	Change nice value

# Built-in Shell Commands (4)

SH	CSH	Description
(N/A)	history	Display history list
(N/A)	rehash	Evaluate the internal hash table of the contents of directories
(N/A)	source	Read and execute a file

## References:

- <https://it.cs.nycu.edu.tw/unix-basic-commands>
- [http://www.unix.org.ua/oreilly/unix/unixnut/ch04\\_06.htm](http://www.unix.org.ua/oreilly/unix/unixnut/ch04_06.htm)
- [http://publib.boulder.ibm.com/infocenter/pseries/index.jsp?topic=/com.ibm.aix.doc/aixuser/usrosdev/list\\_c\\_builtin\\_cmds.htm](http://publib.boulder.ibm.com/infocenter/pseries/index.jsp?topic=/com.ibm.aix.doc/aixuser/usrosdev/list_c_builtin_cmds.htm)
- <https://www.freebsd.org/cgi/man.cgi?query=tcsch>
- <https://www.freebsd.org/cgi/man.cgi?query=sh>

# Input/Output Redirection

- There are 3 default file descriptors

Integer value	Name
0	stdin (Standard Input)
1	stdout (Standard Output)
2	stderr (Standard Error)

- Using man command to read more information
  - [sh\(1\)](#): Redirection
  - [tcsch\(1\)](#): Input/Output

# Input/Output Redirection (2)

Method	Name
<code>cmd &lt; file</code>	Open the file as stdin of cmd
<code>cmd &gt; file</code>	Write stdout of cmd in the following file. Truncates existing files. (tcsh: use "set noclobber" to avoid overwriting)
<code>cmd &gt;&gt; file</code>	Append stdout of cmd to the following file
<code>2&gt;&amp;1</code>	Merge stdout with stderr
<code>cmd1   cmd2</code>	Pipe stdout of cmd1 into stdin of cmd2

# File and Directory Related Commands

Command	Purpose
<code>ls</code>	List a directory's content
<code>pwd</code>	Print working directory
<code>cd</code>	Change to other directory
<code>mkdir</code>	Make(create) a new directory
<code>rmdir</code>	Remove existing empty directory
<code>cat</code>	Concatenate file
<code>cp</code>	Copy file



# File and Directory Related Commands (2)

Command	Purpose
<code>ln</code>	Link files
<code>mv</code>	Move file
<code>rm</code>	Remove file
<code>stat</code>	Display file status

# Select and File Processing Related Commands

Command	Purpose
head	Display first lines of a file
tail	Select trailing lines
grep	Select lines
diff	Compare and select difference in two files
wc	Count characters, words or lines of a file
uniq	Select uniq lines
cut	Select columns

# Select and File Processing Related Commands (2)

Command	Purpose
<code>sort</code>	Sort and merge multiple files together
<code>sed</code>	Edit streams of data
<code>awk</code>	Pattern scanning and processing language

# Select and File Processing Related Commands (3) - Example Usage

- Look first few lines or last few lines
  - `$ head /var/log/message`
  - `$ tail /var/log/message`
    - `-n` : specific how many lines
- Find the occurrence of certain pattern in file
  - `$ grep -l tsaimh *`
    - Print the **filename** that has “tsaimh” as content
  - `$ grep -n tsaimh /etc/passwd`
    - Print the **line number** when using grep

# Select and File Processing Related Commands (4) - Example Usage

- List tsaimh's id, uid, home, shell in /etc/passwd
  - `$ grep tsaimh /etc/passwd | cut -f1,3,6,7 -d:`
    - `-f1,3,6,7` : fetch 1st ,3rd ,6th ,7th column
    - `-d` : separation symbol

```
tsaimh:*:1001:20:Meng-Hsun Tsai:/home/tsaimh:/bin/tcsh
```

```
$ grep tsaimh /etc/passwd | cut -f1,3,6,7 -d:  
tsaimh:1001:/home/tsaimh:/bin/tcsh
```

# Select and File Processing Related Commands (5) - Example Usage

- Cut out file permission and file name from ls output
  - `$ ls -l | grep -v ^total | cut -c 1-11,47-`
    - `-c1-12` : 1st~12th characters (start from 1, instead of 0)
    - `-c47-` : characters after 47th character (include 47th )

```
$ ls -l
total 2312
-rw-r--r--  1 tsaimh  ta  875394 Aug 14 13:37 00_Syllabus.pdf
-rw-r--r--  1 tsaimh  ta  841270 Aug 12 15:59 01_Install_FreeBSD.pdf
-rw-r--r--  1 tsaimh  ta  457582 Aug 12 15:59 02_Installing_Applications.pdf
$ ls -l | grep -v ^total | cut -c 1-11,47-
-rw-r--r--  00_Syllabus.pdf
-rw-r--r--  01_Install_FreeBSD.pdf
-rw-r--r--  02_Installing_Applications.pdf
```

# Select and File Processing Related Commands (6) - Example Usage

- Use awk to generate the same behavior of cut
  - `$ ls -l | grep -v ^total | awk '{print $1 " " $9}'`
    - Result is same as [P.30](#)

```
$ ls -l
total 2312
-rw-r--r--  1 tsaimh  ta  875394 Aug 14 13:37 00_Syllabus.pdf
-rw-r--r--  1 tsaimh  ta  841270 Aug 12 15:59 01_Install_FreeBSD.pdf
-rw-r--r--  1 tsaimh  ta  457582 Aug 12 15:59 02_Installing_Applications.pdf
$ ls -l | grep -v ^total | awk '{print $1 " " $9}'
-rw-r--r-- 00_Syllabus.pdf
-rw-r--r-- 01_Install_FreeBSD.pdf
-rw-r--r-- 02_Installing_Applications.pdf
```

# Select and File Processing Related Commands (7) - Example Usage

- Use awk to generate the same behavior of cut
  - \$ awk -F: '{print \$1 " " \$6}' /etc/passwd
    - -F : separation symbol

```
tsaimh:*:1001:20:Meng-Hsun Tsai:/home/tsaimh:/bin/tcsh
```

```
$ awk -F: '{print $1 " " $6}' /etc/passwd  
tsaimh /home/tsaimh
```



# Select and File Processing Related Commands (8) - Example Usage

- Options of "sort" command
  - -r : reverse
  - -u : unique keys
  - -n : numeric keys sorting
    - Default: string sorting, 14 > 123
  - -k : specific columns to sort with
  - -t : field separator

# Select and File Processing Related Commands (9) - Example Usage

- List directory contents and sort by file size decreasingly
  - `$ ls -al | sort -n -k 5,5 -r`
    - `-k` : specific columns to sort with
    - `-r` : reverse

```
% ls -l | sort -n -k 5,5 -r
-rw-r--r--  1 tsaimh  ta  875394 Aug 14 13:37 00_Syllabus.pdf
-rw-r--r--  1 tsaimh  ta  841270 Aug 12 15:59 01_Install_FreeBSD.pdf
-rw-r--r--  1 tsaimh  ta  457582 Aug 12 15:59 02_Installing_Applications.pdf
```

# Select and File Processing Related Commands (10) - Example Usage

- Sort contents of /etc/passwd by username and remove annotations
  - `$ sort -t: -k 1,1 /etc/passwd | grep -v ^#`
    - `-t` : field separator
    - `-k` : specific columns to sort with

```
games:*:7:13:Games pseudo-user:/usr/games:/usr/sbin/nologin
git_daemon:*:964:964:git daemon:/nonexistent:/usr/sbin/nologin
hast:*:845:845:HAST unprivileged user:/var/empty:/usr/sbin/nologin
kmem:*:5:65533:KMem Sandbox:/:/usr/sbin/nologin
tsaimh:*:1001:20:Meng-Hsun Tsai:/home/tsaimh:/bin/tcsh
```

# Select and File Processing Related Commands (11) - Example Usage

- List records in /etc/hosts sorted by IPv4 address

```
$ sort -t. -n -k 1,1 -k 2,2 -k 3,3 -k 4,4 '/etc/hosts' | grep -v ^#
```

- -n : numeric keys sorting

- Before sorting

```
# In the presence of the domain name service or NIS, this file may
# not be consulted at all; see /etc/nsswitch.conf for the
# resolution order.
#
::1          localhost localhost.my.domain
127.0.0.1    localhost localhost.my.domain
140.113.17.26 nctucs.tw
64.233.187.95 www.googleapis.com googleapis.l.google.com
```

# Select and File Processing Related Commands (12) - Example Usage

- List records in /etc/hosts sorted by IPv4 address

```
$ sort -t. -n -k 1,1 -k 2,2 -k 3,3 -k 4,4 '/etc/hosts' | grep -v ^#
```

- -n : numeric keys sorting

- After sorting

```
:::1 localhost localhost.my.domain
64.233.187.95 www.googleapis.com googleapis.l.google.com
127.0.0.1 localhost localhost.my.domain
140.113.17.26 nctucs.tw
```

# Select and File Processing Related Commands (13) - Example Usage

- Translate characters

- `$ echo "Hello World" | tr "a-z" "A-Z"`
  - Change all alphabet to uppercase

```
$ echo "Hello World" | tr "a-z" "A-Z"  
HELLO WORLD
```

- `$ tr -d "\t" < file1`
  - Delete TAB in file1
- `$ tr -s " " " " < file1`
  - Delete multiple space in file1

# Select and File Processing Related Commands (14) - Example Usage

- Translate characters
  - `$ grep tsaimh /etc/passwd | tr ":" "\n"`
    - Change all ":" to "\n"

```
$ grep tsaimh /etc/passwd | tr ":" "\n"
tsaimh
*
1001
20
Meng-Hsun Tsai
/home/tsaimh
/bin/tcsh
```

# xargs Command

- xargs – construct argument list(s) and execute utility
  - -n number
  - -I replstr (every)
  - -J replstr (first only)
  - -s size
  - ...



# xargs Command (2)

```
% ls
2.sh      3.csh      4.csh      4.sh      bsd1.ping
testin
% ls | xargs echo
2.sh 3.csh 4.csh 4.sh bsd1.ping testin
% ls | xargs -n1 echo
2.sh
3.csh
4.csh
4.sh
bsd1.ping
testin
```

# xargs Command (3)

```
% ls | xargs -I % -n1 echo % here %  
2.sh here 2.sh  
3.csh here 3.csh  
4.csh here 4.csh  
4.sh here 4.sh  
bsd1.ping here bsd1.ping  
testin here testin
```

# xargs Command (4)

```
% ls | xargs -J % -n1 echo % here %  
2.sh here %  
3.csh here %  
4.csh here %  
4.sh here %  
bsd1.ping here %  
testin here %
```

# xargs Command (5)

- Example : ping all hosts in file

```
$ cat host
www.google.com
bsd1.cs.nctu.edu.tw
linux3.cs.nctu.edu.tw
cs.nctu.edu.tw

$ cat host | xargs -n1 ping -c 1 | grep "bytes from"
64 bytes from 64.233.188.103: icmp_seq=0 ttl=47 time=6.944 ms
64 bytes from 140.113.235.135: icmp_seq=0 ttl=57 time=1.451 ms
64 bytes from 140.113.235.153: icmp_seq=0 ttl=57 time=1.612 ms
64 bytes from 140.113.235.47: icmp_seq=0 ttl=57 time=1.856 ms
```

# The Unix Philosophy

- [https://en.wikipedia.org/wiki/Unix\\_philosophy](https://en.wikipedia.org/wiki/Unix_philosophy)
- Lots of little tools, each good at one thing
  - Use them together to achieve your goal
- Try other shells (install from package/ports)
  - zsh
    - Oh-my-zsh: <https://github.com/robbyrussell/oh-my-zsh>
  - fish

# ShellCheck

- Finds bugs in your shell scripts
- <https://www.shellcheck.net/>
- devel/hs-ShellCheck
- `pkg install hs-ShellCheck`

# Appendix

## Command History in (t)csh

國立陽明交通大學資工系資訊中心

Computer Center, Department of Computer Science, NYCU

# Command History in (t)csch

Commands	Description
!n	exec previous command line n (see history)
!-n	exec current command line minus n
!!	exec last command (the same as !-1)
!str	exec previous command line beginning with str
!?str	exec previous command line containing str

```
% history
10  8:31  cp ypwhich.1 ypwhich.1.old
11  8:31  vi ypwhich.1
12  8:32  diff ypwhich.1.old ypwhich.1
13  8:32  history
% !?old
```



# Command History in (t)csch (2)

Commands	Description
!!:n	use the nth word of previous comm
!!:m-n	select words m ~ n of previous command
!!:*	use all arguments of previous command
!!:s/str1/str2/	substitute str1 with str2 in previous command

```
% history
15  8:35  cd /etc
16  8:35  ls HOSTS FSTAB
17  8:35  history
% cat !-2:*:s/HOSTS/hosts/:s/FSTAB/fstab → cat hosts fstab
```

- [tcsch\(1\)](#): History Substitution