

# Some tips

bananaapple

# Who am I?

- ID : bananaapple
- 學校科系：交通大學資工系
- 年級：大四
- 目前為 Bamboofox 中的一員
- 學習資安約一年



# Outline

- 作業系統選擇
- Virtual Box VMware Player
- 電腦架構
- RAM
- OS
- i386 and amd64
- Vim
- Tmux
- Fast tips

# 作業系統選擇

- 沒有最好的作業系統
- 選擇自己最熟悉、用起來最順手的
- 個人偏愛使用 Windows 搭配虛擬機 Linux
- Windows 小算盤 ( 程式設計師模式 )
- Putty 或 Cygwin
- 虛擬機請務必安裝 Guest Addition
- 才可以方便地使用共用剪貼簿和複製貼上

# 作業系統選擇

The screenshot shows a web browser window displaying a CTF problem page for "[Reverse] flag-robot [50]". The page includes a description, problem source (XCTF-2015), and a hint. A terminal window is overlaid on the right, showing a netcat listener on 140.113.194.85:49161 that has connected to an IP address. The terminal output shows a robot loading and waiting for a flag. The hint states the flag is 96 characters long and should be in BAMBOOF[RE]... format. A table at the bottom lists users and their scores for this problem.

**[Reverse] flag-robot [50]**

**Description**  
Get the flag! <http://140.113.194.85:3000/data/flag-robot>

**Problem Source**  
XCTF-2015

**Hint**  
The flag is 96 characters, you should put it in BAMBOOF[RE]...

**User** | **Problem** | **Score**

User	Problem	Score
Lays	[Reverse] flag-robot	50
mrhoap	[Reverse] flag-robot	50
Betty	[Reverse] flag-robot	50

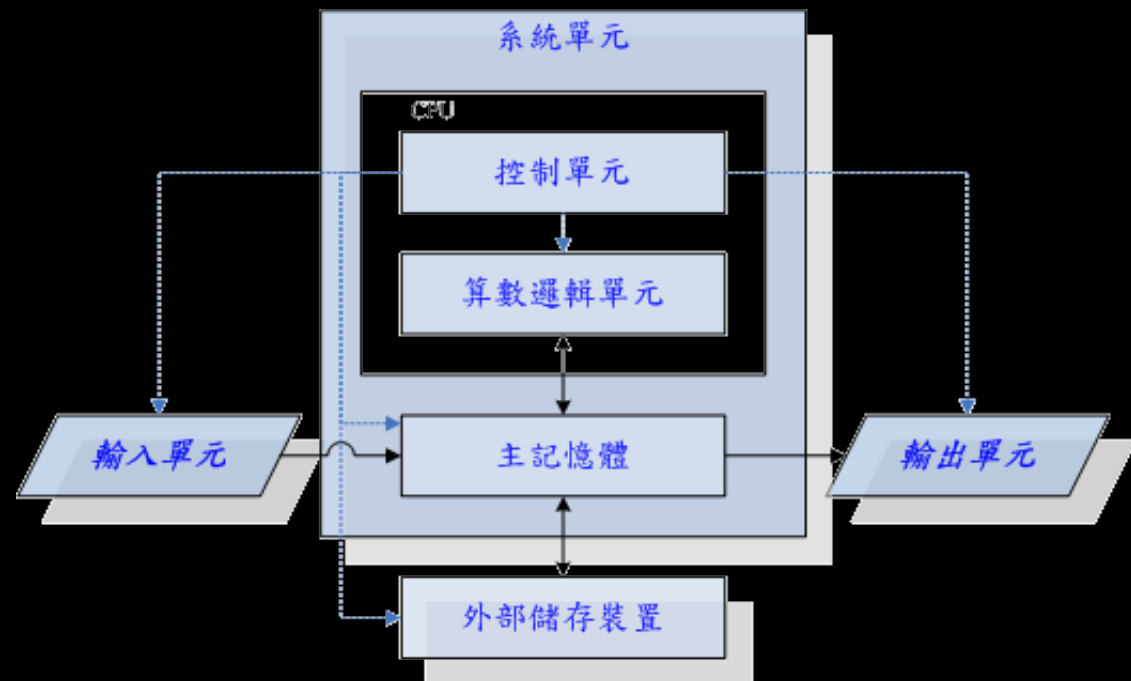
Copyright ©2015 SIGLAB/CSNSL, Inc.

# Virtual Box VMware Player

- Vmware Player
- 支援Windows、Linux
- 有方便的自動安裝(設定好帳號和密碼後幫你灌系統，目前有Ubuntu Kali還沒有)
- 使用 Ctrl+Alt 來切換
- 個人推薦使用 Vmware
- Virtual Box
- 支援Windows、Linux、OS X
- Virtual Box 5.0出來後
- 把原本 Vmware 有的 drag and paste.....都做出來了
- 基本上和現在的Vmware沒什麼差異
- 使用Ctrl來切換

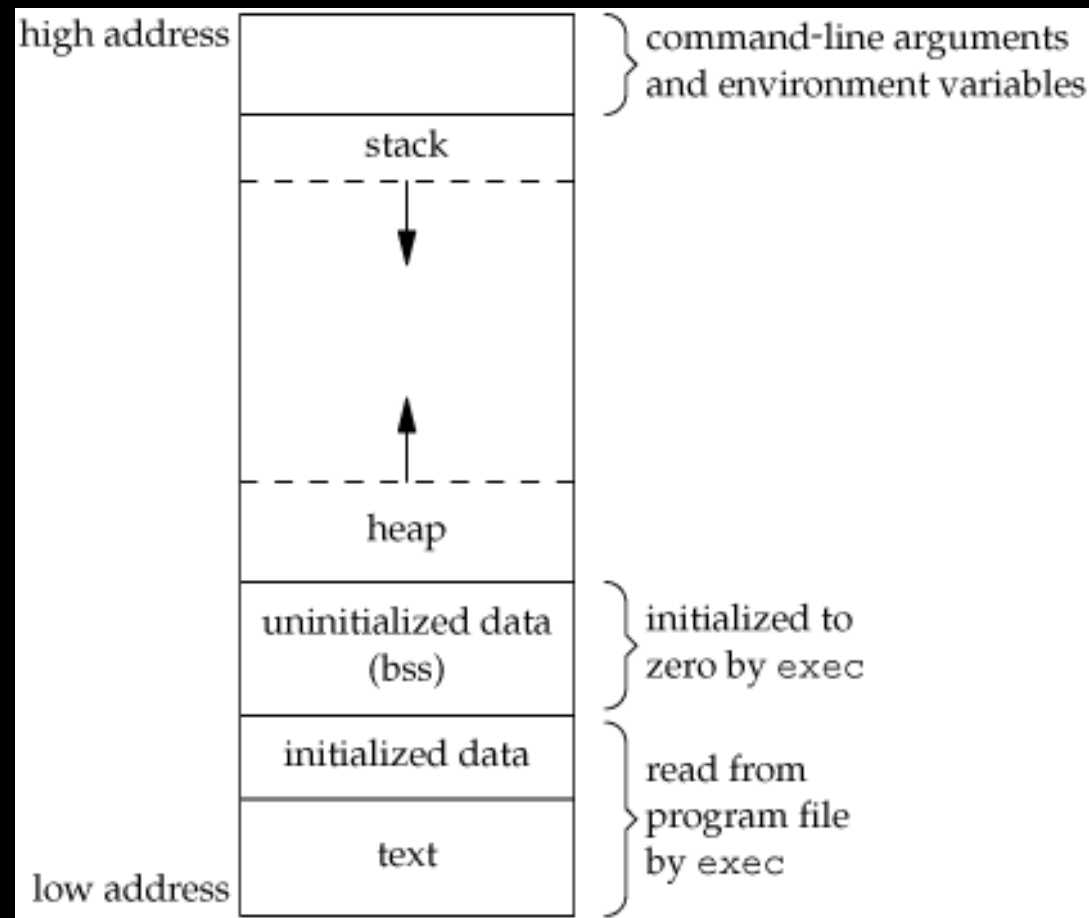
# 電腦架構

- 先來講講電腦架構吧
- 輸入單元：滑鼠、鍵盤.....任何能讓你傳送訊息的元件
- 輸出單元：螢幕、音響、印表機.....
- 主記憶體就是我們常說的RAM
- CPU就是整個電腦的核心
- 所有的運算都在CPU進行
- 外部儲存裝置：硬碟、隨身碟



# RAM

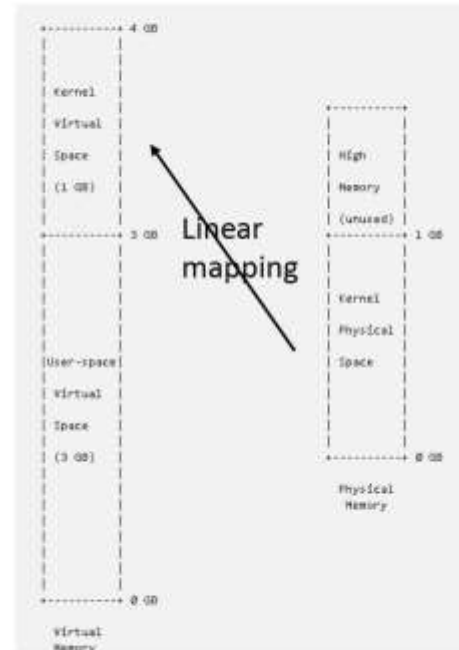
- 現在我們能看到的 memory address 都稱作 Virtual Address
- 可以達到 memory 隔離
- 避免存取到其他process的 memory
- 使用 Paging 來實作
- 在硬體實作上使用 MMU ( Memory Management Unit ) 將 Virtual Address 轉換為 Physical Address





# RAM

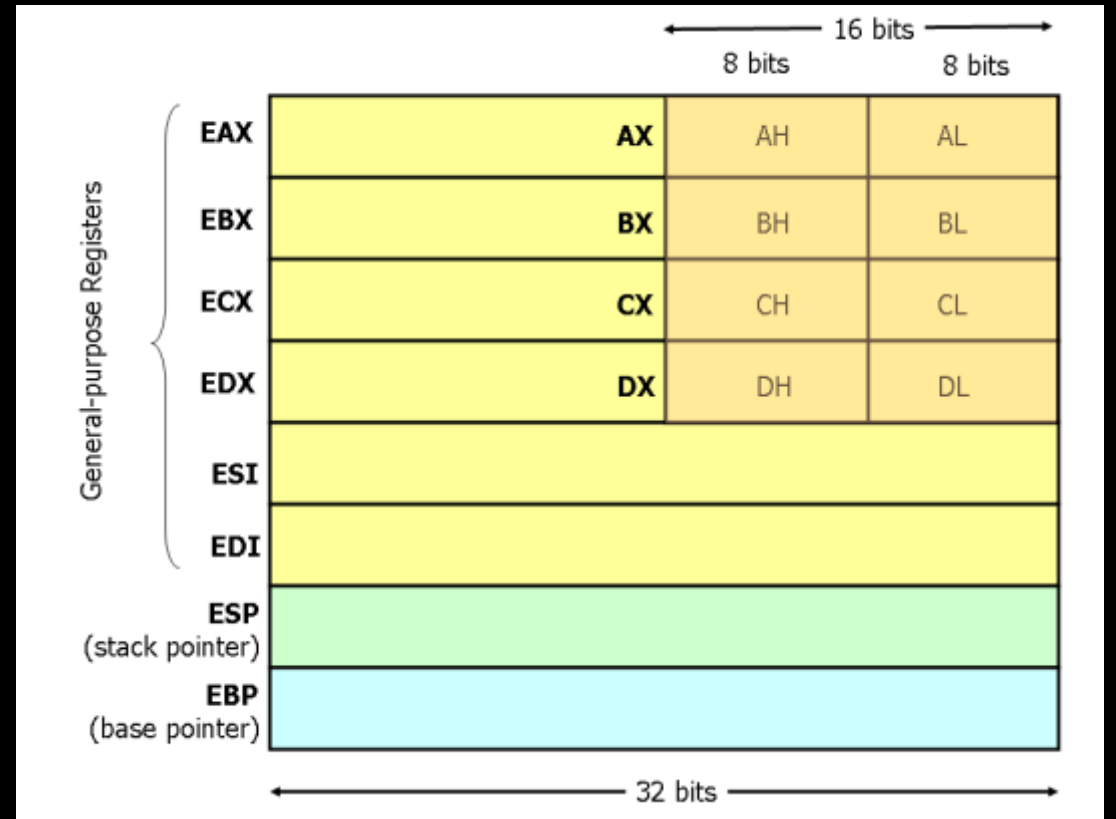
## Kernel Address Mapping



- The kernel uses linear mapping from the 1<sup>st</sup> GB of the physical memory to the 4<sup>th</sup> GB of the virtual memory
- Physical memory beyond the 1<sup>st</sup> GB (high memory) can be accessed by kernel **only** via kernel page mapping (i.e., kernel page table)

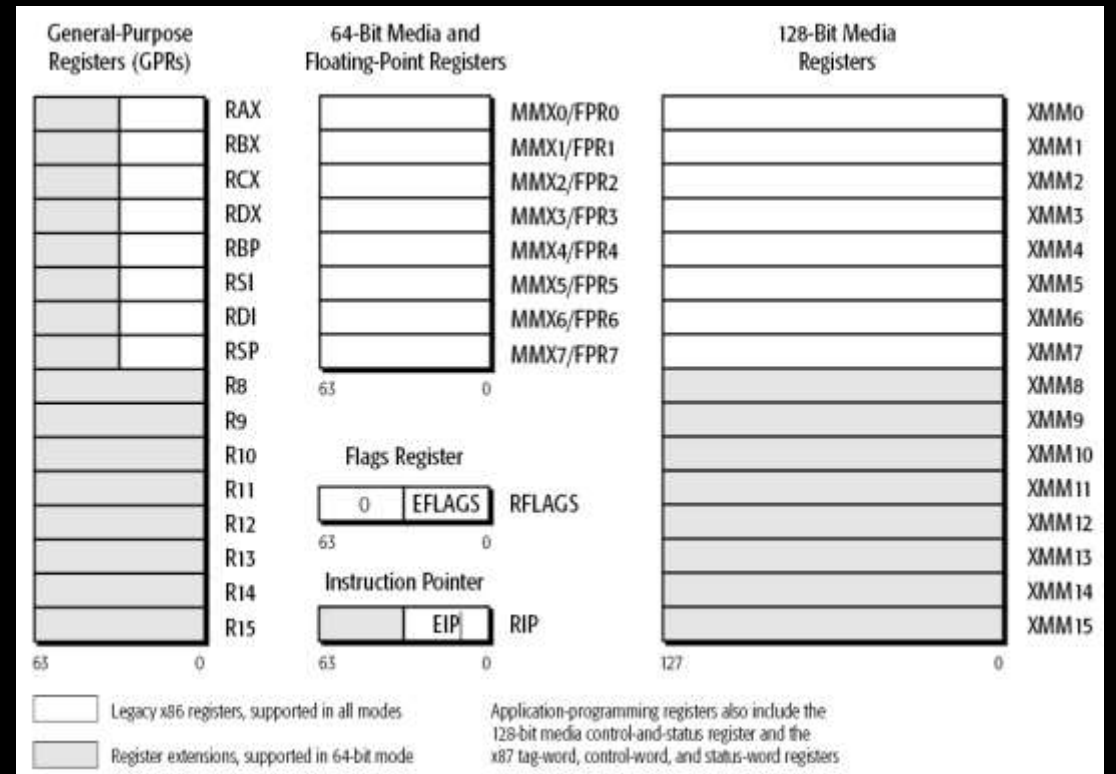
# i386 and amd64

- 通常我們會把 CPU 和指令集架構一起講
- CPU 分為 32 bits 和 64 bits
- 這裡的 32 bits 和 64 bits 指的是 CPU registers 的大小
- i386 是一個統稱代表所有 32 bits 架構的 CPU
- IA-32的機器有i386、i486、i586、i686.....也可以稱為 x86



# i386 and amd64

- amd64、x86-64、x64指的都是 64 bits的指令集架構
- amd64 是因為 amd 率先使用了 64 bits 的技術
- 同樣的所有的 registers 都是 64 bits
- 現在大多數電腦作業系統都使用 64 bits



# i386 and amd64

## Q&A time

- 要是在 32 bits 的 CPU 上插超過 4GB 的 RAM 會怎麼樣?

沒有功用因為 x86 的 CPU 只能定址到  $2^{32}$  的 memory

- Program Files 和 Program Files (x86) 資料夾有什麼差?

因為 x64 的 CPU 也可以執行 32 bits 的程式，所以會把 x86 的程式安裝在這個資料夾裡

# OS

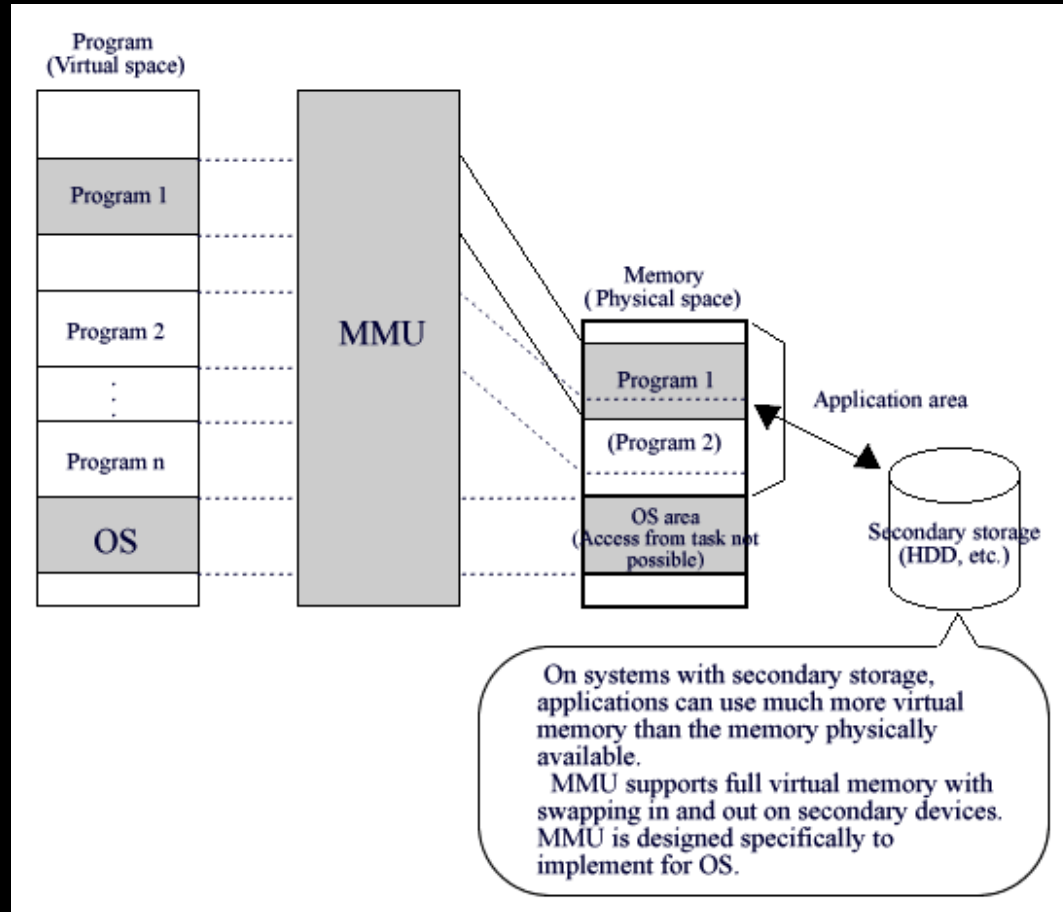
- 一支長駐在記憶體中的程式
- 扮演硬體與軟體間的橋梁
- 有效率地規劃並使用硬體資源
- 提供 API 給 program 來存取硬體資源
- Linux System call

[http://docs.cs.up.ac.za/programming/asm/derick\\_tut/syscalls.html](http://docs.cs.up.ac.za/programming/asm/derick_tut/syscalls.html)

- 分成兩種 Mode ( Kernel Mode and User Mode )



# OS



# OS

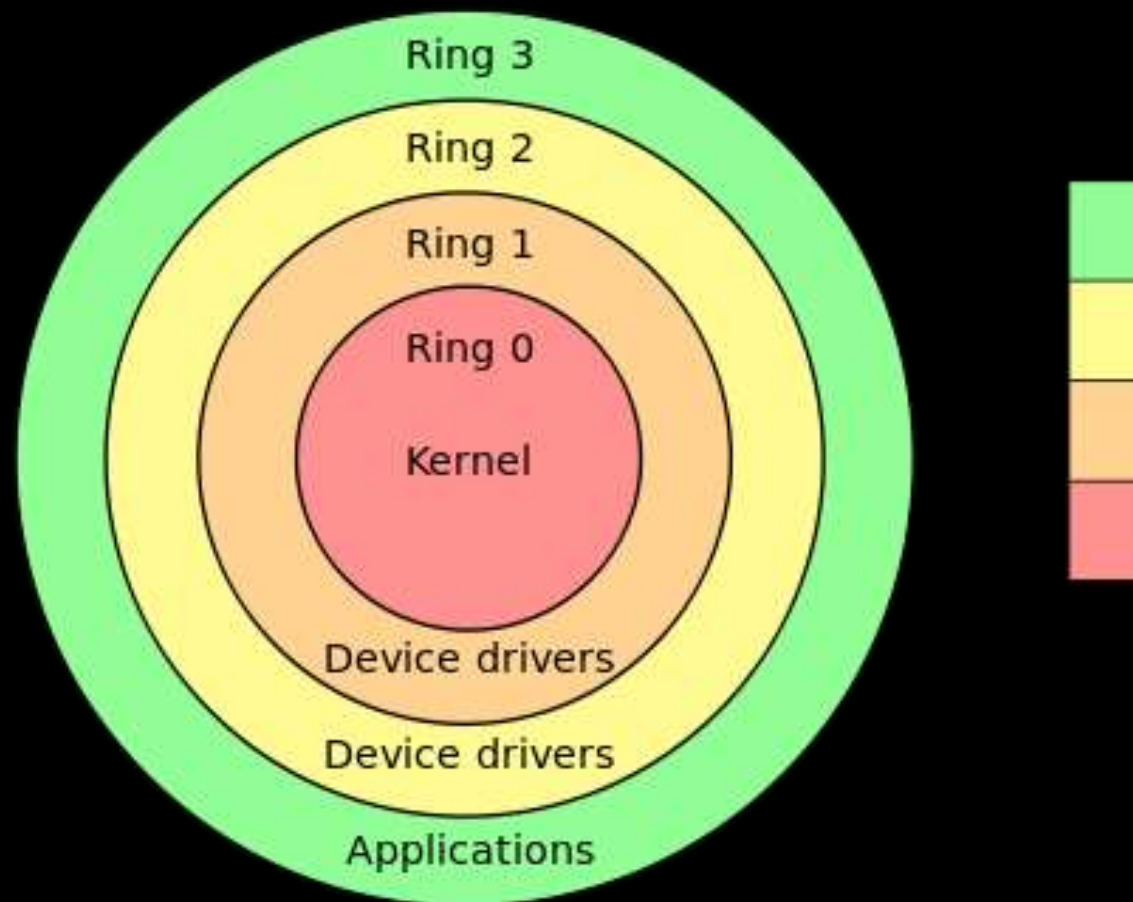
- 實際上只有使用 Ring 0 ( User Mode ) 和 Ring 3 ( Kernel Mode )

- Kernel Mode

所有指令都可以執行

- User Mode

只能執行一些有限的指令，可以透過 Interrupt 來切換到 Kernel Mode



# Vim

- Command-line based text editor
- open terminal type vimtutor
- 教學部分請各位自行參考
- Vbird

[http://linux.vbird.org/linux\\_basic/0310vi.php](http://linux.vbird.org/linux_basic/0310vi.php)

- Study Area

<http://www.study-area.org/tips/vim/>





# Vim

- Vundle

[https://github.com/VundleVim/Vu  
ndle.vim](https://github.com/VundleVim/Vundle.vim)

- Powerline

<https://github.com/powerline/powerline>

- NERD-tree

<https://github.com/scroolose/nerdtree>

```
" Installing plugins to /home/starcraftman/.vim/bundle
Plugin 'gmarik/Vundle.vim'
Plugin 'Valloric/YouCompleteMe'
Plugin 'scrooloose/syntastic'
Plugin 'bling/vim-airline'
Plugin 'SirVer/ultisnips'
Plugin 'edsono/vim-matchit'
Plugin 'elzr/vim-json'
Plugin 'honza/vim-snippets'
Plugin 'justinmk/vim-sneak'
+ Plugin 'kien/ctrlp.vim'
Plugin 'ludovicchabant/vim-lawrencium'
Plugin 'majutsushi/tagbar'
Plugin 'mhinz/vim-signify'
+ Plugin 'plasticboy/vim-markdown'
Plugin 'scrooloose/nerdcommenter'
Plugin 'sjl/gundo.vim'
+ Plugin 'tpope/vim-fugitive'
Plugin 'tpope/vim-sleuth'
Plugin 'tpope/vim-surround'
Plugin 'tyru/open-browser.vim'
Plugin 'vim-scripts/a.vim'
v Plugin 'tomasr/molokai'
v Plugin 'flazz/vim-colorschemes'
Helptags
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
set nocompatible " Required
filetype off
set rtp+=~/./vim/bundle/Vundle.vim
call vundle#begin()

" Let Vundle manage itself.
Plugin 'gmarik/Vundle.vim'


" Plugins
Plugin 'Valloric/YouCompleteMe'
Plugin 'scrooloose/syntastic'
Plugin 'bling/vim-airline'
Plugin 'SirVer/ultisnips'
Plugin 'edsono/vim-matchit'
Plugin 'elzr/vim-json'
Plugin 'honza/vim-snippets'
Plugin 'justinmk/vim-sneak'
Plugin 'kien/ctrlp.vim'
Plugin 'ludovicchabant/vim-lawrencium'
Plugin 'majutsushi/tagbar'
Plugin 'mhinz/vim-signify'
Plugin 'plasticboy/vim-markdown'
Plugin 'scrooloose/nerdcommenter'
Plugin 'sjl/gundo.vim'
|| Plugin 'tpope/vim-fugitive'
Plugin 'tpope/vim-sleuth'
Plugin 'tpope/vim-surround'
Plugin 'tyru/open-browser.vim'
Plugin 'vim-scripts/a.vim'


" Color schemes
Plugin 'tomasr/molokai'
Plugin 'flazz/vim-colorschemes'


" Required, plugins available after.
call vundle#end()
filetype plugin indent on
```

<view>> [Vundle] Installer << 92% | 24: 1 .vimrc 38% | 322: 1  
Processing 'flazz/vim-colorschemes'

# Screen and Tmux

- 不想開太多 Terminal?
- 想回到上次的工作階段?
- Keep online
- 既然 Tmux 比較新我們就學 Tmux 吧!!!
- Debian GNU / Linux

`apt-get install tmux`

# Screen and Tmux

```
ile.c
libtool: compile: x86_64-pc-linux-gnu-gcc -DHAVE_CONFIG_H -I. -I. -I../include -I../include -I./ref -I/usr/include/et -pipe -O2 -march=native -D_LARGE_FILES= -Wall -Wmissing-prototypes -Wpointer-arith -Wbad-function-cast -Wmissing-declarations -Wnested-externs -pipe -O2 -march=native -c file.c -fPIC -DPIC -o .libs/libhx509_la-file.o
/bin/sh ../libtool --tag=CC --mode=compile x86_64-pc-linux-gnu-gcc -DHAVE_CONFIG_H -I. -I. -I../include -I../include -I./ref -I/usr/include/et -pipe -O2 -march=native -D_LARGE_FILES= -Wall -Wmissing-prototypes -Wpointer-arith -Wbad-function-cast -Wmissing-declarations -Wnested-externs -pipe -O2 -march=native -c -o libhx509_la-sel.lo `test -f 'sel.c' || echo './'`sel.c
libtool: compile: x86_64-pc-linux-gnu-gcc -DHAVE_CONFIG_H -I. -I. -I../include -I../include -I./ref -I/usr/include/et -pipe -O2 -march=native -D_LARGE_FILES= -Wall -Wmissing-prototypes -Wpointer-arith -Wbad-function-cast -Wmissing-declarations -Wnested-externs -pipe -O2 -march=native -c sel.c -fPIC -DPIC -o .libs/libhx509_la-sel.o
/bin/sh ../libtool --tag=CC --mode=compile x86_64-pc-linux-gnu-gcc -DHAVE_CONFIG_H -I. -I. -I../include -I../include -I./ref -I/usr/include/et -pipe -O2 -march=native -D_LARGE_FILES= -Wall -Wmissing-prototypes -Wpointer-arith -Wbad-function-cast -Wmissing-declarations -Wnested-externs -pipe -O2 -march=native -c -o libhx509_la-sel-gram.lo `test -f 'sel-gram.c' || echo './'`sel-gram.c
libtool: compile: x86_64-pc-linux-gnu-gcc -DHAVE_CONFIG_H -I. -I. -I../include -I../include -I./ref -I/usr/include/et -pipe -O2 -march=native -D_LARGE_FILES= -Wall -Wmissing-prototypes -Wpointer-arith -Wbad-function-cast -Wmissing-declarations -Wnested-externs -pipe -O2 -march=native -c sel-gram.c -fPIC -DPIC -o .libs/libhx509_la-sel-gram.o
```

---

```
HOST="x86_64-pc-linux-gnu"
x86_64_pc_linux_gnu_CFLAGS="-pipe -O2 -march=native"
i686_pc_linux_gnu_CFLAGS="-pipe -O2 -march=native"

case "${CATEGORY}/${PN}" in
  sys-apps/paludis)
    NORMAL >> /etc/paludis/bashrc < sh << 9% : 1: 1
```

```
[exbull:0] [1:vim] | 2:zsh |
```

```
1 [|||||||] 62.9% Tasks: 48, 8 thr; 1 running
2 [|||||||] 22.5% Load average: 1.34 1.07 0.62
Mem[|||||||] 103/247MB Uptime: 00:21:04
Swp[|] 7/15359MB
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	IORW	TIME+	Command
7583	paludisbu	20	0	8652	2112	1788	S	59.3	0.8	0	1:26.62	sydbox -
72	root	20	0	19132	2452	2304	S	0.6	1.0	0	0:01.11	/usr/lib
271	tureba	20	0	23928	6660	2388	S	0.0	2.6	0	0:04.87	tmux -u2
14177	paludisbu	20	0	6952	2400	1736	S	0.0	0.9	0	0:00.03	make all
12147	root	20	0	480M	19820	15032	S	0.0	7.8	0	0:04.67	cave exe
16659	tureba	20	0	14272	2920	2404	R	0.0	1.2	0	0:00.76	htop
14030	paludisbu	20	0	6980	2316	1656	S	0.0	0.9	0	0:00.04	make all
235	tureba	20	0	76444	3464	2740	S	0.0	1.4	0	0:01.78	sshd: tu
7584	root	20	0	118M	17788	15168	S	0.0	7.0	0	0:00.11	cave per
994	tureba	20	0	29212	8012	4508	S	0.0	3.2	0	0:00.22	vim /etc
26696	root	20	0	118M	17788	15168	S	0.0	7.0	0	0:00.59	cave per

```
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice +F9Kill F10Qu
```

README	autom4te.cache	configure	lnet	snmp
Rules	build	configure.ac	lustre	stamp-h1

```
[11:04:40|1023] (tureba@exbull)% cd ../ompi
tre)
[11:04:46|1024] (tureba@exbull)% ls
(master 952be15 ~/o
mpi)
AUTHORS Makefile.am VERSION config.lt libtool
Doxyfile Makefile.in aclocal.m4 config.status ompi
HACKING Makefile.ompi-rules autogen.pl configure opal
INSTALL NEWS autom4te.cache configure.ac orte
LICENSE README config contrib oshmem
Makefile README.JAVA.txt config.log examples test
[11:04:46|1025] (tureba@exbull)%
(master 952be15 ~/o
mpi)
[11:07:11|1025] (tureba@exbull)%
(master 9[11:07:27|
1025][11:07:32|1025] (tureba@exbull)%
(master 952be1
5 ~/om[11:09:44|1025] (tureba@exbull)%
(master 9
[11:09:52|1025] (tureba@exbull)%
(master 952be15 ~/ompi)
[~] | 2015-04-28 11:09
```

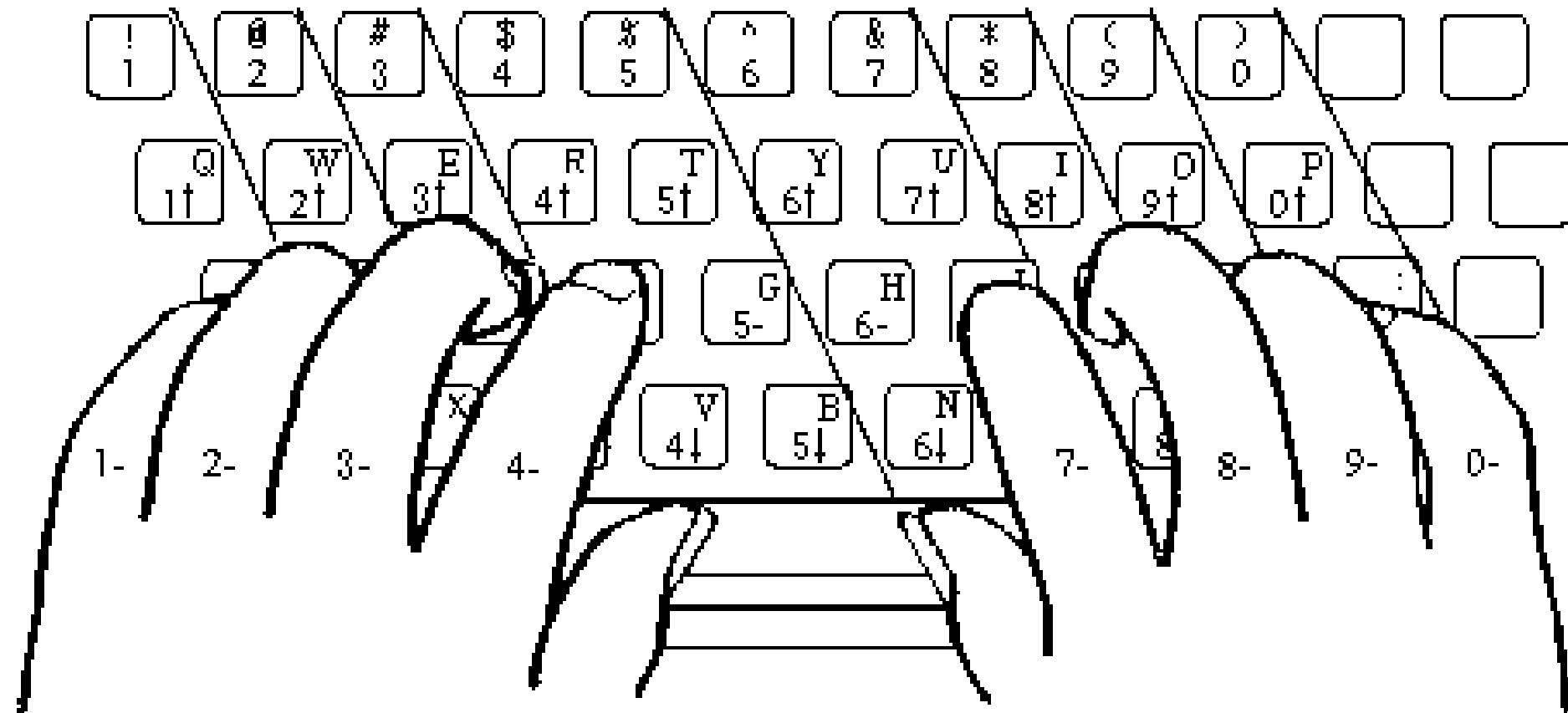
# Screen and Tmux

- `Ctrl+b` 組合鍵 : `Ctrl` 按住不放再按**b**
- `Ctrl+b c` : 建立新的視窗
- `Ctrl+b &` : 刪除目前的視窗
- `Ctrl+b n` : 切換到下一個視窗
- `Ctrl+b p` : 切換到上一個視窗
- `Ctrl+b d` : detach 目前的 session
- `Ctrl+b attach` : 回到上次 detach 的 session
- `Ctrl+b %` : 左右分割兩個視窗

# Fast tips

- 能不要碰滑鼠盡量不要碰
- Bash 快捷鍵像是 Ctrl+a 可以跳到行首
- 熟悉 vim 裡面的 Mode 切換和快捷鍵使用
- 熟悉 nc、wget、cat、echo.....指令
- 有現成的工具就用現成的，不要重複去寫需要的功能
- 請盡量保持手型像是下一張投影片這樣
- 左手食指放在 f 右手食指放在 j 上

# Fast tips



# Reference

- VMWare Player vs. VirtualBox

<http://teddy-chen-tw.blogspot.tw/2010/07/vmware-player-vs-virtualbox.html>

- Vbird鳥哥私房菜

[http://linux.vbird.org/linux\\_basic/0105computers.php](http://linux.vbird.org/linux_basic/0105computers.php)

- Memory layout of c program

<http://www.geeksforgeeks.org/memory-layout-of-c-program/>

- 程式設計師的自我修養

<http://www.books.com.tw/products/0010456858>



# Reference

- Wiki os

[https://en.wikipedia.org/wiki/Operating\\_system](https://en.wikipedia.org/wiki/Operating_system)

- Wiki x86

<https://zh.wikipedia.org/wiki/X86>

- Wiki x86-64

<https://zh.wikipedia.org/wiki/X86-64>

- Screen and Tmux

[https://nasa.cs.nctu.edu.tw/sa/2015/slides/IRC\\_tmux\\_screen.pdf](https://nasa.cs.nctu.edu.tw/sa/2015/slides/IRC_tmux_screen.pdf)