Computer Architecture

Lab1:Building Experiment Environment

VirtualBox Introduction

VirtualBox can run many guest operating systems on a host operating system in X86/AMD64 machine.

We use it to build a Linux environment for developing our full system simulation platform.

QEMU Introduction

- Unlike Virtualbox is binding with IA32, QEMU can run OSes and programs made for one machine on a different machine.
- QEMU makes this characteristic possible by supporting the portable dynamic translation.

QEMU Introduction (Cont.)

portable dynamic translation



Building Linux EnvironmentGet Virtualbox

https://www.virtualbox.org/wiki/Downloads

- Get CentOS image
 - http://61.164.110.188:82/Centos/
 - □ We use CentOS-5.8-x86_64-bin-DVD(1&2) as example.
 - □ You can chose the proper one or other Linux distribution such as Ubuntu.

Building Virtual Machine

- Install Virtualbox
- Build the virtual machine
 - □ 點選新增後按照步驟執行(如果是採用centos,版本可選Red Hat 64)

□ 虛擬機器資料夾可在"檔案->喜好"設定內修改



Setting Virtual Machine

■ 點選系統->存放裝置 選擇光碟機圖示

」選擇虛擬光碟機並且選取剛剛所下載的CentOS映像檔



Setting Virtual Machine (Cont.)

選擇"共用資料夾"並且選擇掛載之資料夾,勾取自動掛載。之後 此虛擬機器一開機即可透過此資料夾與host OS共享資料



■ 執行啟動就會自動進入CentOS安裝程式

🦉 CentOS-5.8 [執行中] - Oracle VM VirtualBox	2 CentOS-5.8 (執行中) - Oracle VM VirtualBox 時発(M) 結果(Q) 装要(D) 登録(Q)
機器(M) 檢視(V) 裝置(D) 說明(H)	
CentOS-5	The default installation of CentOS includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?
Community ENTerprise Operating System	Desktop - KDE Server Please select any additional repositories that you want to use for software installation.
 To install or upgrade in graphical mode, press the <enter> key.</enter> To install or upgrade in text mode, type: linux text <enter>.</enter> 	You can further customize the software selection now, or after install via the software
- Use the function keys listed below for more information. [F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue] boot: _	O Customize Jater <u>©</u> <u>Customize now</u> <u>©</u> <u>Customize now</u> <u>©</u> <u>Customize now</u> <u>©</u> <u>Customize now</u> <u>©</u> <u>Customize now</u> <u>©</u> <u>Customize now</u>
😂 😳 🖉 🗗 🛄 🚫 🕑 Right Cul 🔐	Belease Notes

- Base System選取JAVA和System Tools
- Development除了KDE不選,其他全選
- CentOS可用root登入,但之後的範例都是以使用者登入為預設 狀況(使用者模式下的終端機以su指令可進入root模式)

a CentOS-5.8 [執行中] - Oracle VM VirtualBox 器(M) 檢視(V) 裝置(D) 説明(H) CENTOS		 CentOS-5.8 (執行中) - Oracle VM VirtualBox 機器(M) 檢視(V) 装置(D) 説明(H) ● CentOS 	
Applications	Administration Tools	Base System	Development Libraries
Base System	🔿 🗹 Base	Cluster Storage	💿 💕 🗹 Development Tools
Cluster Storage	👌 🗹 Dialup Networking Support	Clustering	👻 🗹 GNOME Software Development
Clustering	🗉 🔍 🧝 🖾 👘	Desktop Environments	💂 🖉 Java Development
Desktop Environments	🕤 🗆 Legacy Software Support	Development	🗉 🔣 🗆 KDE Software Development
Development	💿 🗆 OpenFabrics Enterprise Distribut	Languages	🔀 🛛 Legacy Software Development
Languages	System Tools	Servers	💥 🗹 Ruby
Servers		Virtualization	
This group is a collection of graphic managing user accounts and config	al administration tools for the system, such as for uring system hardware.	The packages in this group are core lib	raries needed to develop applications. 36 of 67 optional packages selected Optional packages
<u>R</u> elease Notes	(⊉ <u>B</u> ack ⇒ <u>N</u> ext	<u>Release Notes</u>	⊕ <u>Back</u> ↓ Pext
	😂 😳 🖉 🗗 🛄 🚫 🖳 Right Ctrl 🔡		😂 😳 🖉 🖶 🧰 Right Ctrl 👘

- 由於官方停止支援CentOS-5.8
- 修改CentOS-Base.repo



輸入指令 vim /etc/yum.repos.d/CentOS-Base.repo
將紅框內的部分註解掉,加上綠框內的部分

Applications Pla	ices System 😸 🎯 🤤 🔂	Applications Pl	aces System 🏽 🏵
Computer	caslab@localhost:/home/caslab	Computer	caslab@localhost:/home/caslab
	Elle Edit View Terminal Tabs Help		<u>File Edit View Terminal Tabs H</u> elp
	# CentOS-Base.repo		#
			# If the mirrorlist= does not work for you, as a fall back you can try the
caslab's Home	# The mirror system uses the connecting IP address of the client and the	caslab's Home	# remarked out baseurl= line instead.
	# update status of each mirror to pick mirrors that are updated to and		
	# unless you are manually picking other mirrors.		
	#		[base]
Trash	# If the mirrorlist= does not work for you, as a fall back you can try the	Trash	name=CentOS-\$releasever - Base
	<pre># remarked out baseurl= line instead.</pre>		<pre>#mirrorlist=http://mirrorlist.centos.org/?release=sreleasever&arch=\$basearch&repo=os "botto:"</pre>
	#		baseurl=http://mitror.centos.org/centos/sreleasever/os/spasearch/
			apacheck=1
	[base]		gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
	name=CentOS-\$releasever - Base		
	mirrorlist=http://mirrorlist.centos.org/?release=\$releasever&arch=\$basearch&repo=os		#released updates
	<pre>#basedic=http://mitrof.centos.org/centos/sieteasever/os/sbasearch/ anachec=1</pre>		name=CentOS-\$releasever - Updates
	gpgkev=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Cent0S-5		#mirrorlist=http://mirrorlist.centos.org/?release=\$releasever&arch=\$basearch&repo=updates
			#baseurl=http://mirror.centos.org/centos/\$releasever/updates/\$basearch/
	#released updates		baseurl=http://vault.centos.org/5.11/updates/\$basearch/
	[updates]		gpgcneck=1 angkev_file:///etc/nki/rpm-ang/RPM-GPG-KEY-CentOS-5
	mirrorlist=http://mirrorlist.centos.org/?release=Sreleasever&arch=Sbasearch&repo=update		gegreg-ricer//recepted/tem geg/rith and rich central s
	<pre>#baseurl=http://mirror.centos.org/centos/\$releasever/updates/\$basearch/</pre>		#additional packages that may be useful
	gpgcheck=1		[extras]
	gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5		name=Cent05-\$releasever - Extras
	#additional packages that may be useful		#harrortist=http://mirrortist.centos.org/reteasever/extras/shasearch/
	[extras]	P	baseurl=http://yault.centos.org/5.11/extras/Sbasearch/
	pame=CentOS-Sreleasever - Extras		gpgcheck=1
	mirrorlist=http://mirrorlist.centos.org/?release=\$releasever&arch=\$basearch&repo=extras		gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
	#baseurl=http://mirror.centos.org/centos/\$releasever/extras/\$basearch/		Hadditional packages that extend functionality of existing packages
	gpgcneck=1 gonkey=file:///etc/nki/rpm-gng/RPM-GPG-KEY-Cent0S-5		Centoslusi
	12,0-1 Top		"/etc/yum.repos.d/CentOS-Base.repo" 55L, 2091C 40,1 28%

- 安裝必要套件: yum install kernel-devel kernel-headers gcc gcc-c++
- 再以" #yum upgrade" 來更新CentOS所有套件
- 更新完且重開機後,為了讓CentOS與Host OS互動性更好,我們必須安裝Guest Additions(執行VBoxLinuxAdditions.run後重開機)



Install QEMU

Get QEMU

- http://wiki.qemu.org/download/qemu-0.15.1.tar.gz
- 建立工作資料夾,並將qemu-0.15.1.tar.gz複製到 workstation進行解壓縮
 - mkdir workstation
 - □ cp ./qemu-0.15.1.tar.gz ./workstation
 - □ cd workstation/
 - □ tar –zxvf qemu-0.15.1.tar.gz

Install QEMU(Cont.)

設定QEMU的模擬對象以及所要安裝的目錄

- cd qemu-0.15.1
- ./configure --target-list=arm-softmmu,arm-linux-user --prefix=/home/{your username}/qemu-bin
- --target-list為我們所希望模擬的目標選項。這裡的arm-softmmu代表我們想要 QEMU針對整個平台(包含CPU與周邊硬體)做模擬(如realview versatile family), 而arm-linux-user則是只做CPU指令集架構的轉換模擬。兩者的CPU指令集架構 都是透過Binary Transation完成。
- □ --prefix為安裝目的資料夾選項
- make && make install
- 編譯過程如果因為glibc版本過舊會發生compile error
 - 修改qemu-0.15.1/qemu-ga.c 第149行
- if (g_strcmp0(domain, "syslog") == 0)必須改成 if (domain && strcmp(domain, "syslog") == 0)

Install QEMU(Cont.)

- 安裝完成後,/home/{your username}/qemu-bin/bin中有所有的 QEMU執行檔
 - qemu-arm 為 arm-linux-user產物
 - qemu-system-arm 為 arm-softmmu 產物
 - □ 之後的實驗,QEMU的部分我們將透過這兩個執行檔完成。

Cross Compiler

- A cross compiler can build the executable code for the target platform other than the one on which the compiler is run.
- ARM cross compiler
 - □ ARM-elf-gcc, ARM-linux-gcc
 - □RVDS, ADS
 - Build ARM executable code on X86/AMD64 machine

Setting ARM Linux GCC

Get ARM-Linux-gcc (Mentor)

- https://sourcery.mentor.com/sgpp/lite/arm/portal/subscription3057
- 我們以2011.03-41版本(Lite Edition)作範例
 - □ GCC 4.5.2
 - □ Glibc 2.13
 - □ Linux Kernel 2.6.38
- 如果是安裝Ubuntu 64bit版本,則需要另外安裝IA32函 示庫。
 - □ sudo apt-get install ia32-libs

Setting ARM Linux GCC (Cont.)

- 解壓縮並且將執行檔宣告至預設執行區(壓縮檔先行放在 家目錄)
 - □ tar –xvf arm-2011.03-41-arm-none-linux-gnueabi-i686-pclinux-gnu.tar.bz2
 - export PATH= "/home/{your username}/workstation/arm-2011.03/bin:\$PATH"
 - □ 請注意: export此行指令,只要terminal重開就必須再次執行。可用 vim ~/.bashrc 加入此行指令讓terminal開啟時自動執行此行指令。
- 執行"arm-none-linux-gnueabi-gcc –v" 觀察是否設 定完成。

Compile C code

- 利用編輯器完成一段簡短的C程式碼並且使用arm-linux-gcc 編譯。
 - 🗆 cd ~ && mkdir test

10

- □ 在test資料夾中新增test.c 並寫幾行簡單程式
- arm-none-linux-gnueabi-gcc test.c -o test.o
- arm-none-linux-gnueabi-objdump –xD test.o > dump.txt
- □ 從dump.txt可以看出透過cross compiler我們編出ARM code了!

🖹 test.c 🗶	📄 dump.txt 🗶
1 #include <stdio.h></stdio.h>	1
2 3 int main(){ 4	<pre>2 test.o: file format elf32-littlearm 3 test.o 4 architecture: arm, flags 0x00000112:</pre>
<pre>5 printf("COURSE : Computer Architecture\n"); 6</pre>	5 EXEC_P, HAS_SYMS, D_PAGED 6 start address 0x00008380
7 return 0; 8 9}	7

Execute ARM code

■ 最後透過我們一開始所編譯出的qemu-arm來執行test.o

□ 將所在目錄移至test資料夾

- ~/qemu-bin/bin/qemu-arm _L /home/{your username}/workstation/arm-2011.03/arm-none-linuxgnueabi/libc test.o
- 執行檔qemu-arm –L後面所接的是cross compiler所提供的函示庫 (Library),最後的test.o則是我們剛才所編譯出來的檔案。當然您也 可以將test.o與qemu-arm搬移出來直接./qemu-arm –L {armlibrary位置} test.o
- 執行結果會直接秀在terminal上

edi@edi-virtualbox: ~/test

檔案(<u>F</u>) 編輯(<u>E</u>) 檢視(<u>V</u>) 終端機(<u>T</u>) 求助(<u>H</u>)

edi@edi-virtualbox:~/test\$./../qemu-bin/bin/qemu-arm -L /home/edi/arm-2011.03/arm-none-linux-gnueabi/libc test.o COURSE : Computer Architect<u>u</u>re