Lab 1

Embedded Linux Development Environment
Multiple OS in a PC

- VMware

VMware Workstation Player - VMware Products

Try VMware Workstation Player

VMware Workstation Player builds on the industry leading foundation of Workstation Pro, and delivers a streamlined user interface for creating and running operating systems and applications in a virtual machine.

The free version is available for non-commercial, personal and home use. We also encourage students and non-profit organizations to benefit from this offering.

Commercial organizations require paid licenses to use Workstation Player.

Need a more advanced virtualization solution? Check out Workstation Pro for Windows or Workstation Pro for Linux

VMware Workstation 12.5 Player for Windows 64-bit
Download Now

VMware Workstation 12.5 Player for Linux 64-bit
Download Now
Multiple OS in a PC

- **Oracle Virtual Box**

---

**VirtualBox**

**Download VirtualBox**

Here, you will find links to VirtualBox binaries and its source code.

**VirtualBox binaries**

By downloading, you agree to the terms and conditions of the respective license.

- **VirtualBox 5.1.12 platform packages.** The binaries are released under the terms of the GPL version 2.
  - Windows hosts
  - OS X hosts
  - Linux distributions
  - Solaris hosts

- **VirtualBox 5.1.12 Oracle VM VirtualBox Extension Pack** All supported platforms
  Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel chain.
  The Extension Pack binaries are released under the VirtualBox Personal Use and Evaluation License (PUEL).
  Please install the extension pack with the same version as your installed version of VirtualBox.
  If you are using VirtualBox 5.0.30, please download the extension pack here.

- **VirtualBox 5.1.12 Software Developer Kit (SDK)** All platforms
Development Environment

Host: Windows PC
- VMware player
- Linux Host

Target: Embedded Linux
- Application program
- Embedded Linux kernel
- Booting
- Flash Memory

192.168.0.2
192.168.0.120

Ethernet
Bridged

192.168.0.60
Linux

- Debian
- Ubuntu
- Linux Mint
- Redhat
- CentOS
- Fedora
VMware Player
VMware Player
Select **Ubuntu1204_64** and play.
VMware Player

- Or double click **Ubuntu1204_64.vmx** in the folder D:/Ubuntu1204_64
Log In

- ID: control, password:
- ID: root, password: control
- 마우스 포인터가 안 보일 경우에는 Ctrl-Alt
Open Terminal

- `pwd`: print working directory
- `ls`: list directory contents

![Terminal output]

```
control@lab-pc2:~$ pwd
/home/control
control@lab-pc2:~$ ls
arm-2014.05-29-arm-none-linux-gnueabii686- pc-linux-gnu.tar.gz
Fedora
Desktop
Documents
Downloads
examples.desktop
kernel.tar.gz
lic_server
linux_filesystem
linux_filesystem.tar.gz
Music
Pictures
Public
ratlLicenseKeyServer-8-1-1_RHEL_x86.tar
ratlRhapiusy_7-5-3_Linux.tar.gz
reg.log
Rhapsody753
```
New Terminal

- Right button click
Check IP address

- `ifconfig`
Open serial terminal

- Open **Windows** Tera Term or Hyper Terminal and select COM1: serial port
Setup Serial port
Setup Serial port

- 115200, 8,N,1
Save setup
Turn ON power switch
Power on and boot the target
Virtual Machine Network Setting

- Virtual machine settings
## Network Adapter

### Virtual Machine Settings

<table>
<thead>
<tr>
<th>Device</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>1 GB</td>
</tr>
<tr>
<td>Processors</td>
<td>1</td>
</tr>
<tr>
<td>Hard Disk (SCSI)</td>
<td>30 GB</td>
</tr>
<tr>
<td>CD/DVD (SATA)</td>
<td>Auto detect</td>
</tr>
<tr>
<td>Network Adapter</td>
<td>Bridged (Automatic)</td>
</tr>
<tr>
<td>USB Controller</td>
<td>Present</td>
</tr>
<tr>
<td>Sound Card</td>
<td>Auto detect</td>
</tr>
<tr>
<td>Printer</td>
<td>Present</td>
</tr>
<tr>
<td>Display</td>
<td>Auto detect</td>
</tr>
</tbody>
</table>

### Device status

- [ ] Connected
- [ ] Connect at power on

### Network connection

- Bridged: Connected directly to the physical network
  - [ ] Replicate physical network connection state

- NAT: Used to share the host’s IP address
- Host-only: A private network shared with the host
- Custom: Specific virtual network
  - VMnet0

- LAN segment:

  - [ ] LAN Segments...
  - [ ] Advanced...
IP addresses

- Windows: 192.168.0.2
- Ubuntu1204: 192.168.0.120
- Target: 192.168.0.60
IP addresses

![Network settings with IP addresses](image)

- **Hardware Address**: 00:50:FC:BC:E6:74
- **IP Address**: 192.168.0.120
- **Subnet Mask**: 255.255.255.0
- **Default Route**: 192.168.0.1
- **DNS**: 8.8.8.8

**Editing Wired connection 1**
- **Connection name**: Wired connection 1
- **Connect automatically**: Checked
- **Method**: Manual
- **Addresses**:
  - Address: 192.168.0.120, Netmask: 255.255.255.0, Gateway: 192.168.0.1
  - DNS servers: 8.8.8.8
Check target IP address
Check host-target connection

```
control@lab-pc2:~$ ping 192.168.0.60
PING 192.168.0.60 (192.168.0.60) 56(84) bytes of data.
64 bytes from 192.168.0.60: icmp_seq=1 ttl=64 time=0.603 ms
64 bytes from 192.168.0.60: icmp_seq=2 ttl=64 time=0.311 ms
64 bytes from 192.168.0.60: icmp_seq=3 ttl=64 time=1.20 ms
^C
--- 192.168.0.60 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.311/0.706/1.205/0.372 ms
```

```
[root@ACHRO ~]# ping 192.168.0.120
PING 192.168.0.120 (192.168.0.120): 56 data bytes
64 bytes from 192.168.0.120: seq=0 ttl=64 time=1.543 ms
64 bytes from 192.168.0.120: seq=1 ttl=64 time=1.677 ms
64 bytes from 192.168.0.120: seq=2 ttl=64 time=1.619 ms
--- 192.168.0.120 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.543/1.613/1.677 ms
```

Make working directory

- `mkdir`: make directory
- `cd`: change directory
- `vi hello.c`
Press ‘a’ key to start to enter
- Start typing
To finish typing press ‘esc’ key

```c
#include <stdio.h>

void main(void)
{
    printf("Hello\n");
}
```

"hello.c" [New File]
To save and quit press ‘:wq’ and Enter
Compile and run

- gcc hello.c
- arm-none-linux-gnueabi-gcc hello.c
Mount NFS (network file system)

- `mount -t nfs 192.168.0.120:/nfsroot /mnt/nfs -o rw,rsize=4096,nolock`
Run hello.c in target
Run hello.c in target
Run hello.c in target
Run hello.c in target
Run hello.c in target
Makefile

all: app install

app:       hello.c
           arm-none-linux-gnueabihf-gcc -o hello hello.c

install:   cp hello /nfsroot
- ‘vi hello.c’ to start editing
- Move the cursor using arrow keys or ‘h,j,k,l’ keys
- Press ‘i’ to insert
- To finish typing press ‘esc’ key
- To save ‘:w’ and Enter
- To quit ‘:q’ and Enter
To delete a character press ‘x’
To append a character ‘a’
To copy a line ‘yy’
To paste a line ‘p’
To delete and copy a line ‘dd’
To delete and copy multiple lines ‘#dd’, for example ‘5dd’ to delete and copy 5 lines
To insert a line ‘o’
To cancel the last action ‘u’
- Line number ‘:set nu’
- Replace a pattern ‘:%s/old/new/’
- Quit without saving ‘:q!’
- To copy from line number #1 to line number #2 to the next line of #3 ‘:#1,#2co#3’
- To move from #1 to #2 to the next line of #3 ‘:#1,#2m#3’
- To delete from #1 to #2 ‘:#1,#2d’
Text Editor
```c
#include <stdio.h>

void main(void)
{
    printf("Hello World\n");
}
```
Workspace Switcher
Shut Down

- Do not just close the VMware window
Exercise

- Hello World를 10번 프린트 하는 프로그램을 vi 에디터를 사용하여 편집한 후, linux host와 target에서 각각 실행해 보십시오.
File Sharing with Windows

- Edit virtual machine settings
Folder sharing

Shared folders expose your files to programs in the virtual machine. This may put your computer and your data at risk. Only enable shared folders if you trust the virtual machine with your data.

- Disabled
- Always enabled
- Enabled until next power off or suspend

Folders

<table>
<thead>
<tr>
<th>Name</th>
<th>HostPath</th>
</tr>
</thead>
</table>

[Virtual Machine Settings interface showing settings for General, Power, Shared Folders, VMware Tools, Unity, and Autologin.]
Welcome to the Add Shared Folder Wizard

This wizard will guide you through the steps of adding a new shared folder to your virtual machine.
Folder sharing
Folder sharing
Folder sharing

Specify Shared Folder Attributes
Specify the scope of this shared folder.

Additional attributes
- Enable this share
- Read-only
Folder sharing

Folder sharing exposes your files to programs in the virtual machine. This may put your computer and your data at risk. Only enable shared folders if you trust the virtual machine with your data.

- Disabled
- Always enabled
- Enabled until next power off or suspend

Folders

<table>
<thead>
<tr>
<th>Name</th>
<th>Host Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>D:\</td>
</tr>
<tr>
<td>E</td>
<td>E:\</td>
</tr>
</tbody>
</table>
Folder sharing

```
control@lab-pc2:~/mnt/hgfs
control@lab-pc2:~$ cd /mnt
control@lab-pc2:/mnt$ ls
hgfs
control@lab-pc2:/mnt$ cd hgfs
control@lab-pc2:/mnt/hgfs$ ls
D E
```