

Fall 2004 Embedded Systems

실험 1. Development Environment

1. Purpose

Get acquainted with the development system:

Target board: EZ-X5 board with EzBoot, Embedded Linux

Host computer: PC with Linux, cross compiler, NFS, and minterm

2. Hardware connection

User I/O -- PC -- Serial connection -- Embedded board
(keyboard and display) USB connection
 Ethernet connection

3. Problem 1.1

Compile and run the following hello program on the embedded board using NFS.

```
#include <stdio.h>

void main()
{
    printf("Hello, Embedded Board!\n");
}
```

Problem 1.2

Write a function named factorial(n) which computes the value of n!.

Factorial(n) should call factorial(n-1) recursively.

Factorial(1) simply returns the value of 1.

Factorial(0 or negative) returns the value of -1 (error).

Run the factorial program on the embedded board using NFS.

4. 실험 순서

1) Prepare the host PC (Prepared by TA)

Partition the disk.

Install Linux with Kernel 2.4.18.

Install the cross-development tool.

See EZ-X5 User's Manual Ch 6.

2) Initialize the host PC

Check the version of Linux kernel.

Login 바로 전의 line 에서 확인한다.

Check if the cross development tool exists.

Type /usr/armv5l-linux/bin/gcc --help.

Initialize the NFS system.

See EZ-X5 User's Manual Ch 7.

3) Connect the EZ-X5 and PC as shown in Section 2 (Hardware Connection).

Power on the EZ-X5 using power switch.

Install EzBoot using EzFlash: Skip if already installed.

See EZ-X5 User's Manual Ch 12.

Install embedded Linux (zImage and ramdisk.gz) using EzBoot: Skip if already installed.

See EZ-X5 User's Manual Ch 3.

Check booting of EzBoot and then Linux.

Press space key for EzBoot or return key for Linux.

4) Compile and run the Problem 1.1.

Use the PC cross-development system

Edit the program "hello.c" using vi editor

Cross-compile and link the program to produce hello.

Use the minterm on the PC

Boot the EZ-X5 to run Linux

Download using zmodem protocol

```
$ rz
```

Press Control+a s

Select zmodem

Select file to download as "hello.x"

Check that the file is already downloaded.

```
$ ls la
```

```
hello.x ...
```

Run hello, which should print the following on the console:

```
$ ./hello.x
```

```
Hello, Embedded board!
```

5) Run using NFS

Check if the hello file is accessible via NFS.

Run hello.x, which should print the following on the console:

```
Hello, Embedded Baord!
```

6) Compile and run the Problem 1.2.

.

Prepare your own factorial program.

Cross-development sequence is the same as above.

5. Demonstration (To TA, 9/23 Thu. 4 PM – 6 PM)

6. Report (To TA, Due 10/4 Mon 6 PM) – 각자 작성

Each student should prepare his own report containing:

Purpose
Experiment sequence
Experimental results
Discussion: should be different even for each member of the same team.
References

7. Reference

유영창 외, "EZ-X5 User's Manual", J.D&T Co., Ltd., 2003

8. Remarks

EZ-X5 는 조별로 한 unit 씩 한 학기동안 학부기술실에서 대여받아 사용하기 바랍니다.

Starting EZ-X5

1. System 연결 및 boot

1.1 Linux PC와 Ez-X5 board를 다음 Cable들로 연결한다
USB cable, serial cable, Ethernet cable

1.2 Linux PC를 boot하고, login하여 command prompt(예를 들어 '#')를 기다린다.

2. minicom을 초기화하고 수행한다 (EZ-X5 Manual 3장 참조).

2.1 Init minicom

```
# minicom -s
```

2.2 Set parameters

Serial port setup

```
A. Serial device:          /dev/ttyS0
E: bps/par/bits:          115200 8N1
F. Hardware flow control: NO
G. Software flow control: NO
```

Save setup as dfl

Exit

2.3 초기화는 한번으로 족하며, 그 이후는 다음 command로 minicom을 수행한다.

```
# minicom
```

```
# Startup message가 나타난다.
```

```
WELCOME EZBOOT.X5 V1.4 ..... For PXA255
```

```
.....
```

2.4 만일 startup message가 나타나지 않으면 연결을 재 확인하고, software(EzBoot, kernel, ramdisk)를 다시 설치한다.

2.9 minicom의 종료는 control+a, q 를 차례로 누른다.

3. Ethernet을 초기화한다 (EZ-X5 manual 14장 참조).

3.1 EZ-X5의 reset key를 눌러 시작하고, space bar를 입력하여 EZBOOT를 수행시킨다.
Quickly Autoboot [ENTER] / Goto BOOT-MENU [space bar]... ; Press space
key

```
EZBOOT> help  
.....
```

3.2 유효 적절한 host IP와 target IP를 설정한다. Local IP는 세번째 숫자까지 host IP와 동일해야 한다.

```
EZBOOT> set  
Select>> 2. LOCAL IP 143.248.150.101  
          3. HOST IP 143.248.150.129  
          S. SAVE  
          0. EXIT
```

3.3 Ping 으로 test한다.

```
EZBOOT> ping 143.248.150.129  
.....
```

3.4 Linux 에서도 ping 으로 test 한다.

```
EZBOOT> rst  
$ ping 143.248.150.129  
.....  
Press Control+C
```

Installing EZ-X5

1. Software cdrom을 disk로 복사

```
# mount /dev/cdrom /mnt/cdrom  
# cd /mnt/cdrom  
# mkdir some_directory/ezx5  
# cp -R /mnt/cdrom/* some_directory/ezx5
```

2. EZBOOT 설치 (EZ-X5 manual 12장 참조).

2.1 Start ezflash

```
# cd some_directory/ezx5  
# cd sw/image  
# ./ezflash.x5  
.....
```

2.2 Test flash read

```
# ./ezflash -f  
.....
```

```
Command? R
? Read Start Address ? [ ex] 000 ] : 0
? Read End Address ? [ ex] 100 ] : 100
.....
```

2.3 Write EZBOOT

```
# ./ezflash.x5 ezboot.x5 -w
.....
```

JTAG을 사용한 64KB flash write에 시간이 15분 정도 걸린다.

2.4 Verify EZBOOT

```
# ./ezflash.x5 -v ezboot.x5
.....
```

JTAG을 사용한 64KB flash verify에 시간이 3분 정도 걸린다.

2.5 Test EZBOOT

EZ-X5 board의 Reset key와 keyboard의 space key를 눌러 EZBOOT기 실행되는가 확인한다.

3. Kernel 설치 (EZ-X5 manual 4장 참조)

3.1 EZBOOT를 시작한다.

Reset switch 및 space key 사용.

3.2 zfk command 수행

```
EZBOOT> zfk
Press Control+a, s
Select zmodem
Select file: /home/cdrom/ezx5/sw/image/zImage.x5 ; 701 KB size
..... (Downloading)
READY: Press any key to continue...
..... (Erasing and writing)
```

4. Ramdisk image 설치 (EZ-X5 manual 4장 참조)

4.1 EZBOOT를 시작한다.

Reset switch 및 space key 사용.

4.2 zfr command 수행

```
EZBOOT> zfr
Press Control+a, s
Select zmodem
Select file: /home/cdrom/ezx5/sw/image/ramdisk.x5.gz ; 2855 KB size
..... (Downloading)
READY: Press any key to continue...
..... (Erasing and writing)
```

4.3 Linux가 제대로 수행되는가 test한다.

Installing Cross-Compiler for EZ-X5

1. Software cdrom을 disk로 복사

```
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom
# mkdir some_directory/ezx5
# cp -R /mnt/cdrom/* some_directory/ezx5 ; e.g., /home/cdrom/ezx5
```

2. Compiler directory 설정 및 rpm file 복사

```
# mkdir /home/ezx5/cross_compiler
# cd /home/ezx5/cross_compiler
# cp /home/cdrom/ezx5/cross_compiler/rmp-redhat8.0/* .
# ls
    armv5l-linux-binutils-2.13.90.0.16-ez1.i386.rpm
    armv5l-linux-gcc-3.2.1-ez1.i386.rpm
    armv5l-linux-glibc-2.3.1-ez1.i386.rpm
```

3. RPM 설치

```
# rpm -Uvh -nodeps -force armv5l-linux-binutils-2.13.90.0.16-ez1.i386.rpm
# rpm -Uvh -nodeps -force armv5l-linux-gcc-3.2.1-ez1.i386.rpm
# rpm -Uvh -nodeps -force armv5l-linux-glibc-2.3.1-ez1.i386.rpm
/usr/armv5l-linux directory 밑에 cross-compiler가 설치된다.
```

4. 환경설정

```
# cd /root
# vi .bash_profile
    Include: PATH=...:$PATH:/usr/armv5l-linux/bin:...
# logout and login
```

OR

```
# PATH=$PATH:/usr/armv5l-linux/bin
# export
```