

FAE78

Using Pegasos as development host for PPC targets

Maurie Ommerman, Jacob Pan, Phil Brownfield

June 10, 2004

Freescale Semiconductor Confidential Proprietary. Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2004



Tutorial contents

- **Using Pegasos as Linux development for disk-less PPC target boards**
 - General setup
 - Sandpoint MPC7447A
 - PQIII ADS board
- **Obtain and install new packages from the internet**

Benefits

Run fully featured Linux on multiple disk-less PPC targets

- MPC74xx
- PowerQuicc II, III

Complete native environment

- Native PPC tool chain, e.g. gcc, as, ls, Binutils, gdb, etc. JTAG/COP support too
- No need for cross architecture compiler, libraries

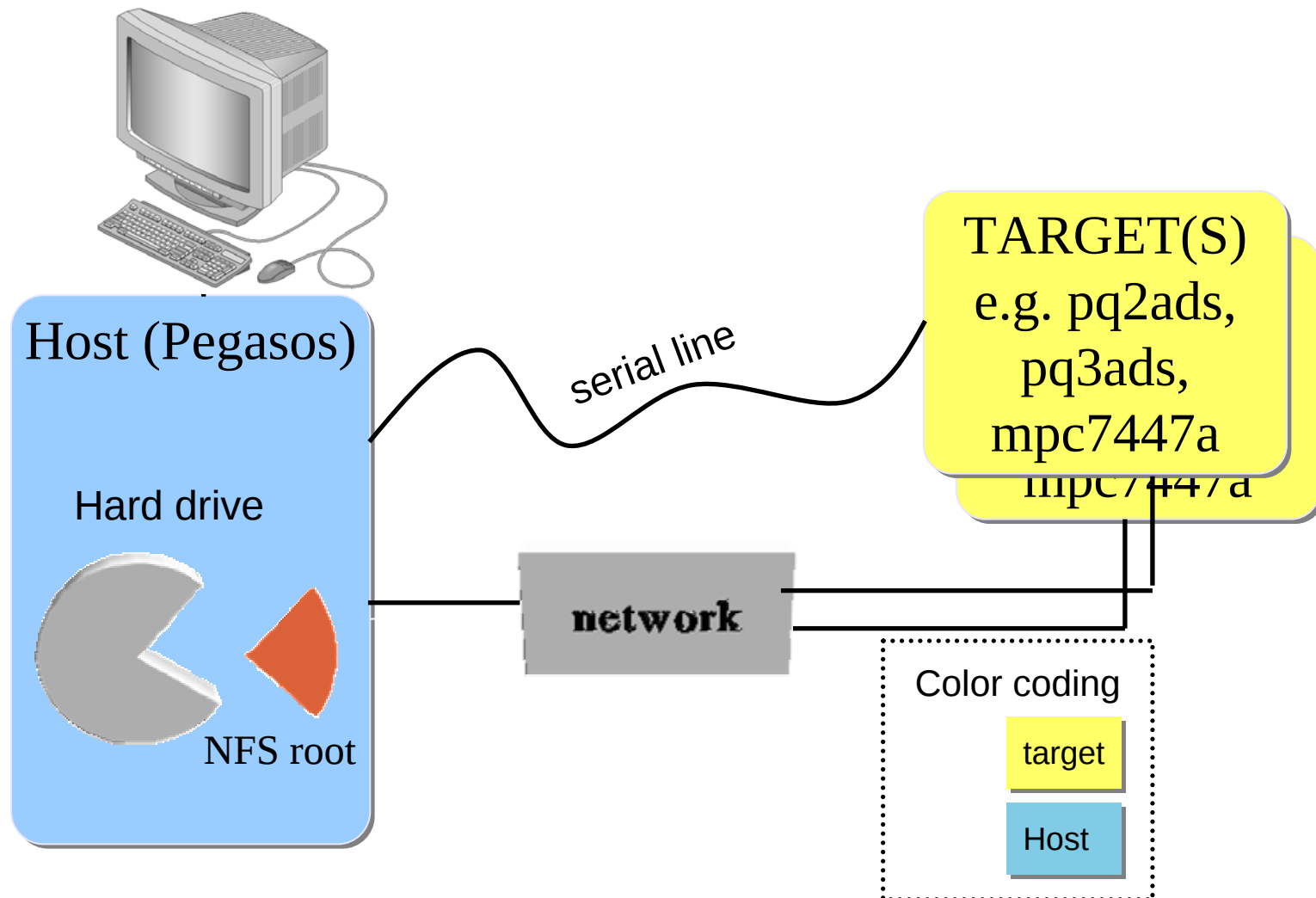
Coding, analyzing and verifying on the host before running on targets

- SIMG4+
- Performance monitoring

Suitable for kernel or device driver development

- Remote access to the target
- crash proof compared to hard drive, easy recovery
- more space compared to RAM disk

Development Environment

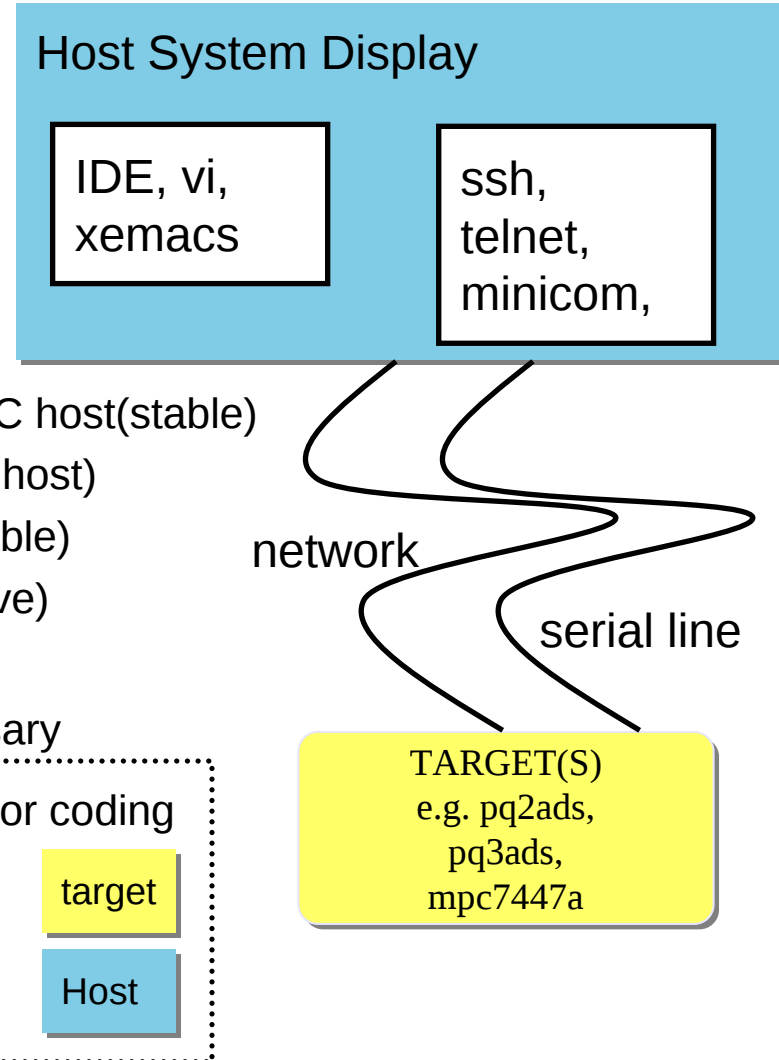


Slide 4

Freescale Semiconductor Confidential Proprietary. Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2004

Typical usage

- **Beginning**
 - Login to host system
 - Boot Linux on target
- **Development, repeat as needed**
 - Host shell/IDE: edit source
 - Host: test/verify executables on PPC host(stable)
 - Target: Load binaries (via NFS from host)
 - Target: test binaries on target (unstable)
 - Target debug tools (on host hard drive)
- **Logoff and go home**
 - Remote access from home if necessary



Concept of Host and Target

Host system (Pegasos)

- running Linux
- large hard disk
- complete tool chain
 - compilers
 - JTAG software
- graphic console

Target system (evaluation/engineering sample boards)

- little initial software
- serial or network console

Host (Pegasos) requirements

Debian Linux distribution pre-installed packages

- GNU tool chain
- TFTP server
- NFS Portmap server
- DHCP server
- Minicom, a serial console emulator
- Target root filesystem

Hardware requirements

- Large hard drive (40GB+)
- Lots of RAM (256MB+)
- Fast processor, e.g. MPC744x at 1GHz+
- Fast or GigE network interface

Target requirements

Firmware/BootROM (e.g. DINK, U-boot)

- download capability: TFTP client
- serial console
- execute a user program
- Pass platform information to the OS

Hardware

- Ethernet interface
- serial port
- RAM

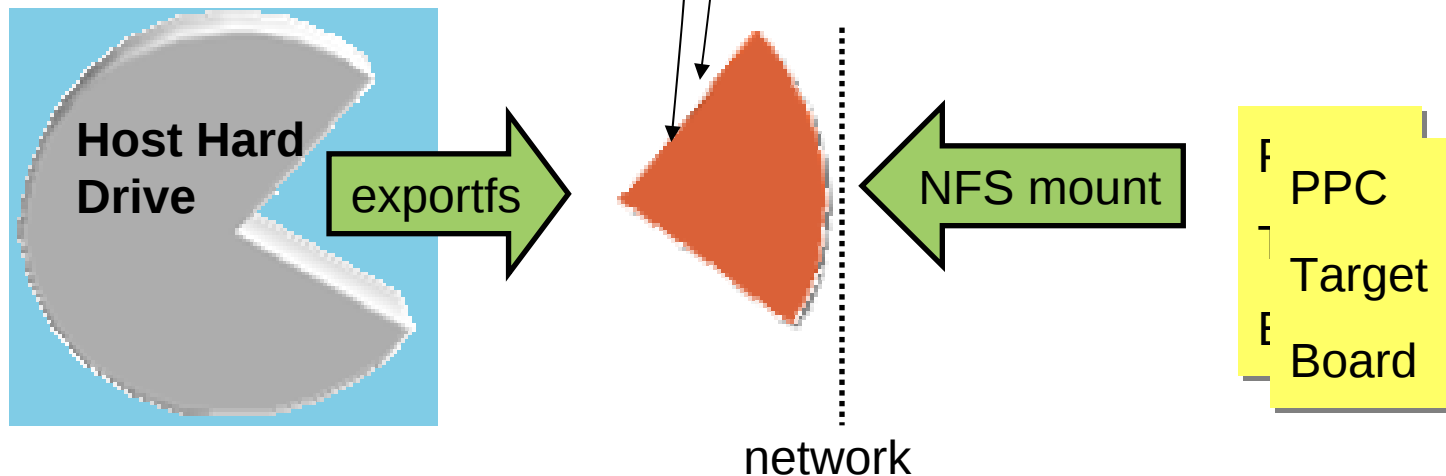
Pegasos Host set-up

Target root partition (preinstalled embedded root file system)

- `/opt/ppc_82xx`
- `/opt/ppc_74xx`

NFS server exports

- Edit `/etc/exports`, add the following line for the target
`/opt/ppc_82xx 10.82.144.199/255.255.255.0(rw,no_root_squash,sync)`
 syntax: root directory, target IP address/target netmask



Set up Pegasos as root NFS server

Activate new NFS service on the host

- **export new directories**

```
debian:/etc# exportfs -r
/home/jacob 10.82.117.93
/opt/ppc_82xx 10.82.144.199/255.255.255.0
/opt/ppc_74xx 10.82.116.254/255.255.252.0
```

- **restart NFS, portmap**

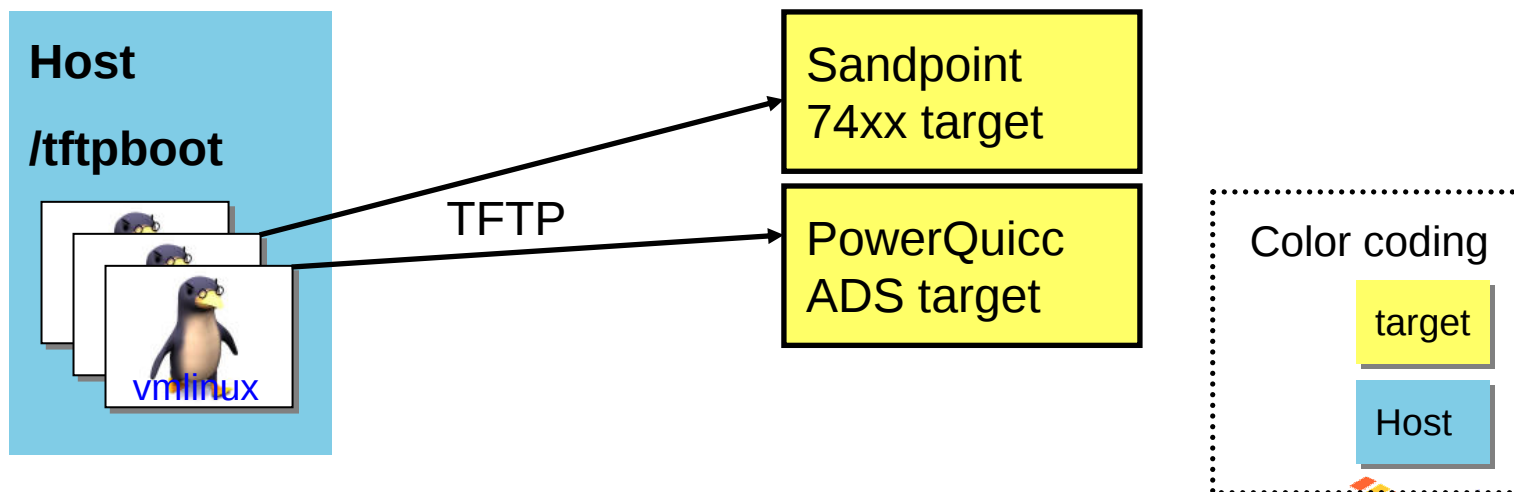
```
debian:root/targets# cat nfsupdate.sh
#This script is to be executed each time after /etc/exports is updated
/etc/rc2.d/S20nfs-kernel-server stop
/etc/rc2.d/S18portmap stop
/etc/rc2.d/S18portmap start
/etc/rc2.d/S20nfs-kernel-server start
```

Host setup procedure (cont'd)

- Prepare bootable kernel images for the targets
 - copy vmlinuz to [/tftpboot](#)
- configure and start TFTP server in [/etc/inetd.conf](#)

#:BOOT: TFTP service is provided primarily for booting. Most sites
 # run this only on machines acting as "boot servers."

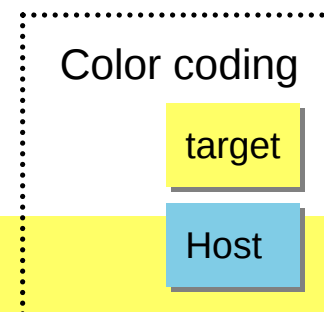
```
tftp      dgram  udp    wait  nobody  /usr/sbin/tcpd  /usr/sbin/in.tftpd /tftpboot
```



Set up target boot parameter (PQIII ADS)

Start in U-Boot, show, set, and save parameters with

- **printenv**
- **setenv**
- **saveenv**



```
gatewayip=10.82.1.254
netmask=255.255.254.0
serverip=10.82.144.201
ipaddr=10.82.144.199
bootfile=uImage
bootcmd=tftpboot; bootm 1000000
rootpath=/opt/ppc_82xx
bootargs=root=/dev/nfs rw nfsroot=10.82.144.201:/opt/ppc_82xx
ip=10.82.144.199:10.82.144.201:10.82.144.254:255.255.255.0:pq3_ads:eth0:off
console=ttyS0,115200
hostname=pq3_ads
```

Target start-up basic steps

1. Target send TFTP read request (RRQ) to the host, start downloading vmlinuz from host /tftpboot/
2. Target start executing at vmlinuz/zImage entry point
3. Target Linux kernel boot wrapper starts on target, parse boot options (e.g. root NFS)
4. Target Linux kernel mounted NFS root to the host exported directory
5. Target executes /sbin/init program from the host hard disk target (e.g. /opt/ppc_74xx)

Target startup (PQIII ADS)

U-Boot console response

```
TFTP from server 10.82.144.201; our IP address is
10.82.144.199
Kernel command line: root=/dev/nfs rw
nfsroot=10.82.144.201:/opt/ppc_82xx
ip=10.82.144.199:10.82.144.201:10.82.144.254:255.255.255.0:pq3
_ads:eth0:off console=ttyS0,115200
```

Target startup (PQIII ADS)

Target mounted root file system via NFS

```
...
IP-Config: Complete:
    device=eth0, addr=10.82.144.199, mask=255.255.255.0,
gw=10.82.144.254,
    host=pq3_ads, domain=, nis-domain=(none),
    bootserver=10.82.144.201, rootserver=10.82.144.201,
rootpath=
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
Looking up port of RPC 100003/2 on 10.82.144.201
Looking up port of RPC 100005/1 on 10.82.144.201
VFS: Mounted root (nfs filesystem).
Freeing unused kernel memory: 248k init
INIT: version 2.78 booting
```

Target start-up (Sandpoint 74xx)

- **Sandpoint start up in DINK**
- **No need to set environment variables**
- **Download bootable kernel image**
 - DINK>> dl -nw -b -fzlimage.elf
 - DINK>> go 110000
- **At boot loader prompt assign boot option;**
 - root=/dev/nfs rw nfsroot=10.82.144.201:/opt/ppc_74xx
 - ip=10.82.144.199:10.82.144.201:10.82.144.254:255.255.255.0:sandpoint:eth0:off
 - console=ttyS0,115200

LAB DEMO

- Sandpoint with MPC7447A Gyrus PMC
- PQIII ADS

Exercise #1: Serial console

- Run **minicom -s** setup COM port parameters

Filenames and paths

File transfer protocols

Serial port setup //step #1 choose /dev/ttyS1, baud rate, etc.

Modem and dialing

Screen and keyboard

Save setup as dfl // step #2, optional

Save setup as..

Exit

Exit from Minicom

- Change permission for non-root user to access /dev/ttyS1 (preferred)
 - **chmod 644 /dev/ttyS1**

Exercise #2: Inspect target and host status

```

jacob@debian: /home/jacob
bash-2.05# cat /proc/cpuinfo
processor       : 0
cpu            : e500
revision      : 1.0 (pvr 8020 0010)
bogomips      : 226.09
Vendor        : Motorola SPS
Machine       : mpc8560ads
bus freq      : 792.000000 MHz
PVR           : 0x80200010
SVR           : 0x80700010
PLL setting   : 0x6
Memory        : 128 MB
bash-2.05# cal
      January 1970
Su Mo Tu We Th Fr Sa
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

CTRL-A Z for help l115200 8N1 | NOR | Minicom 1.83.1 | VT102 | Online 00:09

jacob@debian: ~/home/jacob
drwxr-xr-x  7 root   root   4096 Apr  1 13:33 proj
drwxr-xr-x  2 root   root   4096 May 21 14:47 root
drwxr-xr-x  2 root   root   4096 Mar 15 20:31 sbin
drwxrwxrwt  2 root   root   4096 May 20 21:11 tmp
drwxr-xr-x 14 root   root   4096 Mar 22 16:52 usr
drwxr-xr-x 10 root   root   4096 Mar 15 20:31 var
jacob@debian:~$ cat /proc/cpuinfo
processor       : 0
cpu            : 7457, altivec supported
clock          : 999MHz
revision      : 1.1 (pvr 8002 0101)
bogomips      : 997.37
machine       : CHRP Pegasos2
jacob@debian:~$ cal
      May 2004
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
jacob@debian:~$

```

target console

Host console

Exercise #2: Target and host file system

The screenshot shows a Minicom terminal window with two panes. The top pane is labeled 'target console' and the bottom pane is labeled 'Host console'. A large green arrow points from the target console to the host console, indicating a comparison or mapping of file systems.

target console

```

jacob@debian: /home/jacob
bash-2.05# gcc
gcc: No input files
bash-2.05# whereis dhry21static
dhry21static: /usr/bin/dhry21static
bash-2.05# ls -l /
total 56
drwxr-xr-x  2 root  root    4096 Mar 15  2004 bin
drwxr-xr-x  2 root  root    4096 May 24  2004 dev
-rw-r--r--  1 root  root    175 May 21  2004 dhry.res
drwxr-xr-x 19 root  root    4096 May 24  2004 etc
drwxr-xr-x  2 root  root    4096 Mar 15  2004 images
drwxr-xr-x  3 root  root    4096 May 21  2004 jacob
drwxr-xr-x  4 root  root    4096 Mar 15  2004 lib
drwxr-xr-x  2 root  root    4096 Mar 22  2004 mnt
dr-xr-xr-x 25 root  root      0 Dec 31 19:00 proc
drwxr-xr-x  7 root  root    4096 Apr  1  2004 proj
drwxr-xr-x  2 root  root    4096 May 21  2004 root
drwxr-xr-x  2 root  root    4096 Mar 15  2004/sbin
drwxrwxrwt  2 root  root    4096 May 20  2004 tmp
drwxr-xr-x 14 root  root    4096 Mar 22  2004 usr
drwxr-xr-x 10 root  root    4096 Mar 15  2004 var
CTRL-A Z for help l11 N1 | NOR | Minicom 1.83.1 | VT102 | Online 00:07

```

Host console

```

jacob@debian: /home#
# /etc/exports: the access control list for filesystems which may be exported
# to NFS. See exports(5).
/home/jacob 10.82.117.93(rw,sync)
/opt/ppc_82xx 10.82.144.149/255.255.255.0(rw,no_root_squash)
/opt/ppc_74xx 10.82.116.254/255.255.252.0(rw,no_root_squash)
jacob@debian:~$ ls -l /opt/ppc_82xx
total 60
drwxr-xr-x  2 root  root    4096 Mar 15 20:30 bin
drwxr-xr-x  2 root  root    4096 May 24 21:20 dev
-rw-r--r--  1 root  root    175 May 21 20:23 dhry.res
drwxr-xr-x 19 root  root    4096 May 24 21:20 etc
drwxr-xr-x  2 root  root    4096 Mar 15 20:30 images
drwxr-xr-x  3 root  root    4096 May 21 14:38 jacob
drwxr-xr-x  4 root  root    4096 Mar 15 20:31 lib
drwxr-xr-x  2 root  root    4096 Mar 22 16:40 mnt
drwxr-xr-x  2 root  root    4096 Mar 15 20:31 proc
drwxr-xr-x  7 root  root    4096 Apr  1 13:33 proj
drwxr-xr-x  2 root  root    4096 May 21 14:47 root
drwxr-xr-x  2 root  root    4096 Mar 15 20:31/sbin
drwxrwxrwt  2 root  root    4096 May 20 21:11 tmp
drwxr-xr-x 14 root  root    4096 Mar 22 16:52 usr
drwxr-xr-x 10 root  root    4096 Mar 15 20:31 var
jacob@debian:~$

```

Exercise #3 Compile, execute, and debug a simple program or benchmark

1. Open two X terminals: `xterm -name term1`; `xterm -name term2`
2. In term1, use secure shell login to demo host Pegasos system:
`ssh -l faeguest xxx.xxx.xxx.xxx*`, use `faeguest` as password.
 - `cd /opt/ppc_82xx/root` for PQIII target
 - `cd /opt/ppc_74xx/root` for Sandpoint target
3. In term2, use telnet login to PQIII ADS or Sandpoint board,
`telnet xxx.xxx.xxx.xxx` login as `root`, no password asked

Exercise #3 Edit and compile a test program

```
#define ALTIVEC_ASM    asm volatile("vor 1,1,1")
#define SPE_ASM    asm volatile("evdivwu 1,1,1")
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    FILE * fp;
    char * line = NULL;
    size_t len = 0;
    ssize_t read;
    printf("\n\n-----\n");
    printf("\n2004 FAE Training test program, printing local CPU info \n");
    fp = fopen("/proc/cpuinfo", "r");
    if (fp == NULL)
        exit(EXIT_FAILURE);
    while ((read = getline(&line, &len, fp)) != -1) {printf("%s", line);}
    printf("-----\n");
    if (line)
        free(line);
    // ALTIVEC_ASM;
    return EXIT_SUCCESS;
}
```

```
debian:/opt/ppc_82xx/root# make
gcc -Wall -g -O0 whichcpu.c -o whichcpu
```

Execute the test program in term1(host)

```

debian:/opt/ppc_82xx/root# ./whichcpu
-----
2004 FAE Training test program, printing local CPU info
processor      : 0
cpu           : 7457, altivec supported
clock         : 999MHz
revision      : 1.1 (pvr 8002 0101)
bogomips      : 997.37
machine       : CHRP Pegasos2
-----
debian:/opt/ppc_82xx/root#

```

Execute the same test program in term2(target)

```
bash-2.05# hostname
pq3_ads
bash-2.05# ./whichcpu
```

```
-----
2004 FAE Training test program, printing local CPU info
```

```
processor      : 0
cpu          : e500
revision      : 1.0 (pvr 8020 0010)
bogomips      : 226.09
Vendor        : Motorola SPS
Machine       : mpc8560ads
bus freq      : 792.000000 MHz
PVR           : 0x80200010
SVR           : 0x80700010
PLL setting   : 0x6
Memory        : 128 MB
-----
```

Run benchmarks

- **DHRYSTONE 2.1**
 - /root/benchmarks/dhrystone/
- **LMBench3.0**
 - /root/benchmarks/lmbench-3.0-a3/bin/powerpc-linux-gnu/

Run dhrystone benchmark

```

root@pq3_ads:~# cd benchmarks/dhrystone
root@pq3_ads:~/benchmarks/dhrystone# ls
dhry.h      dhry21_static_libc231  dhry21b.c      libc-2.3.3.so  sptemp
dhry.res    dhry21_static_libc233  dhry21static   makefile       timers_b.c
dhry21      dhry21a.c              input
root@pq3_ads:~/benchmarks/dhrystone#
root@pq3_ads:~/benchmarks/dhrystone# ./dhry21
Dhrystone Benchmark, Version 2.1 (Language: C)
Please give the number of runs through the benchmark: 1000000
...

```

Run LMBench individual kernel

```
root@pq3_ads:~/benchmarks/lmbench-3.0-a3/bin/powerpc-linux-gnu# ls
bw_file_rd  lat_connect  lat_pmake  lat_usleep  loop_o
bw_mem      lat_ctx      lat_proc   lib_debug.o  memsize
bw_mmap_rd  lat_dram_page lat_rand    lib_mem.o    mhz
bw_pipe     lat_fcntl    lat_rpc     lib_stats.o  msleep
bw_tcp      lat_fifo     lat_select  lib_tcp.o    par_mem
bw_unix     lat_fs       lat_sem     lib_timing.o par_ops
cache       lat_http     lat_sig     lib_udp.o    stream
disk        lat_mem_rd   lat_syscall lib_unix.o   timing_o
enough      lat_mmap     lat_tcp     line         tlb
flushdisk   lat_ops      lat_udp     lmbench.a
getopt.o    lat_pagefault lat_unix    lmd
hello      lat_pipe     lat_unix_connect lmhttp
```

Run LMBench individual kernel (PQIII target)

```

root@pq3_ads:~/benchmarks/lmbench-3.0-a3/bin/powerpc-linux-gnu# ./line
32
root@pq3_ads:~/benchmarks/lmbench-3.0-a3/bin/powerpc-linux-gnu# ./mhz
779 MHz, 1.2837 nanosec clock
root@pq3_ads:~/benchmarks/lmbench-3.0-a3/bin/powerpc-linux-gnu# ./lat_ops
integer bit: 1.27 nanoseconds
integer add: 1.26 nanoseconds
integer mul: 5.05 nanoseconds
integer div: 44.19 nanoseconds
integer mod: 30.33 nanoseconds
int64 bit: 2.54 nanoseconds
int64 add: 2.53 nanoseconds
int64 mul: 10.25 nanoseconds
int64 div: 131.77 nanoseconds
int64 mod: 134.20 nanoseconds
float add: 1104.13 nanoseconds
float mul: 1158.97 nanoseconds
float div: 1567.72 nanoseconds
double add: 1015.87 nanoseconds
double mul: 1071.60 nanoseconds
double div: 1478.05 nanoseconds
float bogomflops: 7370.00 nanoseconds
double bogomflops: 6235.00 nanoseconds

```

Run LMBench individual kernel (host)

```

root@debian:/opt/ppc_82xx/root/benchmarks/lmbench-3.0-a3/bin/powerpc-linux-gnu# ./mhz
1000 MHz, 1.0000 nanosec clock
root@debian:/opt/ppc_82xx/root/benchmarks/lmbench-3.0-a3/bin/powerpc-linux-gnu# ./line
64
root@debian:/opt/ppc_82xx/root/benchmarks/lmbench-3.0-a3/bin/powerpc-linux-gnu#
./lat_ops
integer bit: 1.00 nanoseconds
integer add: 1.00 nanoseconds
integer mul: 2.00 nanoseconds
integer div: 23.01 nanoseconds
integer mod: 26.02 nanoseconds
int64 bit: 2.01 nanoseconds
int64 add: 4.41 nanoseconds
int64 mul: 6.11 nanoseconds
int64 div: 89.73 nanoseconds
int64 mod: 99.06 nanoseconds
float add: 5.09 nanoseconds
float mul: 5.00 nanoseconds
float div: 21.02 nanoseconds
double add: 5.17 nanoseconds
double mul: 5.00 nanoseconds
double div: 35.02 nanoseconds
float bogomflops: 48.05 nanoseconds
double bogomflops: 46.05 nanoseconds
root@debian:/opt/ppc_82xx/root/benchmarks/lmbench-3.0-a3/bin/powerpc-linux-gnu#

```

FAE75

Obtain/Install new packages

Maurie Ommerman, Jacob Pan, Phil
Brownfield

June 10, 2004

Freescal Semiconductor Confidential Proprietary. Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.
All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2004



Obtain and install new Debian packages

Advanced Package Tool (apt)

- **Where to get them?**
 - The /etc/apt/sources.list file
- **How to find them?**
 - Apt-cache search
- **How to install them?**
 - Apt-get install package_name
- **Run apt-get update each time after changing sources.list**
- **What if I am using proxy servers?**
 - Export http_proxy=http://usr_name:passwd@wwwproxy.mot.com:portnum
 - Export ftp_proxy=ftp://usr_name:passwd@wwwproxy.mot.com:portnum

Apt Example

```

kaibab:~# whoami
root
kaibab:~# apt-cache search a2ps
a2ps - GNU a2ps 'Anything to PostScript' converter and pretty-printer
a2ps-perl-ja - perl version of Miguel Santana's a2ps (support KANJI)
gpr - GUI for lpr: print files and configure printer-specific options
kaibab:~# apt-get install a2ps
Reading Package Lists... Done
Building Dependency Tree... Done
Suggested packages:
  tetex-bin acroread wdiff
Recommended packages:
  groff
The following NEW packages will be installed:
  a2ps
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 861kB of archives.
After unpacking 3752kB of additional disk space will be used.
Get:1 http://http.us.debian.org/stable/main/a2ps\_4.13b-16 [861kB]
Fetched 861kB in 4s (214kB/s)
Selecting previously deselected package a2ps.
(Reading database ... 118863 files and directories currently installed.)
Unpacking a2ps (from .../a2ps_4.13b-16_powerpc.deb) ...
Setting up a2ps (4.13b-16) ...

```

Using PMON module for code profiling

See Sergei's slides