H.A.L. and Real Time
Linux-1/2

Young Joon Lee
Supported by Dr. Book
February 6, 2002
Before and After

- **Before**
  - Windows NT 4.0, Service pack 4
  - Hyperkernel 2.3
  - M$ Visual C++ 5.0
  - Apache, PHP, MySQL
  - JAVA 1.1 GUI (need Java plug-in for Mac)

- **After**
  - Linux kernel 2.4.4 Redhat 7.1
  - RTLinux 3.1
  - Gcc 2.95.3
  - Apache, PHP, MySQL
  - JAVA 1.3 GUI (need Java plug-in for all Oses)
JAVA 1.1 GUI (AWT)
### Client program - Concept

<table>
<thead>
<tr>
<th>Log out</th>
<th>View Data</th>
<th>Get Data</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAVA Applet</td>
<td></td>
<td></td>
<td>Live Broadcasting (Need RealPlayer)</td>
</tr>
</tbody>
</table>
JAVA 1.3 GUI(Swing)-1
JAVA 1.3 GUI(Swing)-2
Client Program – Result

- New concepts
  - Using PHP
  - Can select which data will be plotted
  - Can see several data at the same plot
  - Can print the plot
The plot is $X$ or $Y$ versus time not $X$ versus $Y$.

<table>
<thead>
<tr>
<th>No.</th>
<th>X axis</th>
<th>Y axis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Desired</td>
<td>Measured</td>
</tr>
<tr>
<td>1</td>
<td>xdespos1.txt</td>
<td>xmeaspos1.txt</td>
</tr>
<tr>
<td>2</td>
<td>xdespos2.txt</td>
<td>xmeaspos2.txt</td>
</tr>
<tr>
<td>3</td>
<td>xdespos3.txt</td>
<td>xmeaspos3.txt</td>
</tr>
<tr>
<td>4</td>
<td>xdespos4.txt</td>
<td>xmeaspos4.txt</td>
</tr>
<tr>
<td>5</td>
<td>xdespos5.txt</td>
<td>xmeaspos5.txt</td>
</tr>
<tr>
<td>6</td>
<td>xdespos6.txt</td>
<td>xmeaspos6.txt</td>
</tr>
<tr>
<td>7</td>
<td>xdespos7.txt</td>
<td>xmeaspos7.txt</td>
</tr>
<tr>
<td>8</td>
<td>xdespos8.txt</td>
<td>xmeaspos8.txt</td>
</tr>
</tbody>
</table>
A new plot program (PHP)-2
A new server program

- Ordinary Linux program
- Rewritten in C
- Using TCP/IP
- Currently no GUI
  - In future version, it will have get GUI

- Need a program without networking
  - Users can use their own input directly
Linux—How to pronounce it?

- Linux is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world.

- How to pronounce it?
Real-time Linux

- **RTLinux**
  - Commercial version [http://www.fsmlabs.com](http://www.fsmlabs.com)
  - RTLinux began life as a research and open source project.

- **RTAI**
  - After RTLinux, it didn’t have some features. They make RTAI.
What should I know to program in RTLinux?

- **Linux programming**
  - Be familiar to Linux environment
    - Know how to copy, delete, make files and directories
    - Be accustomed to telnet environment
    - VI editor
  - C programming skill, don’t need to know c++
    - Gcc – Linux c compiler
- **Device driver programming**
  - Know the differences between the ordinary c program and device driver programming
What should I know to program in RTLinux (con.)

- Thread programming
  - Rtlinux program is composed of several threads of execution
  - POSIX thread programming skill is need

- RTLinux API
  - RTLinux API are well documented by paper and online help
Device driver program and ordinary program

```c
Int init_module(void) {
    ...
    ...}

Void cleanup_module(void)
```

You can load the RT modules and unload the RT modules. You don’t need to shut down the computer when you use RT tasks. You can use it dynamically.
A Distinguishing feature of Real Time System

- Deterministic – Deadlines and other assertion involving time are expressed in terms of **exact** or **fixed** values, rather than aggregate measures such as averages.
Deterministic of RTLinux

- A load with running Netscape
Time for sampling, process, output
Time for process, output
Flow for the entire process

- Data
- Server
- Homing
- Experiment
- Homing

Windows NT

Hyperkernel
New Flow for whole process

Data

TCP/IP Server → Save Data hal.conf → Generate Data
Move Data to Shared Memory

Experiments → Move Data to Shared Memory

Make Data files → Waiting → Plot result
By PHP

Linux

RTLinux

Linux
Threads in HAL

- Pump_on_thread
  - Turn on the pump. Wake start_experiment() up

- End_experiment_thread
  - Calculates time, and if the time is greater than Total_time+500, turn off the pump (for safety)

- Check_pump_thread
  - Check pump all the time independently.
  - When there is no experiment for PUMP_ON_DURATION, then turns off the pump automatically
Threads in HAL (continued)

- **Sampling_thread**
  - Read 3 ADC channels from 0 to 2
  - 0->Y axis, 1->X axis, 2->Load cell

- **Process_thread**
  - Calculate error
  - Calculate u

- **Output_thread**
  - Convert u to DAC value
  - Write 2 DAC channels from 0 to 2
  - 0->Y axis, 1->X axis
Functions in HAL

- **Init_servo_to_go()**
  - Initialize the ServoToGo card.
  - Run one time for whole experiment

- **Fifo1_handler()**
  - Check “/dev/rtf1” file.
  - “/dev/rtf1” is changed by a ordinary Linux program. Then fifo1_handler() is run.
  - Check pump status. If pump is turned off it wakes pump_on_thread on. If pump is turned on, execute start_experiment()
Functions in HAL (continued)

- **Intr_handler()**
  - Whenever IRQ 5 is generated by STG after Interrupt is on, this is run
  - Wakes experiment_thread up
  - Check current time to stop the pump and free the IRQ when current time $\geq$ Total_time

- **Start_experiment()**
  - Wake up the end_experiment_thread
  - Turn on the timer in STG
  - Request IRQ and makes it hard real time IRQ
  - Wake up check_pump_thread
What’s next...

- the HAL program in RTLinux
  - User input
  - PID controller (currently use P controller)
  - Calculate the desired position in RT thread

- Control the user
  - Use PHP and MySQL
  - Add, delete, and edit the user’s information by Web browser
  - Make, delete, and edit the experiment time on the web by students