H.A.L. and Real Time Linux

Young Joon Lee
Supported by Dr. Book
October 17, 2001
HAL-Real Image
What is HAL?

- HAL = Hydraulic Actuated Lifter
  - Is constructed to functionally resemble a fork lift truck
  - Use Hydraulic Power
  - Server and Client structure
    - JAVA Applet
    - TCP/IP socket programming
What is HAL? (continued.)

- Server Computer and Program
  - Pentium III, 640 MB RAM
  - Windows NT 4.0, service pack 4
  - Hyperkernel version 2.3
  - M$ Visual C++ 5.0
  - M$ Personal Webservers->Apache, PHP, MySQL

- Client Program
  - JAVA Applet
  - Netscape or Internet Explorer
Client Program – Login

Information For HAL

User ID is your acme email – gbxxxxxx@prism.gatech.edu
Password is what you know.

I.D. @prism.gatech.edu
Password

Download the program and manual to generate desired position

Maximum Values

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>2000</td>
<td>2500</td>
</tr>
<tr>
<td>I</td>
<td>200</td>
<td>1000</td>
</tr>
<tr>
<td>D</td>
<td>200</td>
<td>1000</td>
</tr>
<tr>
<td>Min. Position (mill-inch)</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Max. Position (mill-Inch)</td>
<td>45000</td>
<td>25000</td>
</tr>
</tbody>
</table>

System ID

When taking a System ID experiment, you cannot input any number in Gain fields.
P=1000, I=40, D=40 are assigned to each axis for System ID. Also there is a position feedback.
The experiment will last for 20 seconds.
Client Program–JAVA Applet
Client Program – Result
Client Program

■ Why JAVA applet?
  ■ Run the applet on every OS
  ■ Security is very strong
  ■ Do not need any other program except Web browser

■ JAVA or JAVA applet?
  ■ JAVA is a standalone program (JVM)
  ■ JAVA applet runs on the Web Browser
Server Program - Image
Server Program

- Windows NT
  - Is not deterministic
    - Interrupt service routines will be delayed by uncertain time intervals by Windows NT scheduler.

- Hyperkernel
  - Has its own scheduler and internal kernel
CPU time and memory resources are partitioned between Windows NT and Hyperkernel.

CPU time allotted for Windows NT and Hyperkernel is proportioned into alternating 250 microsecond segments.
Combined Program

Windows NT

Server Program

Shared Memory

Main Thread

Hyperkernel

ServoToGo Card
Flow for the entire process

Data → Server → Homing → Experiment → Homing

Windows NT

Hyperkernel
ME6404

- Introduction, square trajectory
- Developing the dynamic models
- Find the good P.I.D. gains
- Input shaping
Real Time?

- Must not only produce the correct answer or output, but it must also compute the answer “on time”
- \( \Rightarrow \) A deadline – A limit on the absolute or relative time when a computation must complete

- A qualitative distinction
  - Hard Real Time
  - Soft Real Time
Hard RT and Soft RT

- Hard Real Time
  - Meeting the deadline is absolutely critical (eg. Rocket control)

- Soft Real Time
  - Occasionally allowed to miss deadlines (eg. Updating a video display)
Why Real Time OS?

- Timesharing systems try to optimize average case but RT systems must consider worst case
- What is good for the average case tends to deteriorate the worst case

- Real Time OS
  - VxWorks
  - QNX
  - ATX
  - RTLinux
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.
POSIX?

- Portable Operating System Interface
  - A specification which dictates how operating systems should behave
  - Basic operations involving signaling and threads, making it easier for programmers to port their applications from one operating system to the other.
- POSIX 1003.13/PSE51.
  - Defined as a “minimal realtime system” environment.
Process? & Thread?

- A process – contains both an executing program and a bundle of resources such as the file descriptor table and address space.
- A thread – a sequential flow of control through a program.
Difference between Threads and Processes

- Multi-threaded programming is a form of parallel programming where several threads of control are executing concurrently in the program.

- All threads execute in the same memory space, and can therefore work concurrently on shared data.

- Multi-processes run in their own memory space.
Linux

■ Why?
  ■ It’s free even if using as commercial purposes
  ■ Can use lots of programs
  ■ X-Window System is better than Windows NT
  ■ Does not crash as many as M$ Windows OS
  ■ You don’t need to install the programs again when you upgrade the Linux
X-Windows
RTLinux

- A small POSIX 1003.13/PSE51-compatible hard realtime operating system that runs Linux as its lowest priority thread.
- The worst case time between the moment a hardware interrupt is detected by the processor and the moment an interrupt handler starts to execute is under 15 microseconds on RTLinux running on a generic x86.
- A RTLinux periodic task runs within 35 microseconds of its scheduled time on the same hardware.
It extends the standard Linux programming environment to realtime problems.

RTLinux realtime interrupt handlers and tasks can be connected to ordinary Linux processes—either via a device interface where Linux processes read/write data, or via shared memory.

A standard Linux process can collect data from a realtime handler or task, process and log it and display the results on X-Windows.
RTLinux (continued)

Interrupt control hardware

Real-Time Kernel

Real-Time tasks

Real-Time Fifos or Shared Memory

Linux

Linux process
# Client program - Concept

<table>
<thead>
<tr>
<th>Log out</th>
<th>View Data</th>
<th>Get Data</th>
<th>HELP</th>
</tr>
</thead>
</table>

| JAVA Applet | Live Broadcasting (Need RealPlayer) |
Client program - Concept

View Data

- Using PHP
- Can select which data will be plotted
- Can see several data at the same plot
- Can print the plot
Server GUI

- Tcl/Tk
  - Run this program on almost every OS
  - Execute the TCP/IP server program
  - Does not relate with RT task
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
RTL/Linux
Linux
What I should know

- RTLinux (Module programming)
- C/C++ (Multi-threaded programming)
- JAVA
- Socket programming
- Tcl/Tk
- PHP
- MySQL
- RealSystem Server and RealSystem Producer
- HTML