

Haptic Systems

530-655

Mohsen Mahvash
Lecture 4
12/1/06

Real-Time Programming for Haptic Rendering

- Pseudo code for a simple haptic simulation
- Haptic rendering rate
- Real-time haptic rendering
- Free real-time operating systems
- MS Windows and Haptic Rendering
- OpenHaptics Toolkit and HDAPI
- References

Pseudo code for a simple haptic simulation

inialization()

while(1){

$[\theta_1, \theta_2, \theta_3] = \text{readEncoders}()$

$p = \text{tipPosition}(\theta_1, \theta_2, \theta_3)$

//Virtual Environment

$F = f(p, c, t)$

$\tau = J^T F$

$I_i = \frac{\tau_i}{k_i}$

applyCurrents()

rtTaskWaitAPeriod()

}

Haptic rendering rate

- ❖ Haptic rendering should be performed at a high fixed rate (1000Hz)
- ❖ The high rate generation of the force samples can locate the harmoni components out of either the haptic device bandwidth or touch sensitivity
- ❖ High rate rendering provides stability

Free Real-time Operating Systems

RTAI Realtime Application Interface

<http://www.aero.polimi.it/~rtai/index.html>

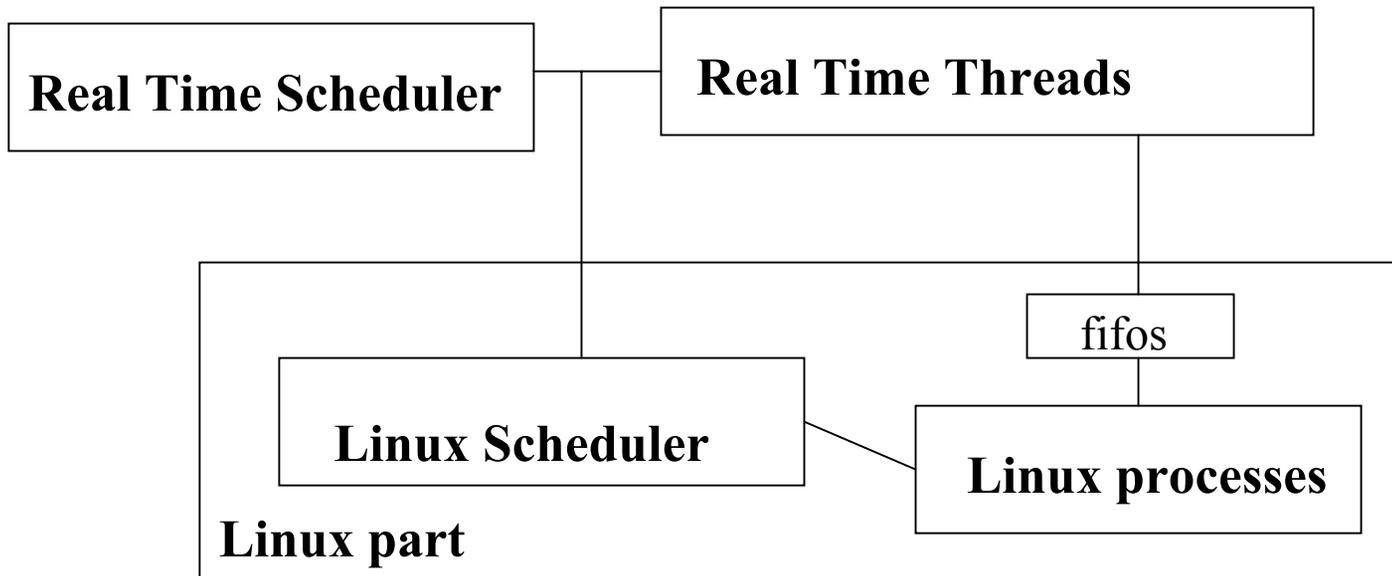
See **Beginner's Guide** and run examples

Also <http://www.rtai.org/>

Real Time Linux

<http://www.fsmlabs.com/>

See **Beginner's Guide** and run examples



Real time rendering with RTAI and RTL

- Write your Haptic thread in C, make a kernel module and add
- it to the real-time kernel by using command `insmod hapticThread`
- Write your graphic and user interaction programs in Linux by using C or C++
- Communicate data between your haptic thread and linux process by via FIFOS
- You can use FLTK and Fluid to build your user interaction menu and create a window for graphics
- Alternatively, you can use Java and Java3d instead of C++ instead of FLTK and OpenGL for graphics rendering and user interaction

MS Windows and Haptic Rendering

Write your program in Visual C as a process with several threads including a haptic thread and a graphic thread

Be careful about communicating data between threads

OpenHaptics Toolkit

HDAPI: A low level layer for haptic rendering, we use this one.

HLAPI: High level haptic rendering added to graphics applications

HDAPI

Two main components: device, scheduler

Typical use of HDAPI:

- Initialize the device and enable it
- Write your haptic rendering function and define it as a callback function
- Initialize the scheduler and start it
- Stop the scheduler and disable device when you are done

Data communication: Use a callback function to communicate data

Callback function: not invoked explicitly by the programmer,
another function that receives the callback function's address
call the function

References

3D Touch™ SDK OPEN Haptics™ TOOLKIT, Programming Guide,
<http://www.sensable.com/>

Tomorrow

Introduction to Teleoperation