

Rehabilitation Robot Software Documentation

Nicholas Berezny

October 12, 2018

1 Real Time Linux

1.1 Background

Real time systems are systems requiring that computations be made by strict time deadlines - in other words, they must be deterministic. Missing deadlines may result in damage to the system or to its environment. This category can be further subdivided into soft real-time and hard real-time. Hard real-time systems usually mathematically verify that deadlines will not be missed. For example, QNX is a hard real time OS. Soft real-time relaxes this condition, but still contains many of the features of a real-time system. Realtime Linux, for example.

Typical real time procedures include memory-locking, multithreading, setting priorities and schedulers, and testing latencies. Memory-locking ensures that no pagefaults occur during execution, which can cause significant delays. Multi-threading is a form of parallel computing that can speed up your process. Threads must have a priority set based on how critical they are to the functioning of the system. Thread execution is handled by the kernel, which delegates processing time according to the set scheduler (first-in first-out, round robin, etc.)

This software runs on a real-time enabled version of Linux which uses the PREEMPT_RT patch.

1.2 Installation

The following is an outline of the installation process for a PREEMPT_RT patched linux kernel with Ubuntu. Other linux flavours may also be used. More detailed instruction can be found at the Linux Foundation Website.

1. Download the linux kernel and the patch. The latest stable release of the patch is 4.14 (as of 11/10/2018)
2. Patch the kernel through the command line
3. Configure the kernel. Make sure to select "Fully Preemptible Kernel"
4. Install the kernel on a machine running Ubuntu

1.3 Libraries

2 Controller

2.1 Outline

2.2 Initialization

2.3 Controller

2.4 Communication

3 Node JS

3.1 Background

3.2 Installation

3.3 React

3.4 Web Sockets

4 UI Server

4.1 Outline

4.2 Server

4.3 UI

4.4 Communication

5 UI Server