Introduction

MutekH is a free and portable operating system for embedded platforms, ranging from micro-controller to multiprocessor systems. It's heavily used on multiprocessor platforms in various projects and was originally designed to natively support processors heterogeneity.

MutekH is very modular as it is ?exokernel based; it is composed of the Hexo hardware abstraction layer and the Mutek base kernel. The exokernel comes with several operating system interface libraries and services libraries. It was designed to be easily extended with new libraries and platform support, and allows development of kernel land and user land applications. It is fully configurable to match every application needs and platform constraints, moreover it has an easy to use build system and testsuite.

MutekH is currently used in several research projects, laboratories and universities and is actively developed. It was originally started at the ?SoC department of the ?LIP6 Laboratory (?UPMC in Paris). Some publications used MutekH as a supporting OS. The MutekhManifesto page gives more details about project goals.

It comes with tutorials and documentation.

Features

Architecture

MutekH is composed of the following major components: The Hexo hardware abstraction layer which deals with processor and platform abstraction, the Mutek kernel which offer usual hardware independent kernel base services, some operating system interface libraries supporting kernelland or userland applications, some services libraries and drivers. See modules list below for details. Processor specific and platform specific codes of the HAL are located in separate sub-modules.

Processor support

MutekH HAL has support for the following processors:

- Arm family of processors.
- Mips family of processors.
- PowerPc family of processors.
- Sparc family of processors.
- Intel x86 family of processors.
- Altera Nios2 soft core processors.
- Lattice LatticeMico32 soft core processors.
- Experimental support code for the Xilinx Microblaze soft core is available in the microblaze branch.

Platform support

MutekH HAL has built-in support for the following platforms:

- Soclib multiprocessor platforms with embedded software debugging features.
- Pc platform with x86 multiprocessor support, runs on real hardware or on emulators like qemu.
- Simple platforms with single processor (i.e. micro-controller platforms)

- Unix processes emulation which enables kernel and application to run embedded in Linux or OsX process(es).
- ?Aeroflex Gaisler platforms based on sparc leon processor.

Modules overview

Several modules are available:

- Base modules
  - Hexo HAL (hexo)
  - Mutek base kernel (mutek)
  - C library (libc)
  - Container library (gpct)
- Device drivers for various peripherals
- Parallel programming and Operating system interface libraries
  - Native Posix threads support (libpthread)
  - GNU OpenMP runtime library (libgomp)
  - A native implementation of ?Capsule parallel programming library. (libcapsule)
  - ?MutekS, a static OS for ?DSX SoC design tool (libsrl)
  - Unix library (development just started) (libunix)
- Major services libraries:
  - TPC/IP stack networking library (libnetwork)
  - File system support library (libvfs) along with file system drivers (FAT 16/32, ISO9660, RamFS, NFS)
  - ELF binary file format handling library (libelf)
  - Flattened device tree library (libfdt)
- Other useful libraries:
  - ?Lua scripting library (liblua)
  - ?Fdlibm standard math library (libm)
  - ?LibTermUI Ansi terminal driver and getline library (libtermui)
  - A simple cryptographic library (libcrypto)
  - ?TinyGL, a very small implementation of a subset of OpenGL (libtinygl)

Some ported applications

MutekH is used in various projects and as been used with many applications, some successfully ported and well known applications include:

- H264 video decoder (multiprocessor)
- MJPEG and Theora video decoder (multiprocessor)
- ?Splash benchmarks.
- ?Doom video game with network support
- Various applications using the ?Lua script engine
- Applications using TinyGL library.

Documentation

Getting started

Several documents are available to help you start using MutekH. You may also want to ?subscribe to the mutekh-users list.

- The Install page explains how to get the source code and install development tools.
- The BuildingExamples page briefly explains how to build example applications.

- The MutekH as Unix process quick start guide is a step by step guide to run MutekH embedded in a GNU/Linux or MacOs user process.

- The MutekH quick start guide for SoCLib is a step by step quide to run MutekH over a Soclib hardware simulator, intended for software developers.
- The MutekH/SoCLib tutorial is a step by step guide to write a simple MutekH application for a customizable Soclib multi-processor hardware simulator, intended for mixed software/hardware development.
- ?SoCLib provides a virtual machine image containing some sample platforms and applications based on MutekH.

- Porting your application is a step by step guide which show how to port the Berkley calculator (`bc`) Unix application on MutekH.
- Using the lua_microshell example

General documentation

- MutekH API reference manual

- Using the BuildSystem to build applications.
- Using Flattened device trees to describe hardware.
- Using MutekH on a AT91SAM7 Arm micro-controller based platform.

Kernel development

- Using the Testsuite.
- Using the BuildSystemDev to add modules and features.
- Writing header documentation for the API reference manual.
- Adding a driver, or adding a new driver class
- Usage of IntegerTypes in MutekH

Getting the source

Latest source code can be fetched from the ?Mercurial repository:

See Install page for detailed instructions.

Contact

- A mailing list is available for questions, announcements... You may freely ?subscribe here.
- A list of major contributors is available here.