

# Introduction

MutekH is a free and portable operating system for embedded platforms, ranging from micro-controller to multiprocessor systems.

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.

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 (Université Pierre et Marie Curie in Paris). A list of major contributors is available here.

The MutekhManifesto page gives more details about project goals.

## Software architecture

MutekH is composed of the following major components:

- The Hexo hardware abstraction layer which deals with processor and hardware platform abstraction and is capable of processor heterogeneity.
- The Mutek kernel which offer various usual kernel base features in a configurable manner.
- Some operating system interface libraries for kernel land or user land applications.
- Some kernel services libraries.
- Device and file system drivers.



## Platform support

MutekH currently supports the following platforms:

- Soclib multiprocessor platforms with Arm, Mips32 and PowerPc processor support.
- Pc platform with x86 multiprocessor support, runs natively or on emulators like qemu.
- Simple platforms with single processor (i.e. micro-controller platforms)
- Unix processes emulation which enables kernel and application to execute natively, embedded in Linux or OsX process(es).

## Getting started

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

- 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 BuildingExamples page briefly explains how to build example applications.
- 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.

- The [MutekH quick start guide for SoCLib](#) is a step by step guide to run MutekH over a complex SoCLib hardware simulator capable of processor heterogeneity, intended for kernel software developers.
- The [?SoCLib livecd](#) provides some sample platforms and applications based on older MutekH revisions.

More advanced topics are available:

- [Using MutekH on a AT91SAM7](#) Arm micro-controller based platform.

## Modules overview

Several modules are available:

- Base modules
  - ◆ Hexo HAL (hexo)
  - ◆ Mutek base kernel (mutek)
  - ◆ C library (libc)
  - ◆ Container library (gpct)
- Operating system interface libraries
  - ◆ Native Posix threads support (libpthread)
  - ◆ 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)
  - ◆ [Device drivers](#) for various peripherals
- Other useful libraries:
  - ◆ [?Lua](#) scripting library (liblua)
  - ◆ [?Fdlibm](#) standard math library (libm)
  - ◆ [?LibTermUI](#) Ansi terminal driver and getline library (libtermui)

## Applications

MutekH is used in various projects and but some successfully ported well known applications include:

- H264 video decoder (multiprocessor)
- MJPEG and Theora video decoder (multiprocessor)
- [?Splash](#) benchmarks.
- [?Doom](#) video game with network support
- Various application using the [?Lua](#) script engine

## Documentation

Quickstart and tutorial documents:

- See Getting started section above for start guides.
- [Porting your application](#)
- [Using MutekH on a AT91SAM7](#)
- [Using the lua microshell example](#)

Developers documentation:

- [MutekH API reference manual](#)
- Writing [header documentation](#) for the API reference manual.
- Using the [BuildSystem](#)
- [Adding a driver](#), or [adding a new driver class](#)
- using [Flattened device trees](#) to describe hardware.
- Usage of [IntegerTypes](#) in MutekH

## Getting the source

MutekH is being actively developed, no tarball releases are available yet.

Latest source code can be downloaded from the svn source tree:

```
svn co https://www.mutekh.org/svn/trunk/mutekh/
```

## Contact

- A mailing list is available for questions, announcements... You may freely [subscribe here](#).

## Developer resources

- [Buildbot status](#)