Dynamic Power Management Specification

From eLinux.org

Contents

Rationale

Many CE products are powered by batteries, not by a wired power supply. Making efficient use of the energy stored in the batteries is very important, both in terms of achieving an acceptable length of product usage per battery charge and in terms of product form factor; an energy-efficient product can be powered by a smaller sized battery than other would be required. This also has a bearing on the manufacturing cost of the product. Power savings can be achieved through the use of low-power system states, entered when the product is inactive or when selected explicitly by the user - for example, by use of the power switch to put the product into a standby state. This type of power management is referred to as static power management.

Dynamic power management refers to managing power while the product is in use, running user applications. Such power management can involve dynamic control of peripheral clocks and power supplies, varying the timer tick frequency during idle periods and CPU frequency/voltage scaling. This uses a combination of user-space and kernel-space software:

```
- a Policy Manager component running in user-space
- a Power Management Engine ("PM-Engine") component running in kernel-space
```

The software selects the system operating state based on a combination of the following:

```
- application requirements
- the parameters for battery lifetime or task deadlines
- CPU loading
- the current device constraints
- the current operating state
- user interaction
```

Specifications

If a platform has an interface to change its power parameters while platform is running, a CELF conforming kernel for the platform MUST satisfy the following:

1. A kernel SHOULD provide a userspace interface to add/remove/modify a power policy.

1. A kernel SHOULD provide interfaces to modify the platform's operating state and power parameters of the platform accordingly.

1. A kernel SHOULD have the capability to set the platform at the operating state of a task while the CPU is executing that task.

1. A kernel SHOULD have a userspace interface to initialize/remove/modify the operating state of a task.

1. A device driver framework MAY provide interface to register device constraints on device open and unregister the constraints on device close.

1. When a CPU is idle, a kernel SHOULD set the CPU at its low power mode if the CPU supports at least one low power mode.

1. System clocks MUST be correct regardless of power parameter changes.

Non-normative Notes

```
- Specific interface for an application to give hints on selecting an appropriate system operating state to the PM-Engine is not required by this specification.
- Recommendations for Hardware Platform
 - The latency of changing power modes of a CPU should be small enough to do it whenever a CPU becomes idle.
 - Memory address/data buses and peripheral buses can be put to low power modes when CPU becomes idle if the latency of putting those buses to low power modes are small enough.
 - CPU voltage/clock changing latency need to be quite small enough to do frequently while a task is running.
```

References

Power management framework - lower layer of PM-Engine:

1. IBM and Monta Vista Software, "Dynamic Power Management for Embedded Systems", Nov., 2002, http://www.research.ibm.com/arl/projects/papers/DPM_V1.1.pdf

1. "Dynamic Power Management", http://sourceforge.net/projects/dynamicpower/

Hooks required in tracking task scheduling etc.:

```
1.#3 "Kernel Hooks", http://www-124.ibm.com/developerworks/oss/linux/projects/kernelhooks/
```

Retrieved from "http://elinux.org/index.php?title=Dynamic_Power_Management_Specification&oldid=20968"

Category: Specification

- This page was last modified on 13 May 2010, at 21:03.
- This page has been accessed 12,167 times.
-