Power Management Strategies

Copyright: © 2017 Intel Corporation
Author: Rafael J. Wysocki <rafael.j.wysocki@intel.com>

The Linux kernel supports two major high-level power management strategies.

One of them is based on using global low-power states of the whole system in which user space code cannot be executed and the overall system activity is significantly reduced, referred to as :doc:`sleep states <sleep-states>`. The kernel puts the system into one of these states when requested by user space and the system stays in it until a special signal is received from one of designated devices, triggering a transition to the `working state` in which user space code can run. Because sleep states are global and the whole system is affected by the state changes, this strategy is referred to as the :doc:`system-wide power management <system-wide>`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\pm\[linux-master][Documentation][admin-guide][pm]strategies.rst`, line 15); *backlink*

Unknown interpreted text role "doc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\pm\[linux-master][Documentation][admin-guide][pm]strategies.rst`, line 15); *backlink*

Unknown interpreted text role "doc".

The other strategy, referred to as the :doc:`working-state power management <working-state>`, is based on adjusting the power states of individual hardware components of the system, as needed, in the working state. In consequence, if this strategy is in use, the working state of the system usually does not correspond to any particular physical configuration of it, but can be treated as a metastate covering a range of different power states of the system in which the individual components of it can be either `active` (in use) or `inactive` (idle). If they are active, they have to be in power states allowing them to process data and to be accessed by software. In turn, if they are inactive, ideally, they should be in low-power states in which they may not be accessible.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\pm\[linux-master][Documentation][admin-guide][pm]strategies.rst`, line 25); *backlink*

Unknown interpreted text role "doc".

If all of the system components are active, the system as a whole is regarded as "runtime active" and that situation typically corresponds to the maximum power draw (or maximum energy usage) of it. If all of them are inactive, the system as a whole is regarded as "runtime idle" which may be very close to a sleep state from the physical system configuration and power draw perspective, but then it takes much less time and effort to start executing user space code than for the same system in a sleep state. However, transitions from sleep states back to the working state can only be started by a limited set of devices, so typically the system can spend much more time in a sleep state than it can be runtime idle in one go. For this reason, systems usually use less energy in sleep states than when they are runtime idle most of the time.

Moreover, the two power management strategies address different usage scenarios. Namely, if the user indicates that the system will not be in use going forward, for example by closing its lid (if the system is a laptop), it probably should go into a sleep state at that point. On the other hand, if the user simply goes away from the laptop keyboard, it probably should stay in the working state and use the working-state power management in case it becomes idle, because the user may come back to it at any time and then may want the system to be immediately accessible.