NAME

uclampset − manipulate the utilization clamping attributes of the system or a process

SYNOPSIS

uclampset [options] [**−m** *uclamp_min*] [**−M** *uclamp_max] _command argument*

uclampset [options] [**−m** *uclamp_min*] [**−M** *uclamp_max*] **−p** *PID*

DESCRIPTION

uclampset sets or retrieves the utilization clamping attributes of an existing *PID*, or runs *command* with the given attributes.

Utilization clamping is a new feature added in v5.3. It gives a hint to the scheduler about the allowed range of utilization the task should be operating at.

The utilization of the task affects frequency selection and task placement. Only schedutil cpufreq governor understands handling util clamp hints at the time of writing. Consult your kernel docs for further info about other cpufreq governors support.

If you’re running on asymmetric heterogeneous system like Arm’s big.LITTLE. Utilization clamping can help bias task placement. If the task is boosted such that *util_min* value is higher than the little cores' capacity, then the scheduler will do its best to place it on a big core.

Similarly, if *util_max* is smaller than or equal the capacity of the little cores, then the scheduler can still choose to place it there even if the actual utilization of the task is at max.

Setting a task’s *uclamp_min* to a none zero value will effectively boost the task as when it runs it’ll always start from this utilization value.

By setting a task’s *uclamp_max* below 1024, this will effectively cap the task as when it runs it’ll never be able to go above this utilization value.

The full utilization range is: [0:1024]. The special value −1 is used to reset to system’s default.

OPTIONS

−m
Set *util_min* value.

−M
Set *util_max* value.

−a, **−−all−tasks**
Set or retrieve the utilization clamping attributes of all the tasks (threads) for a given PID.

−p, **−−pid**
Operate on an existing PID and do not launch a new task.

−s, **−−system**
Set or retrieve the system−wide utilization clamping attributes.

−R, **−−reset−on−fork**
Set **SCHED_FLAG_RESET_ON_FORK** flag.

−v, **−−verbose**
Show status information.

−h, **−−help**
Display help text and exit.

−V, **−−version**
Print version and exit.

USAGE

The default behavior is to run a new command:
uclampset *[−m uclamp_min] [−M uclamp_max] command* [*arguments*]

You can also retrieve the utilization clamping attributes of an existing task:
uclampset −p *PID*

Or set them:
uclampset −p *PID [−m uclamp_min] [−M uclamp_max]*

Or control the system−wide attributes:
uclampset −s *[−m uclamp_min] [−M uclamp_max]*

PERMISSIONS

A user must possess **CAP_SYS_NICE** to change the scheduling attributes of a process. Any user can retrieve the scheduling information.

NOTES

The system wide utilization clamp attributes are there to control the *allowed* range the tasks can use. By default both *uclamp_min* and *uclamp_max* are set to 1024. This means users can set the utilization clamp values for their task across the full range [0:1024].

For example:

```
uclampset −s −m 512 −M 700
```

will prevent any task from being boosted higher than 512. And all tasks in the systems are capped to a utilization of 700. Effectively rendering the maximum performance of the system to 700.

Consult your kernel docs for the exact expected behavior on that kernel.

AUTHORS

Qais Yousef <qais.yousef@arm.com>

SEE ALSO

nice(1), **renice**(1), **taskset**(1), **sched**(7)

See **sched_setscheduler**(2) and **sched_setattr**(2) for a description of the Linux scheduling scheme.

REPORTING BUGS

For bug reports, use the issue tracker at https://github.com/util−linux/util−linux/issues.

AVAILABILITY

The **uclampset** command is part of the util−linux package which can be downloaded from Linux Kernel Archive <https://www.kernel.org/pub/linux/utils/util−linux/>.