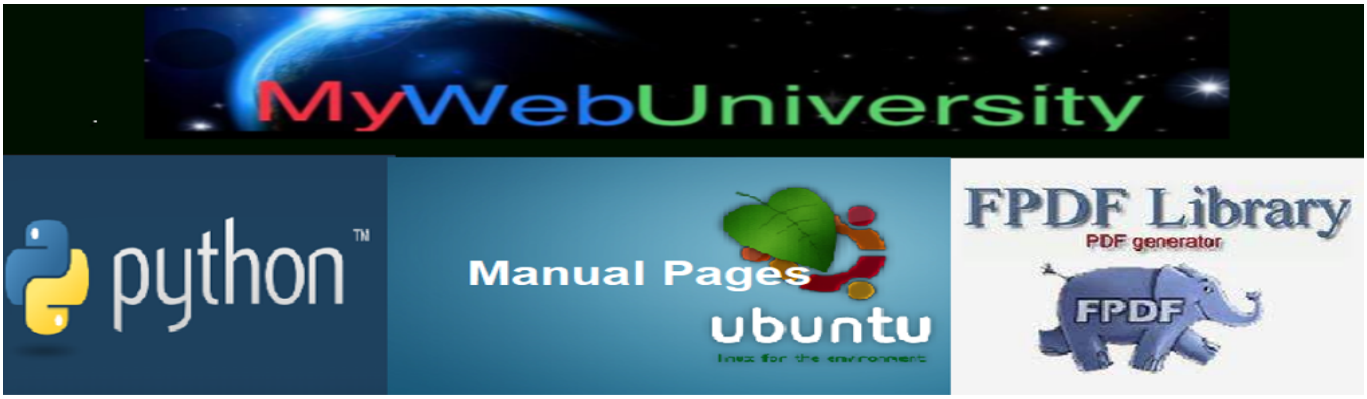

Rocky Enterprise Linux 9.2 Manual Pages on command 'chrt.1'

$ man chrt.1

CHRT(1) User Commands CHRT(1)

NAME

chrt - manipulate the real-time attributes of a process

SYNOPSIS

chrt [options] priority command argument ...

chrt [options] -p [priority] PID

DESCRIPTION

chrt sets or retrieves the real-time scheduling attributes of an existing PID, or runs command with the given attributes.

POLICIES

-o, --other

Set scheduling policy to SCHED_OTHER (time-sharing scheduling). This is the default Linux scheduling policy.

-f, --fifo

Set scheduling policy to SCHED_FIFO (first in-first out).

-r, --rr

Set scheduling policy to SCHED_RR (round-robin scheduling). When no policy is defined, the SCHED_RR is used as the default.

-b, --batch

Set scheduling policy to SCHED_BATCH (scheduling batch processes). Linux-specific, supported since 2.6.16. The priority argument has to be set to zero.

-i, --idle

Set scheduling policy to SCHED_IDLE (scheduling very low priority jobs).

Linux-specific, supported since 2.6.23. The priority argument has to be set to zero.

-d, --deadline

Set scheduling policy to SCHED_DEADLINE (sporadic task model deadline scheduling). Linux-specific, supported since 3.14. The priority argument has to be set to zero. See also --sched-runtime, --sched-deadline and --sched-period. The relation between the options required by the kernel is runtime ? deadline ? period. chrt copies period to deadline if --sched-deadline is not specified and deadline to runtime if --sched-runtime is not specified. It means that at least --sched-period has to be specified. See sched(7) for more details.

SCHEDULING OPTIONS

-T, --sched-runtime nanoseconds

Specifies runtime parameter for SCHED_DEADLINE policy (Linux-specific).

-P, --sched-period nanoseconds

Specifies period parameter for SCHED_DEADLINE policy (Linux-specific).

-D, --sched-deadline nanoseconds

Specifies deadline parameter for SCHED_DEADLINE policy (Linux-specific).

-R, --reset-on-fork

Use SCHED_RESET_ON_FORK or SCHED_FLAG_RESET_ON_FORK flag. Linux-specific, supported since 2.6.31.

Each thread has a reset-on-fork scheduling flag. When this flag is set, children created by fork(2) do not inherit privileged scheduling policies. After the reset-on-fork flag has been enabled, it can be reset only if the thread has the CAP_SYS_NICE capability. This flag is disabled in child processes created by fork(2).

More precisely, if the reset-on-fork flag is set, the following rules apply for subsequently created children:

? If the calling thread has a scheduling policy of SCHED_FIFO or SCHED_RR, the policy is reset to SCHED_OTHER in child processes.

? If the calling process has a negative nice value, the nice value is reset to zero in child processes.

OPTIONS

-a, --all-tasks

Set or retrieve the scheduling attributes of all the tasks (threads) for a given PID.

-m, --max

Show minimum and maximum valid priorities, then exit.

-p, --pid

Operate on an existing PID and do not launch a new task.

-v, --verbose

Show status information.

-V, --version

Display version information and exit.

-h, --help

Display help text and exit.

USAGE

The default behavior is to run a new command:

chrt priority command [arguments]

You can also retrieve the real-time attributes of an existing task:

chrt -p PID

Or set them:

chrt -r -p priority PID

PERMISSIONS

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

NOTES

Only SCHED_FIFO, SCHED_OTHER and SCHED_RR are part of POSIX 1003.1b Process Scheduling.

The other scheduling attributes may be ignored on some systems.

Linux' default scheduling policy is SCHED_OTHER.

AUTHORS

Robert Love <rml@tech9.net>, Karel Zak <kzak@redhat.com>

SEE ALSO

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

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

REPORTING BUGS

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

AVAILABILITY

The chrt 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/>.