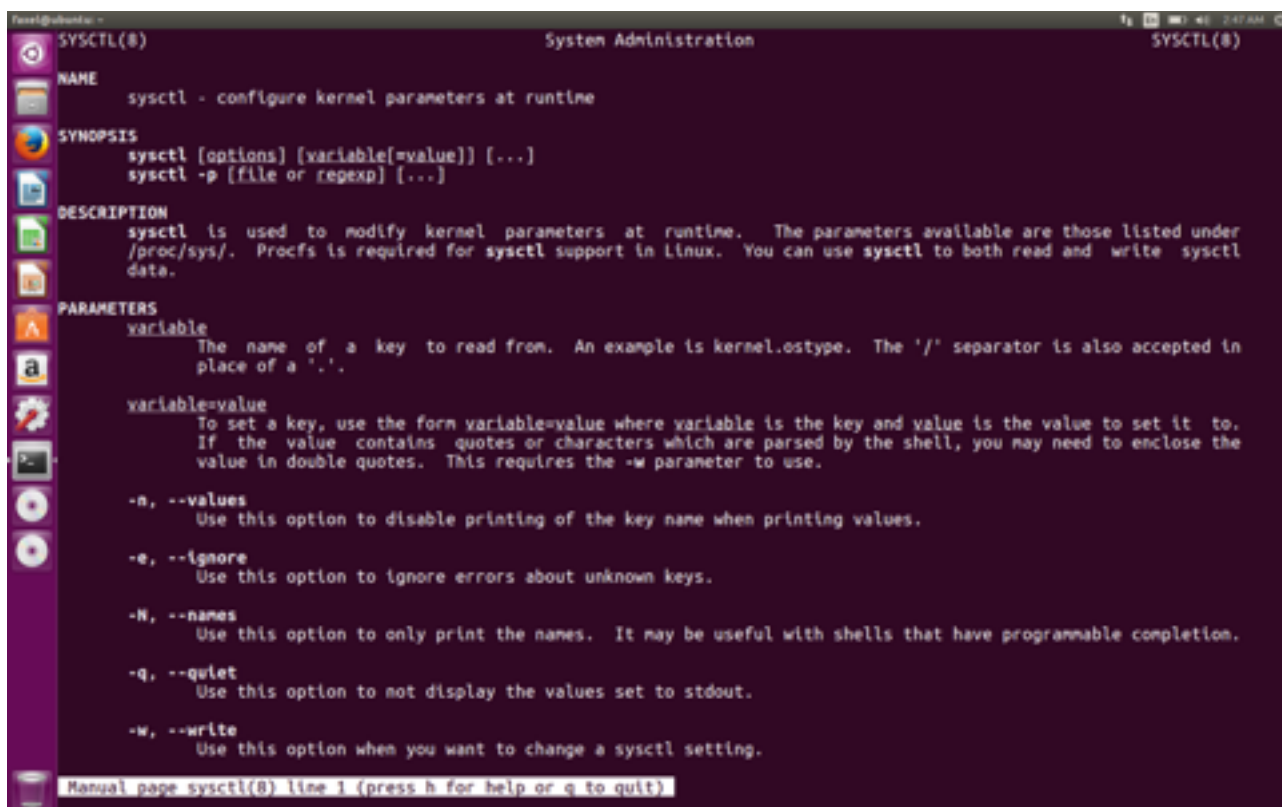


## کام اول بخش ۱

۱- به دست آوردن برش های زمانی زمان بند:

<https://doc.opensuse.org/documentation/leap/tuning/html/book.sle.tuning/cha.tuning.taskscheduler.html>

با توجه به منبع بالا می توان از sysctl استفاده کرد . manpage برای این دستور به شکل زیر است:



```
SYSCTL(8)                                System Administration                                SYSCTL(8)

NAME
    sysctl - configure kernel parameters at runtime

SYNOPSIS
    sysctl [options] [variable=value] [...]
    sysctl -p [file or regexp] [...]

DESCRIPTION
    sysctl is used to modify kernel parameters at runtime. The parameters available are those listed under
    /proc/sys/. Procfs is required for sysctl support in Linux. You can use sysctl to both read and write sysctl
    data.

PARAMETERS
    variable
        The name of a key to read from. An example is kernel.ostype. The '/' separator is also accepted in
        place of a '.'.

    variable=value
        To set a key, use the form variable=value where variable is the key and value is the value to set it to.
        If the value contains quotes or characters which are parsed by the shell, you may need to enclose the
        value in double quotes. This requires the -w parameter to use.

    -n, --values
        Use this option to disable printing of the key name when printing values.

    -e, --ignore
        Use this option to ignore errors about unknown keys.

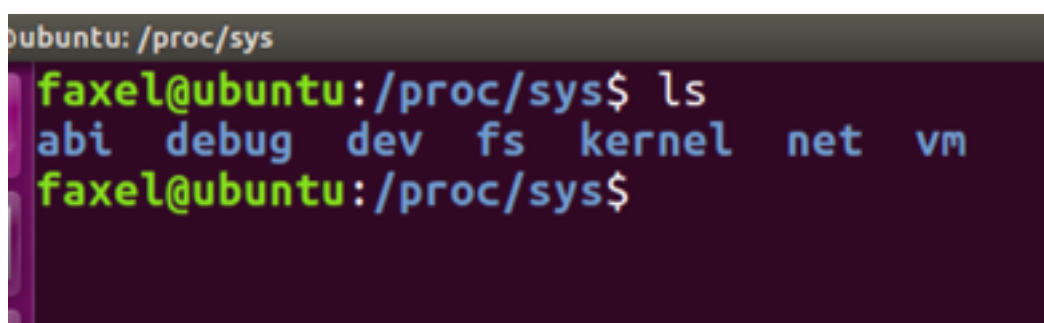
    -N, --names
        Use this option to only print the names. It may be useful with shells that have programmable completion.

    -q, --quiet
        Use this option to not display the values set to stdout.

    -w, --write
        Use this option when you want to change a sysctl setting.

Manual page sysctl(8) line 1 (press h for help or q to quit)
```

محتوای /proc/sys/ به شکل زیر است



```
ubuntu: /proc/sys
foxel@ubuntu:/proc/sys$ ls
abi debug dev fs kernel net vm
foxel@ubuntu:/proc/sys$
```

دستور زیر را وارد میکنیم:

```
sudo sysctl -A | grep "timeslice"
```

این دستور همه ی دایرکتوری های بالا را میگرد ( به علت وجود -A ) سپس هر خطی که در آن timeslice نوشته شده باشد را چاپ میکند:

```
fael@ubuntu: /proc/sys$ ls
abi debug dev fs kernel net vm
fael@ubuntu: /proc/sys$ sysctl -a |grep "timeslice"
No command 'sysctl' found, did you mean:
  Command 'sysctl' from package 'procps' (main)
sysctl: command not found
fael@ubuntu: /proc/sys$ sudo sysctl -a |grep "timeslice"
[sudo] password for fael:
sudo: sysctl: command not found
fael@ubuntu: /proc/sys$ sudo sysctl -a |grep "timeslice"
dev.parpport.default.timeslice = 200
kernel.sched_rr_timeslice_ms = 25
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s5.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
fael@ubuntu: /proc/sys$
```

همان طور که دیده میشود time slice برای RR به اندازه 25ms است .

دستور زیر اطلاعات بیشتری را نیز میدهد:

```
fael@ubuntu: /proc/sys$ sudo sysctl -a |grep "timeslice"
dev.parpport.default.timeslice = 200
kernel.sched_rr_timeslice_ms = 25
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s5.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
fael@ubuntu: /proc/sys$ sudo sysctl -A | grep "sched" | grep -v "domain"
grep: invalid argument 'omain' for '--directories'
Valid arguments are:
- 'read'
- 'recurse'
- 'skip'
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s5.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
fael@ubuntu: /proc/sys$ sudo sysctl -A | grep "sched" | grep -v "domain"
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s5.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
kernel.sched_autogroup_enabled = 1
kernel.sched_cfs_bandwidth_slice_us = 5000
kernel.sched_child_runs_first = 0
kernel.sched_latency_ns = 12000000
kernel.sched_migration_cost_ns = 500000
kernel.sched_min_granularity_ns = 1500000
kernel.sched_nr_migrate = 32
kernel.sched_rr_timeslice_ms = 25
kernel.sched_rt_period_us = 1000000
kernel.sched_rt_runtime_us = 950000
kernel.sched_shares_window_ns = 10000000
kernel.sched_time_avg_ns = 1000
kernel.sched_tunable_scaling = 1
kernel.sched_wakeup_granularity_ns = 2000000
fael@ubuntu: /proc/sys$
```

#### `sched_latency_ns`

Targeted preemption latency for CPU bound tasks. Increasing this variable increases a CPU bound task's timeslice. A task's timeslice is its weighted fair share of the scheduling period:

$$\text{timeslice} = \text{scheduling period} * (\text{task's weight} / \text{total weight of tasks in the run queue})$$

The task's weight depends on the task's nice level and the scheduling policy. Minimum task weight for a `SCHED_OTHER` task is 15, corresponding to nice 19. The maximum task weight is 88761, corresponding to nice -20.

Timeslices become smaller as the load increases. When the number of runnable tasks exceeds `sched_latency_ns/sched_min_granularity_ns`, the slice becomes `number_of_running_tasks * sched_min_granularity_ns`. Prior to that, the slice is equal to `sched_latency_ns`.

This value also specifies the maximum amount of time during which a sleeping task is considered to be running for entitlement calculations. Increasing this variable increases the amount of time a waking task may consume before being preempted, thus increasing scheduler latency for CPU bound tasks. The default value is 6000000 (ns).

#### `sched_child_runs_first`

A freshly forked child runs before the parent continues execution. Setting this parameter to 1 is beneficial for an application in which the child performs an execution after fork. For example `make -j<NO_CPUS>` performs better when `sched_child_runs_first` is turned off. The default value is 0.

#### `sched_time_avg_ms`

This parameter sets the period over which the time spent running real-time tasks is averaged. That average assists CFS in making load-balancing decisions and gives an indication of how busy a CPU is with high-priority real-time tasks.

The optimal setting for this parameter is highly workload dependent and depends, among other things, on how frequently real-time tasks are running and for how long.