Lab Presentation Kernel Arguments (Linux)

VE482 Team 1

Introduction

Configuration
Building
Booting
Runtime

References

Lab Presentation
Kernel Arguments (Linux)

VE482 Team 1

UM-SJTU Joint Institute

November 27, 2018

1 Introduction

2 Configuration
- Building
- Booting
- Runtime

3 References

What are kernel arguments?

Lab Presentation Kernel Arguments (Linux)

VE482 Team 1

Introduction

Configuration
Building
Booting
Runtime

References

Definition

kernel arguments, also known as Kernel parameters, are used to customize the behavior of OS.

There are three ways to pass options to the kernel and thus control its behaviour:

- When building the kernel.
- When starting the kernel (usually, when invoked from a boot loader).
- At runtime (through the files in /proc and /sys).

Kernel arguments list

Lab Presentation Kernel Arguments (Linux)

VE482 Team 1

Introduction

Configuration

Building

Booting

Runtime

References

parameter	Description
root=	Root filesystem.
rootflags=	Root filesystem mount options.
ro	Mount root device read–only on boot (default[1]).
rw	Mount root device read–write on boot.
initrd=	Specify the location of the initial ramdisk.
init=	Run specified binary instead of `/sbin/init` (symlinked to systemd in Arch) as init process.
init=/bin/sh	Boot to shell.
systemd.unit=	Boot to a specified target.
resume=	Specify a swap device to use when waking from hibernation.
nomodeset	Disable Kernel mode setting.
zswap.enabled	Enable Zswap.
panic=	Time before automatic reboot on kernel panic.
debug	Enable kernel debugging (events log level).
mem=	Force usage of a specific amount of memory to be used.
maxcpus=	Maximum number of processors that an SMP kernel will bring up during bootup.
selinux=	Disable or enable SELinux at boot time.
netdev=	Network devices parameters.
video=<videosetting>	Override framebuffer video defaults.

Notes

- Not all parameters are always available. Most are associated with subsystems and work only if the kernel is configured with those subsystems built in. They also depend on the presence of the hardware they are associated with.
- Parameters either have the format parameter or parameter=value.
- All kernel parameters are case-sensitive. Most of them are lower case, writing those in upper case does not work.

Configure on Building

Lab Presentation Kernel Arguments (Linux)
VE482 Team 1
Introduction
Configuration
Building
Booting
Runtime
References

The first way is to set the parameters before compilation. Kernel configuration is set in its .config file and properly set it can change the configuration.

More Advanced Configuration

- menuconfig, nconfig - menulized configuraiton
- xconfig, gconfig - GUI configuration which requires graphical support

On Booting

Lab Presentation Kernel Arguments (Linux)
VE482 Team 1
Introduction
Configuration
Building
Booting
Runtime
References

GRUB

- Press 'e' when the menu shows up and add them on the linux line:

```
linux /boot/vmlinuz-linux root=UUID=978e3e81
    -8048-4ae1-8a06-aa727458e8ff quiet splash
```

Press Ctrl+x to boot with these parameters.

GRUB

- To make the change persistent after reboot, while you could manually edit /boot/grub/grub.cfg with the exact line from above, the best practice is to:

  Edit /etc/default/grub and append your kernel options to the GRUB_CMDLINE_LINUX_DEFAULT line:

```
1 GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

  And then automatically re-generate the grub.cfg file with:

```
1 # grub-mkconfig -o /boot/grub/grub.cfg
```

In Runtime

Lab Presentation Kernel Arguments (Linux)

VE482 Team 1

Introduction

Configuration

Building

Booting

Runtime

References

Some kernel arguments can also be modified in runtime, and there are mainly two ways to do this.

Using procfs

Kernel arguments are saved as files in the proc file systems (which is always mounted at /proc in linux) directly write to the files can change the kernel arguments.

The basic step is using echo to change these files.

```
1    echo SOME_VALUE > /proc/sys/SOME_DIRECTORY/
         SOME_FILE
```

Using sysctl

sysctl is a command to configure kernel parameters. There are two ways to use it

- sysctl SOME_PARAMETER=SOME_VALUE
- Edit the configuration file (/etc/sysctl.conf) and run "sysctl -p"

References

- https://wiki.archlinux.org/index.php/Sysctl
- https://www.ibm.com/developerworks/library/l-proc/index.html
- https://wiki.archlinux.org/index.php/Kernel_parameters