

Linux Professional Institute Certification

LPI Leval1 Exam 101

LPI Leval1 Exam 102

目次

§ 1 システムアーキテクチャ	5
§ 1.1 BIOS	5
§ 1.2 デバイス	5
§ 1.2.1 デバイス確認	5
§ 1.2.2 デバイスドライバロード	8
§ 1.3 システムの起動	9
§ 1.3.1 システム起動の流れ	9
§ 1.3.2 システム起動時イベント	9
§ 1.3.3 init プロセス	10
§ 1.4 ランレベル	11

§ 1 システムアーキテクチャ

§ 1.1 BIOS

入出力基本システム (BIOS : Basic Input Output System)

コンピュータに接続されたディスクドライブ、キーボード、ビデオカードなどの周辺機器を制御するプログラム群のこと。

- OS を起動するためのプログラムをディスクから読み込んで実行
- デバイスの動作を設定
- 基本的な入出力を制御する

§ 1.2 デバイス

§ 1.2.1 デバイス確認

Linux カーネルが認識しているデバイスは /proc 以下で確認が可能 (図 1)。

```
[moriC@localhost ~]$ ls /proc
1      1284 1470 1545 1741 1827 1865 2063 228 33 849      execdomains  kpagecount  sched_debug  uptime
10     1285 1471 1560 1742 1833 1871 2066 23 336 9       fb           kpageflags  schedstat    version
1008   1289 1487 1563 1776 1834 1872 2067 24 38      ACPI        loadavg     scsi         vmallocinfo
1033   13   1495 1579 1784 1835 19   2069 245 4      asound      fs           locks        self         vmstat
1075   1326 15   16   1786 1838 1900 2071 246 40     buddyinfo   interrupts  mdstat       slabinfo     stat
11     1329 1506 162 1790 1841 1900 2072 25 41     bus         iomem       meminfo      softirqs     zoneinfo
1188   1349 1518 1649 18   1843 1937 2082 26 5      cgroups     ioports     misc         swaps
1199   1374 1526 1655 1810 1844 1949 2084 27 560    cmdline     ipmi        modules      sys
12     1382 1531 17   1814 1845 1987 2085 28 6      cpuinfo     irq         mounts        sysrq-trigger
1204   1387 1533 1700 1815 1848 1988 2086 29 7      crypto      kallsyms    mtd          sysvipc
1220   14   1535 1706 1810 1853 1989 21   3 72     devices     kcore       mtrr         timer_list
1237   141 1537 1712 1819 1855 1992 2111 30 8      diskstats   key-users   net          timer_stats
1250   142 1543 1723 1821 1857 2   22 31 802   dma         keys        pagetypeinfo timer_stats
1275   1463 1544 1733 1824 1861 20   226 32 803   driver      kmsg        partitions   tty
```

図 1: /proc の内容

ファイル名	情報
cpuinfo	CPU 情報
meminfo	メモリ情報
net	ネットワーク情報
mounts	マウントされたデバイス情報
partitions	パーティション情報
bus	bus 情報

表 1: /proc のファイル例

Linux はハードウェアへのアクセスを抽象化するデバイスファイルを持っており、これのデバイスファイルの読み書きを通してハードウェアにアクセスする。このとき、/dev ディレクトリに接続されているデバイスが認識されていなければ利用できない (図 2)。

```
[moriC@localhost ~]$ ls -l /dev | grep -e '^d' -e '^l'
```

lrwxrwxrwx. 1 root root	13	6月 18 19:42 2014	MAKEDEV -> /sbin/MAKEDEV
drwxr-xr-x. 2 root root	80	6月 18 19:42 2014	VolGroup
drwxr-xr-x. 2 root root	620	6月 18 19:42 2014	block
drwxr-xr-x. 2 root root	60	6月 18 19:42 2014	bsg
drwxr-xr-x. 3 root root	60	6月 18 19:42 2014	bus
drwxr-xr-x. 2 root root	2660	6月 18 19:42 2014	char
lrwxrwxrwx. 1 root root	11	6月 18 19:42 2014	core -> /proc/kcore
drwxr-xr-x. 3 root root	80	6月 18 19:42 2014	cpu
drwxr-xr-x. 5 root root	100	6月 18 19:42 2014	disk
lrwxrwxrwx. 1 root root	3	6月 18 19:42 2014	fb -> fb0
lrwxrwxrwx. 1 root root	13	6月 18 19:42 2014	fd -> /proc/self/fd
drwxr-xr-x. 2 root root	40	6月 18 19:42 2014	hugepages
drwxr-xr-x. 3 root root	220	6月 18 19:42 2014	input
drwxr-xr-x. 2 root root	100	6月 18 19:42 2014	mapper
drwxr-xr-x. 2 root root	60	6月 18 19:42 2014	net
drwxr-xr-x. 2 root root	0	6月 18 19:42 2014	pts
drwxr-xr-x. 2 root root	60	6月 18 19:42 2014	raw
lrwxrwxrwx. 1 root root	4	6月 18 19:42 2014	root -> dm-0
lrwxrwxrwx. 1 root root	4	6月 18 19:42 2014	rtc -> rtc0
drwxrwxrwt. 2 root root	140	6月 18 19:46 2014	shm
drwxr-xr-x. 3 root root	180	6月 18 19:42 2014	snd
lrwxrwxrwx. 1 root root	15	6月 18 19:42 2014	stderr -> /proc/self/fd/2
lrwxrwxrwx. 1 root root	15	6月 18 19:42 2014	stdin -> /proc/self/fd/0
lrwxrwxrwx. 1 root root	15	6月 18 19:42 2014	stdout -> /proc/self/fd/1
lrwxrwxrwx. 1 root root	4	6月 18 19:42 2014	systty -> tty0

図 2: /dev の内容

デバイスを確認するためのコマンドも用意されている。

コマンド名	情報
lspci	PCI デバイス情報 (図 3)
lsusb	USB デバイス情報 (図 4)
dmidecode	DMI 情報 (図 5)
fdesk -l	ディスクの空き容量情報
free -m	メモリの空き容量情報
bus	bus 情報

表 2: デバイス情報を確認できるコマンド例

```
[moriC@localhost ~]$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphics Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 00)
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)
```

図 3: lspci の内容

```
[moriC@localhost ~]$ lsusb
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

図 4: lsusb の内容

```
[root@localhost ~]# dmidecode --type bios
# dmidecode 2.11
SMBIOS 2.5 present.

Handle 0x0000, DMI type 0, 20 bytes
BIOS Information
    Vendor: innotek GmbH
    Version: VirtualBox
    Release Date: 12/01/2006
    Address: 0xE0000
    Runtime Size: 128 kB
    ROM Size: 128 kB
    Characteristics:
        ISA is supported
        PCI is supported
        Boot from CD is supported
        Selectable boot is supported
        8042 keyboard services are supported (int 9h)
        CGA/mono video services are supported (int 10h)
        ACPI is supported
```

図 5: dmidecode の内容

§ 1.2.2 デバイスドライバロード

デバイスドライバ (device driver)

デバイスを利用するために必要な制御プログラムで必要なデバイスドライバを取り込むことを「ロードする」という。基本的には自動的にロードされる。

lsmod

ロードされているカーネルモジュールを確認するコマンド (図 6)

```
[moriC@localhost ~]$ lsmod
Module                Size  Used by
fuse                  73530  0
autofs4               26513  3
8021q                 25349  0
garp                  7152   1 8021q
stp                   2218   1 garp
llc                   5546   2 garp,stp
ipt_REJECT            2351   2
nf_conntrack_ipv4     9506   2
nf_defrag_ipv4        1483   1 nf_conntrack_ipv4
iptable_filter        2793   1
ip_tables             17831   1 iptable_filter
ip6t_REJECT           4628   2
nf_conntrack_ipv6     8748   2
nf_defrag_ipv6       11182   1 nf_conntrack_ipv6
xt_state              1492   4
nf_conntrack          79758   3 nf_conntrack_ipv4,nf_conntrack_ipv6,xt_state
ip6table_filter       2889   1
ip6_tables            18732   1 ip6table_filter
ipv6                  317340  35 ip6t_REJECT,nf_conntrack_ipv6,nf_defrag_ipv6
uinput                7992   0
ppdev                 8537   0
microcode             112685  0
parport_pc            22690   0
```

図 6: lsmod の内容

modprobe

手でデバイスドライバをロードするコマンド

§ 1.3 システムの起動

§ 1.3.1 システム起動の流れ



図 7: システム起動の流れ

1. BIOS ではハードウェアのチェックや初期化を実行し、ディスクのブートセクタを読み出しブートローダへ制御を移す。
2. ブートローダでは ハードディスクからカーネルをメモリへ読み込む。
3. カーネルではメモリの初期化やシステムクロックの設定を実行し、init プログラム実行へ移す。
4. init プログラムでは初期化スクリプトを実行してランレベルに応じて指定したデーモンを起動する。

§ 1.3.2 システム起動時イベント

dmesg

システム起動時の処理を確認できるコマンド。カーネルが出力したメッセージを一時的に蓄えておくバッファの内容を表示する。/var/log/message のログファイルにシステム起動時の状況が保存されている。(図 8)

```

[moriC@localhost ~]$ dmesg
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 2.6.32-431.el6.x86_64 (mockbuild@c6b8.bsys.dev.centos.org) (gcc version 4.4.7
20120313 (Red Hat 4.4.7-4) (GCC) ) #1 SMP Fri Nov 22 03:15:09 UTC 2013
Command line: ro root=/dev/mapper/VolGroup-lv_root rd_NO_LUKS rd_NO_MD rd_LVM_LV=VolGroup/l
v_swap crashkernel=128M KEYBOARDTYPE=pc KEYTABLE=jp106 rd_LVM_LV=VolGroup/lv_root LANG=ja_
JP.UTF-8 rd_NO_DM rhgb quiet
KERNEL supported cpus:
  Intel GenuineIntel
  AMD AuthenticAMD
  Centaur CentaurHauls
BIOS-provided physical RAM map:
BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
BIOS-e820: 0000000000100000 - 000000007fff0000 (usable)
BIOS-e820: 000000007fff0000 - 0000000080000000 (ACPI data)
BIOS-e820: 00000000ffff0000 - 0000000100000000 (reserved)
  
```

図 8: dmesg の内容 (抜粋)

§ 1.3.3 init プロセス

init プロセス

Linux システムで最初に行われるプロセスで、`/etc/inittab` ファイル (図 9) の設定を読み込み、システムに必要なサービスを順次起動していく。一般的な UNIX 系 OS では SysVinit が実行される。

```
[moriC@localhost ~]$ cat /etc/inittab
# inittab is only used by upstart for the default runlevel.
#
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# System initialization is started by /etc/init/rcS.conf
#
# Individual runlevels are started by /etc/init/rc.conf
#
# Ctrl-Alt-Delete is handled by /etc/init/control-alt-delete.conf
#
# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/serial.conf,
# with configuration in /etc/sysconfig/init.
#
# For information on how to write upstart event handlers, or how
# upstart works, see init(5), init(8), and initctl(8).
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

図 9: inittab の内容

SysVinit

SystemV init の略。UNIX SystemV で採用された起動の仕組み。あらかじめ決められた順位サービスを起動していくため、サービスの起動時間によって全体の起動時間が変動する。

1. `/etc/inittab` ファイルを読み込む。
2. `/etc/rc.sysinit` スクリプトを実行。
3. `/etc/rc` スクリプトを実行。
4. `/etc/rc` スクリプトが `/etc/rc(ランレベル).d` ディレクトリ以下のスクリプトを実行。

Upstart

イベント駆動型の init デーモンで、ブート時のタスクの起動とシャットダウン時のタスクの停止を非同期に行い、同時にシステム動作中の管理を行う。

systemd

ソケット起動型サービスとバス起動型サービスを組み合わせた init デーモンで、相互に依存したサービス群をより並列的に起動できる。

§ 1.4 ランレベル