

LPICレベル1技術解説無料セミナー

2017年4月22日

NPO法人 LPI-Japan
Technology Director 和田 真輝
masaki.wada@lpi.or.jp



本日の内容

1. LPICレベル1試験の概要説明

- a. Linuxとは？
- b. LPICとは？
- c. LPICレベル1とは？

2. 主題101:システムアーキテクチャの説明

- a. ハードウェア設定の決定と構成
- b. システムのブート
- c. ランレベル/ブートターゲットの変更と
システムのシャットダウンまたはリブート

3. 主題101の例題解説



本日の内容

1. LPICレベル1試験の概要説明

- a. Linuxとは？
- b. LPICとは？
- c. LPICレベル1とは？

2. 主題101:システムアーキテクチャの説明

- a. ハードウェア設定の決定と構成
- b. システムのブート
- c. ランレベル/ブートターゲットの変更と
システムのシャットダウンまたはリブート

3. 主題101の例題解説



Linuxとは？

* はじまり

- ・ 1991年、フィンランドの学生だった^{リーナス トーバルズ}Linus TorvaldsがLinuxカーネル（OSの基本機能を担う核となるソフトウェア）を公開。

* 自由なソフトウェア

- ・ GPL v2ライセンス
- ・ ソースコードが公開

* フリーウェア

- ・ 無償で利用可能なものが多い
（CentOS, Ubuntu, Fedora, Debianなど）

* オープンな開発体制（コミュニティ）

- ・ 世界中の企業、開発者が開発に参加



ディストリビューション

* ディストリビューション

- ・ Linuxカーネルだけだと動作しません。Linuxカーネルに様々なソフトウェアをまとめ、パッケージとしたものをインストールして利用します。その配布の形態をディストリビューションと呼びます。

* 主なLinuxディストリビューション

Debian系	Red hat系	Slackware系
・ Debian GNU/Linux ・ Ubuntu	・ Fedora ・ CentOS	・ Slackware ・ openSUSE

補足：Linuxとは元々OSの核となるカーネルを指します。そのため、狭義のLinuxではLinuxはカーネルを指します。広義のLinuxでは、Linuxカーネルに様々なソフトウェアをまとめ、パッケージとしたものを指します。現在、Linuxと言うと広義のLinuxを指す場合が多いです。



Linuxの幅広い活用分野

Linuxが圧倒的に強い分野



家電機器



GPS



テレビ



デジカメ



ハードディスク
レコーダー



アンドロイド Tizen
スマートホン

組込みOS



- ・パソコン
- ・旧オフコン市場

Linuxが圧倒的に強い分野



ネット系
サーバー



- ・スーパーコンピュータ
- ・クラウドサーバー
- ・ミッションクリティカルな
分野でのサーバー
- ・メインフレーム



ロボティクス



本日の内容

1. LPICレベル1試験の概要説明

- a. Linuxとは？
- b. LPICとは？**
- c. LPICレベル1とは？

2. 主題101:システムアーキテクチャの説明

- a. ハードウェア設定の決定と構成
- b. システムのブート
- c. ランレベル/ブートターゲットの変更と
システムのシャットダウンまたはリブート

3. 主題101の例題解説



**200カ国以上で
53万人以上が受験**

世界最大規模のLinux資格試験



LPICの高い評価

『資格取り方選び方全ガイド 2017年』（高橋書店）



「ジャンル別受験者数ランキング
コンピュータ部門」にて、
LPICが1位に選ばれました！



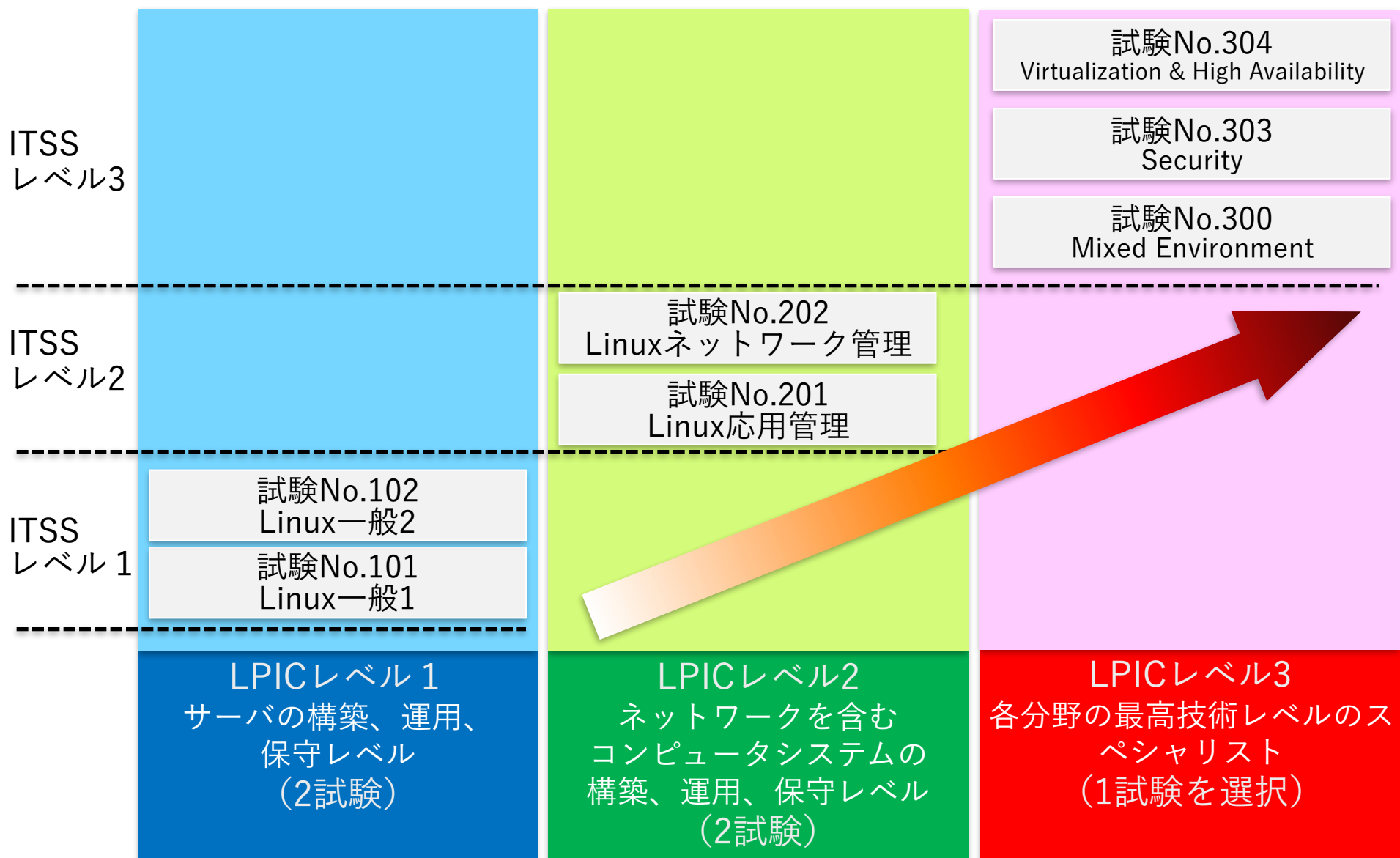
LPIC国内受験者総数

2017年3月末現在



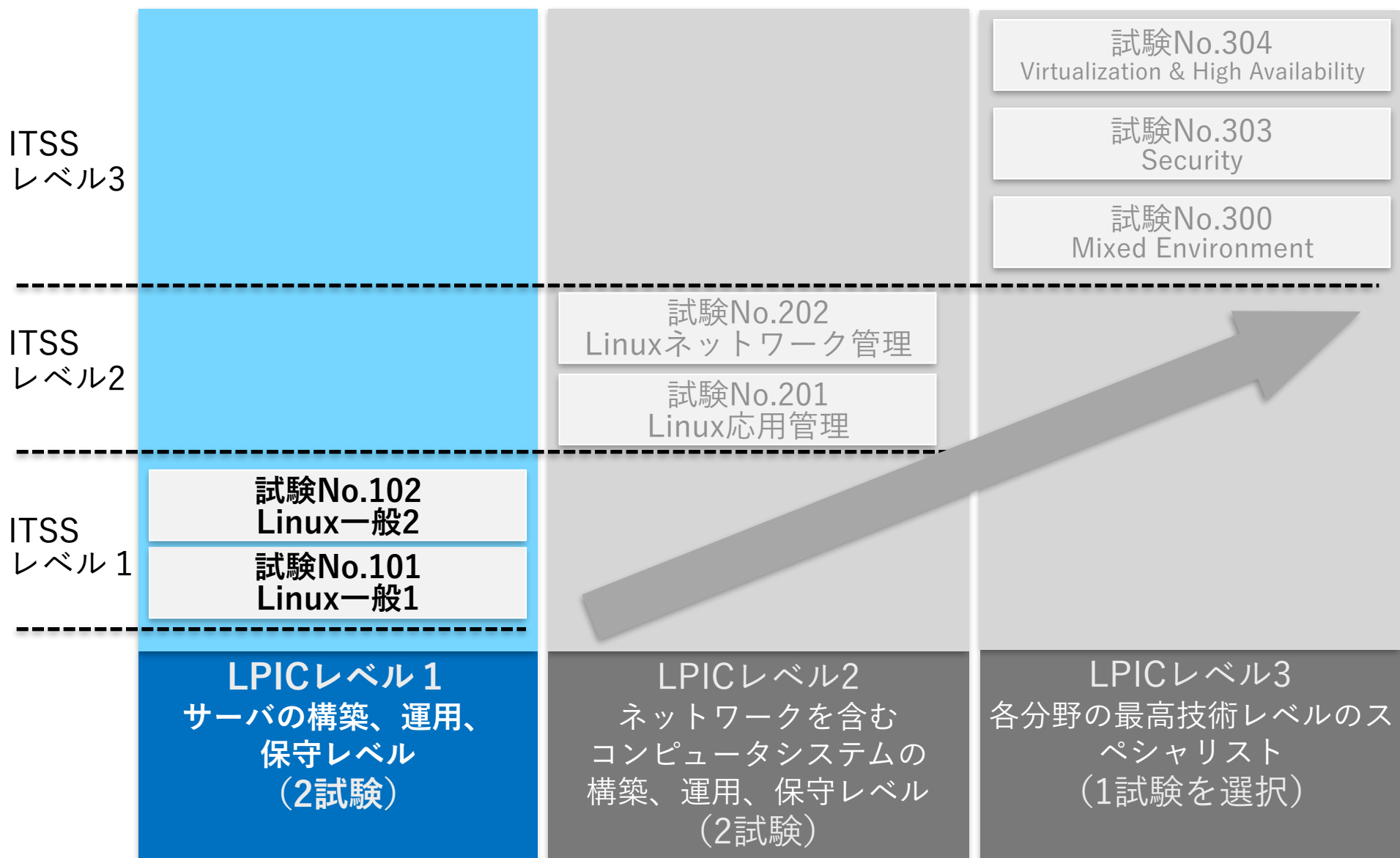


LPIC試験の構成と概要





LPIC試験の構成と概要





本日の内容

1. LPICレベル1試験の概要説明

- a. Linuxとは？
- b. LPICとは？
- c. LPICレベル1とは？**

2. 主題101:システムアーキテクチャの説明

- a. ハードウェア設定の決定と構成
- b. システムのブート
- c. ランレベル/ブートターゲットの変更と
システムのシャットダウンまたはリブート

3. 主題101の例題解説



「ファーストレベルLinux専門家」を認定する資格試験

- * 基礎的なLinux操作技術のスキル指標を確認できます。
 - ・ 実務で必要なLinuxの基本操作とシステム管理が行える。
 - ・ Linuxサーバの構築・運用・保守に必要な操作とシステム管理ができる。
 - ・ Linuxディストリビューションを利用するために必要な知識がある。
- * 資格の活かせる主な職種

Linuxサーバ環境の構築・運用・保守

インフラエンジニア

ネットワークエンジニア

SE営業職

セキュリティエンジニア

Linuxや組込み系のソフトウェア開発業務

データセンター構築

IT研修インストラクター



2試験の合格により認定

101試験

主題101：システムアーキテクチャ

主題102：Linuxのインストールとパッケージ管理

主題103：GNUとUnixのコマンド

主題104：デバイス、Linuxファイルシステム、ファイルシステム階層標準

102試験

主題105：シェル、スクリプト、およびデータ管理

主題106: ユーザインターフェイスとデスクトップ

主題107：管理業務

主題108：重要なシステムサービス

主題109：ネットワークの基礎

主題110：セキュリティ



2試験の合格により認定

101試験

主題101：システムアーキテクチャ

主題102：Linuxのインストールとパッケージ管理

主題103：GNUとUnixのコマンド

主題104：デバイス、Linuxファイルシステム、ファイルシステム階層標準

102試験

主題105：シェル、スクリプト、およびデータ管理

主題106: ユーザインターフェイスとデスクトップ

主題107：管理業務

主題108：重要なシステムサービス

主題109：ネットワークの基礎

主題110：セキュリティ



本日の内容

1. LPICレベル1試験の概要説明

- a. Linuxとは？
- b. LPICとは？
- c. LPICレベル1とは？

2. 主題101:システムアーキテクチャの説明

- a. ハードウェア設定の決定と構成
- b. システムのブート
- c. ランレベル/ブートターゲットの変更と
システムのシャットダウンまたはリブート

3. 主題101の例題解説

基本的なシステムハードウェア

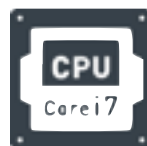
出力装置

モニタなど



入力装置

キーボード、マウスなど



CPU

コンピュータの制御(頭脳)



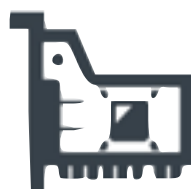
メモリ

データを一時的に記憶



ハードディスク

データを保存



拡張カード

コンピュータの機能を拡張

コールドプラグデバイス:電源を切って抜き差し



USB

キーボード、マウス、メモリ

ホットプラグデバイス:電源を切らずに抜き差し

補足：デバイス：コンピュータシステムに接続するハードウェア機器・装置をデバイスと呼びます。

ハードウェアを制御する基本の入出力プログラム（ファームウェア）です。

＊ 役割

- ・ OSを起動するためのプログラム(**ブートローダ**)の呼び出し。
- ・ ハードウェアの設定。

＊ 主な設定可能項目

- ・ ブートデバイスの順序
- ・ ハードウェアの使用・不使用
- ・ 時刻の設定 など



補足：ファームウェア

ハードウェアを直接制御するためのソフトウェアです。BIOSは、CPUやメモリーなどを接続する基盤であるメイン（マザー）ボードに事前に組み込まれています。



Linuxから見たハードウェア

Linuxは、ハードウェアをデバイスファイルという形で扱います。

* デバイスファイルの場所と作成

- ・ **/dev**ディレクトリにあり、**udev**によって動的に作成されます。
- ・ なお、udevの設定は、**/etc/udev/rules.d**ディレクトリにあります。

```
[root@localhost ~]# ls /dev/
```

```
autofs      core        fuse        mapper      oldmem      sda
stderr      tty13       tty21       tty3         tty38       tty46       tty54       tty62       uhid        vcs4        vcsa6
block       tty13       cpu         hidraw0      mcelog      port        sda1
stdin       tty14       tty22       tty30       tty39       tty47       tty55       tty63       uinput      vcs5        vfio
bsg         tty14       cpu_dma_latency hpet        mem         ppp         sda2
stdout      tty15       tty23       tty31       tty4         tty48       tty56       tty7        urandom     vcs6        vga_arbiter
tty11       tty2        tty28       tty36       tty44       tty52       tty60       ttyS2       vcs2        vcsa4
console     full        loop-control nvram        rtc0        sr0
tty12       tty20       tty29       tty37       tty45       tty53       tty61       ttyS3       vcs3        vcsa5
.
.
```

(省略)

補足：“ls”コマンド

ファイルやディレクトリの情報を表示する際に利用するコマンドです。



デバイス情報の確認

デバイス情報は、/procディレクトリ配下のファイルで確認できます。

- ・ 主な/procファイル

/proc/cmdline	ブートローダのパラメータ
/proc/cpuinfo	CPUの情報
/proc/interrupts	割り込みの情報
/proc/meminfo	メモリの情報

- ・ デバイス情報の確認方法

```
[root@localhost ~]# cat /proc/cpuinfo
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 78
model name    : Intel(R) Core(TM) m7-6Y75 CPU @ 1.20GHz
.
.
(省略)
```

補足：“cat”コマンド

ファイルの内容を表示するコマンドです。



デバイス情報は、コマンドでも確認することができます。

- ・ コマンド

lspci	PCIで接続されたデバイスのリストを表示
lsusb	USBで接続されたデバイスのリストを表示

- ・ 実行例 (lspci)

```
[root@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)
.
.
(省略)
```

- ・ 実行例 (lsusb)

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



デバイスドライバ情報の確認

デバイスを制御するためのプログラムをデバイスドライバと呼びます。

- ・ コマンド

lsmod	カーネルモジュールのリストを表示
modprobe	カーネルモジュールをロード/アンロードする

- ・ 実行例 (lsmod)

```
[root@localhost ~]# lsmod
Module                Size  Used by
ip6t_REJECT           4372   2
nf_conntrack_ipv6     7985   3
nf_defrag_ipv6        26468  1 nf_conntrack_ipv6
xt_state              1492   3
nf_conntrack          79537  2 nf_conntrack_ipv6,xt_state
ip6table_filter        2889   1
```

- ・ 実行例 (modprobe : usb-storageのロード(取込み))

```
[root@localhost ~]# modprobe usb-storage
```

- ・ 実行例 (modprobe : usb-storageのアンロード(取外し))

```
[root@localhost ~]# modprobe -r usb-storage
```



本日の内容

1. LPICレベル1試験の概要説明

- a. Linuxとは？
- b. LPICとは？
- c. LPICレベル1とは？

2. 主題101:システムアーキテクチャの説明

- a. ハードウェア設定の決定と構成
- b. システムのブート**
- c. ランレベル/ブートターゲットの変更と
システムのシャットダウンまたはリブート

3. 主題101の例題解説



システムのブート(起動)手順

電源ON

電源をONにするとBIOSが起動します。

BIOS

BIOSは、ハードウェアの初期化を行い、ディスクからブートローダを起動します。

ブートローダ

ブートローダは、ディスクからLinuxカーネルをメモリへロード(読み込み)します。

Linuxカーネル

Linuxカーネルは、SysVinitまたはsystemdを起動します。

SysVinit/systemd

SysVinitまたはsystemdは、各種サービスを起動します。

ログイン画面

起動が完了し、ログイン画面が表示されます。



ブート時のイベントを確認（ログファイル）

ブート時のイベントは、ログファイルで確認できます。

- ・ `/var/log/dmesg`

```
[root@localhost ~]# cat /var/log/dmesg
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Initializing cgroup subsys cpuacct
[    0.000000] Linux version 3.10.0-514.6.1.el7.x86_64
(builder@kbuilder.dev.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-11)
(GCC) ) #1 SMP Wed Jan 18 13:06:36 UTC 2017
```

（省略）

- ・ `/var/log/messages`

```
[root@localhost ~]# cat /var/log/messages
Apr  9 21:06:01 localhost rsyslogd: [origin software="rsyslogd" swVersion="7.4.7" x-
pid="863" x-info="http://www.rsyslog.com"] rsyslogd was HUPed
Apr  9 21:22:45 localhost rsyslogd: [origin software="rsyslogd" swVersion="7.4.7" x-
pid="864" x-info="http://www.rsyslog.com"] start
Apr  9 21:22:35 localhost kernel: Initializing cgroup subsys cpuset
Apr  9 21:22:35 localhost kernel: Initializing cgroup subsys cpu
Apr  9 21:22:35 localhost kernel: Initializing cgroup subsys cpuacct
Apr  9 21:22:35 localhost kernel: Linux version 3.10.0-514.6.1.el7.x86_64
```

（省略）



ブート時のイベントを確認（コマンド）

ブート時のイベントはコマンドでも確認できます。

- ・ コマンド

dmesg	カーネルが出力したメッセージを表示
-------	-------------------

- ・ 実行例

```
[root@localhost ~]# dmesg
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Initializing cgroup subsys cpuacct
[    0.000000] Linux version 3.10.0-514.6.1.el7.x86_64
(builder@kbuilder.dev.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-11)
(GCC) ) #1 SMP Wed Jan 18 13:06:36 UTC 2017
[    0.000000] Command line: BOOT_IMAGE=/vmlinuz-3.10.0-514.6.1.el7.x86_64 root=/dev/
mapper/centos-root ro crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap
rhgb quiet LANG=ja_JP.UTF-8
.
.
(省略)
```

補足：“dmesg”コマンドと”/var/log/dmesg”ログ

dmesgコマンドは、/var/log/dmesgの中身を表示しています。



本日の内容

1. LPICレベル1試験の概要説明

- a. Linuxとは？
- b. LPICとは？
- c. LPICレベル1とは？

2. 主題101:システムアーキテクチャの説明

- a. ハードウェア設定の決定と構成
- b. システムのブート
- c. ランレベル/ブートターゲットの変更と
システムのシャットダウンまたはリブート

3. 主題101の例題解説



システムのブート(起動)手順

電源ON

電源をONにするとBIOSが起動します。

BIOS

BIOSは、ハードウェアの初期化を行い、ディスクからブートローダを起動します。

ブートローダ

ブートローダは、ディスクからLinuxカーネルをメモリへロード(読み込み)します。

Linuxカーネル

Linuxカーネルは、SysVinitまたはsystemdを起動します。

SysVinit/systemd

SysVinitまたはsystemdは、各種サービスを起動します。

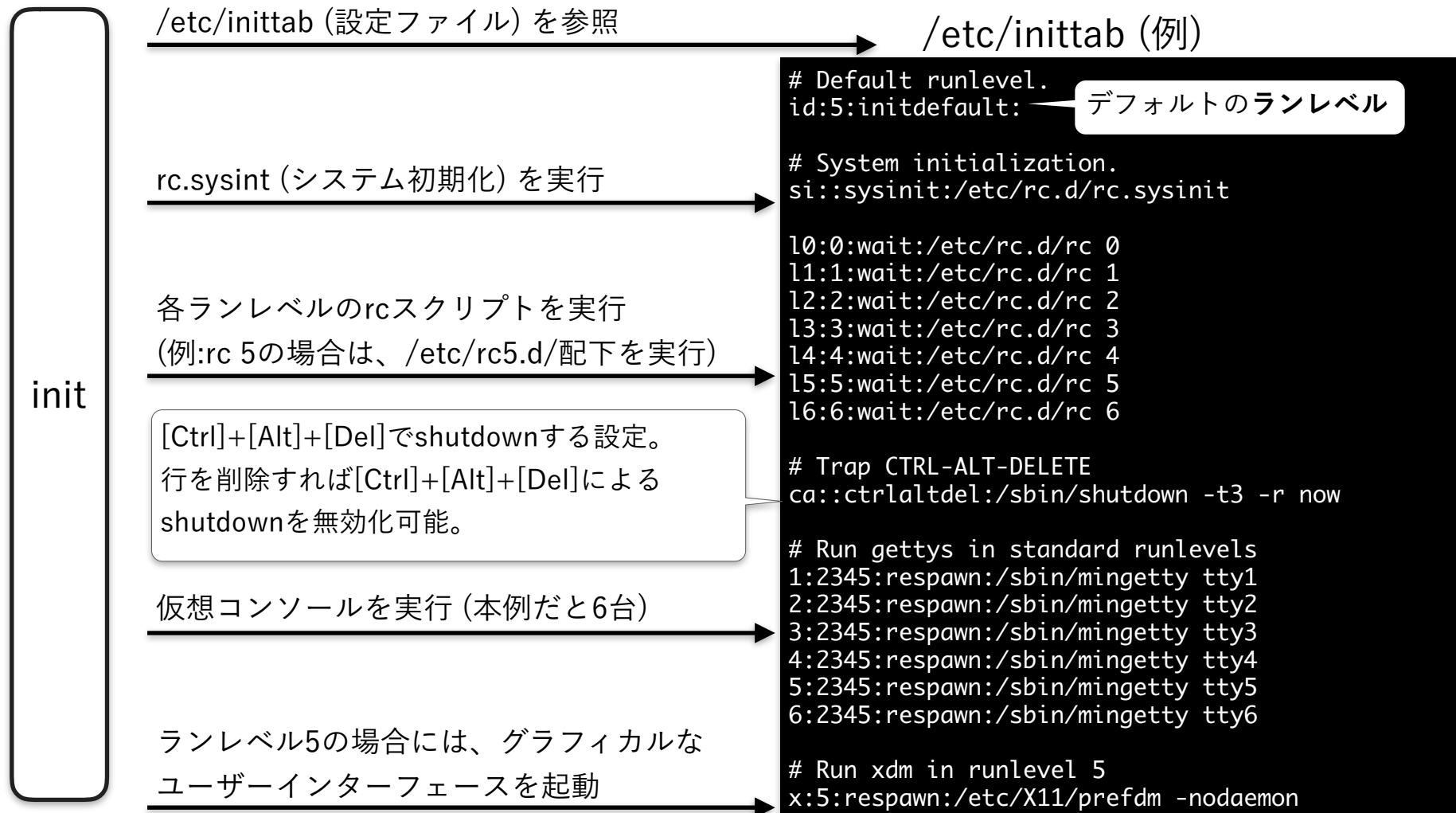
ログイン画面

起動が完了し、ログイン画面が表示されます。



SysVinit (古くからある起動プロセス)

init(デフォルトでは/sbin/init)が最初のプログラム(PID=1)として起動します。initは、**/etc/inittab**の設定に従って各種プロセスを起動させます。





SysVinitでは、システムが実行するサービスの動作レベルとして**ランレベル**があります。

ランレベル	内容
0	停止
1	シングルユーザーモード (1人のみログイン可能)
2	マルチユーザーモード (CUI、NFS停止) (複数人がテキストモードでのログインが可能。 Network File System以外のネットワークが利用可能)
3	マルチユーザーモード (CUI) (複数人がテキストモードでのログイン可能。 全てのネットワークが利用可能)
4	未使用
5	マルチユーザーモード (GUI) (複数人がグラフィカルモードでのログイン可能。)
6	再起動

注意：ランレベルは、ディストリビューションによって僅かですが異なります。
本資料では、CentOSを対象としています。



ランレベルの確認

- ・ コマンド

runlevel	直前および現在のランレベルを表示。 (直前のランレベルがない場合は、'N'を表示)
----------	--

- ・ 実行例

```
[root@localhost ~]# runlevel  
N 3
```

現在のランレベルの変更

- ・ コマンド

init	ランレベルを変更。
telinit	ランレベルを変更。

- ・ 実行例 (init : シングルユーザモードへの変更)

```
[root@localhost ~]# init 1
```

- ・ 実行例 (telinit : 再起動)

```
[root@localhost ~]# telinit 6
```



/etc/init.d/配下にある起動スクリプトを利用して、サービスの起動、終了などができます。

- ・ 実行例（sshdの起動）

```
[root@localhost ~]# /etc/init.d/sshd start  
sshd を起動中: [ OK ]
```

- ・ 実行例（sshdの停止）

```
[root@localhost ~]# /etc/init.d/sshd stop  
sshd を停止中: [ OK ]
```

- ・ 実行例（sshdの状態確認）

```
[root@localhost ~]# /etc/init.d/sshd status  
openssh-daemon (pid 1146) を実行中...
```

補足：プロセスとサービスの違い。

プロセスは、OS が管理するプログラムの実行単位。サービスは、利用者から見た主に機能的な単位。

SysVinitを採用したシステムでは、shutdownコマンドでシステムのシャットダウン、再起動ができます。

- ・ コマンド

shutdown	-h	システムのシャットダウン
	-r	システムの再起動

- ・ 実行例（システムの即時シャットダウン）

```
[root@localhost ~]# shutdown -h now
```

- ・ 実行例（指定時間のシステムの再起動、17:54に実行した例）

```
[root@localhost ~]# shutdown -r 17:55

Broadcast message from root@localhost.localdomain
(/dev/pts/0) at 17:54 ...

The system is going down for reboot in 1 minute!
```

補足：即時、指定時間は、シャットダウン、再起動の双方で実行可能です。



Upstartは、SysVinitに代わるイベント駆動型のinitプログラムとして開発されました。

* SysVinitの課題

- ・ ランレベル毎に設定された起動手順に従ってプロセスを起動する。
- ・ 並列に起動することができず、起動完了までに時間がかかる。

* Upstart

- ・ イベント発生を契機として、イベントと紐付いている処理が起動する。
- ・ 非同期に起動することができ、短時間で起動完了ができる。



systemdでは、initの代わりに**systemd**プロセスが起動し、サービスを管理します。起動処理は**Unit**という単位（設定ファイル）に分かれています。

スクリプト内の処理を細分化しUnitとして定義

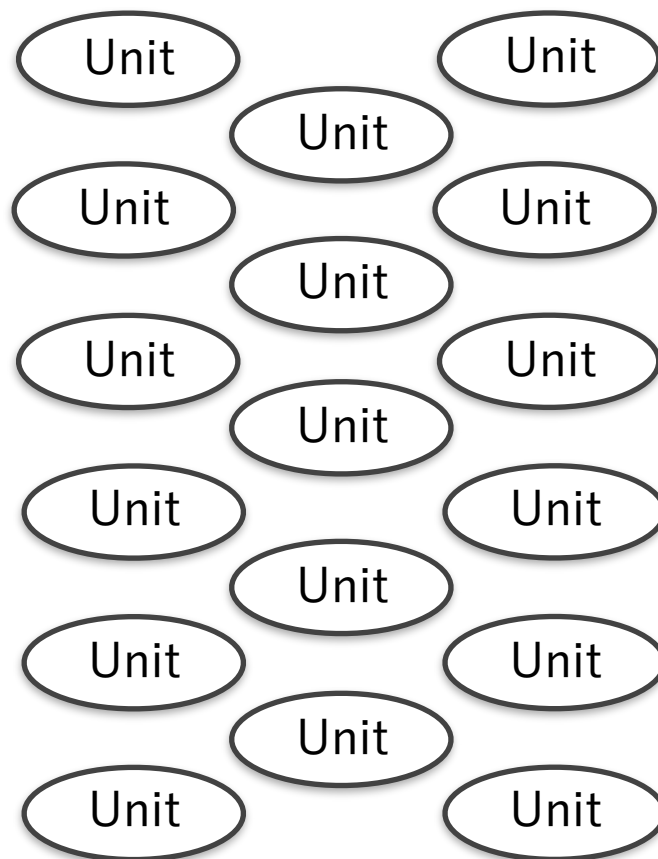
```
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```





Unitの定義ファイルは、以下に格納されています。

格納先	内容
/etc/systemd/system	利用者による設定
/usr/lib/systemd/system	システムのデフォルト設定

Unitは、その役割によって種類が別れています。

Unit	内容
service	プロセスの起動／停止に関する設定。 ファイル名は「プロセス名.service」となる
mount	ファイルシステムのマウント／アンマウントに関する設定。 ファイル名は「マウントポイント.mount」となる
device	システムが認識しているデバイス情報を保持。 udevデーモンによって自動作成される
socket	特定のソケットを監視。
path	特定のパス（ディレクトリ）を監視。
target	複数のUnitをとりまとめるUnit



ターゲットとランレベル

systemdは、システム起動時にまずdefault.targetを起動します。default.targetは、/etc/systemd/system配下に格納されています。また、SysVinitにおけるランレベルに対応したターゲットも準備されています。

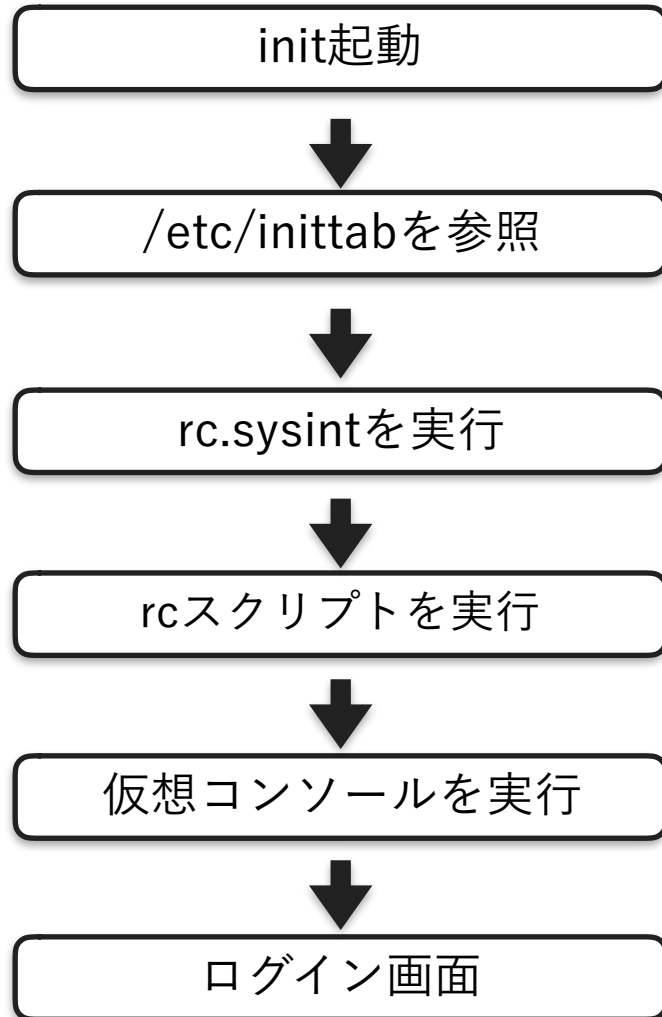
- ・ ターゲットと対応するランレベル

ターゲット	内容	対応ランレベル
poweroff.target	停止	0
rescue.target	シングルユーザーモード	1
multi-user.target	マルチユーザーモード (CUI)	2, 3
graphical.target	マルチユーザーモード (GUI)	5
reboot.target	再起動	6

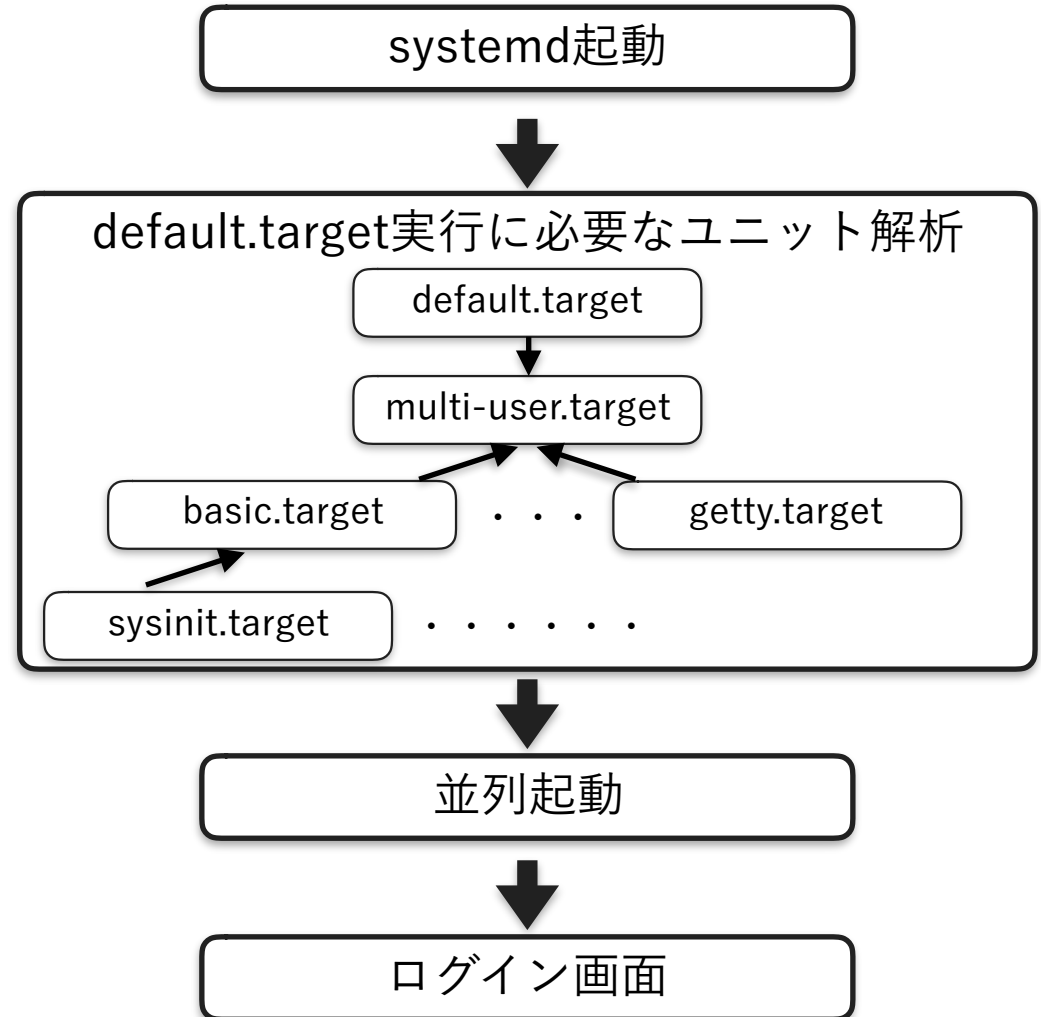


起動処理の違い

SysVinit



systemd





default.targetの確認。

```
[root@localhost ~]# systemctl get-default
multi-user.target
```

default.targetの変更。

- ・ systemctlコマンドによる変更（例：rescue.targetへの変更）

```
[root@localhost ~]# systemctl set-default rescue.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/rescue.target.
```

- ・ ブートローダへのカーネルオプション追加
（例：multi-user.targetへの変更）

```
systemd.unit=multi-user.target
```

現在のターゲットの変更。

- ・ systemctlコマンドによる変更

```
[root@localhost ~]# systemctl isolate graphical.target
```



systemdではsystemctlコマンドを利用して、サービスの起動、終了ができます。

- ・ 実行例（sshdの起動）

```
[root@localhost ~]# systemctl start sshd.service
```

- ・ 実行例（sshdの停止）

```
[root@localhost ~]# systemctl stop sshd
```

- ・ 実行例（sshdの状態確認）

```
[root@localhost ~]# systemctl status sshd.service
• sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset:
  enabled)
  Active: active (running) since 日 2017-04-16 16:13:30 JST; 55min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Process: 903 ExecStart=/usr/sbin/sshd $OPTIONS (code=exited, status=0/SUCCESS)
  .
  .
```

(省略)



システムのシャットダウン、再起動

systemdを採用したシステムでは、システムのシャットダウン、再起動もsystemctlコマンドで実行できます。

- ・ 実行例（システムのシャットダウン）

```
[root@localhost ~]# systemctl poweroff
```

- ・ 実行例（システムの再起動）

```
[root@localhost ~]# systemctl reboot
```



本日の内容

1. LPICレベル1試験の概要説明

- a. Linuxとは？
- b. LPICとは？
- c. LPICレベル1とは？

2. 主題101:システムアーキテクチャの説明

- a. ハードウェア設定の決定と構成
- b. システムのブート
- c. ランレベル/ブートターゲットの変更と
システムのシャットダウンまたはリブート

3. 主題101の例題解説



主題101の例題解説は、LPI-Japan 認定教材の著者および発行元のご好意により特別に許可を頂き、現在販売中の認定教材の中から例題の選定を行っております。

そのため、資料の公開・配布は行わず、LPICレベル1技術解説無料セミナーへのご参加者の皆様のみ当日解説いたします。

LPI-Japan

Open Your Future

Technology Director

和田真輝

maski.wada@lpi.or.jp