

Bagian 1

Instalasi Linux

A. Tujuan

- Mahasiswa memahami kebutuhan sistem operasi Linux.
- Mahasiswa memahami tahap-tahap dalam proses instalasi sistem operasi.

B. Teori

Debian adalah sistem operasi gratis (OS) untuk komputer Anda. Sistem operasi adalah seperangkat program dasar dan utilitas yang membuat komputer Anda berjalan.

Sistem Debian saat ini menggunakan kernel Linux atau kernel FreeBSD. Linux adalah perangkat lunak yang dimulai oleh Linus Torvalds dan didukung oleh ribuan programmer di seluruh dunia. FreeBSD adalah sistem operasi termasuk kernel dan perangkat lunak lainnya.

Debian adalah salah satu distribusi Linux tertua. Pada 2015, rilis Debian dinamai sesuai karakter dalam film Toy Story. The Jessie rilis berisi sekitar 36.000 paket dan rilis terakhir (Debian 9 Stretch Juni 2017, dinamai toy rubber octopus dengan pengisap di delapan lengan panjangnya yang muncul di Toy Story 3)

Tabel Debian releases

name	number	year
Woody	3.0	2002
Sarge	3.1	2005
Etch	4.0	2007
Lenny	5.0	2009
Squeeze	6.0	2011
Wheezy	7	2013
Jessie	8	2015
Stretch	9	2017

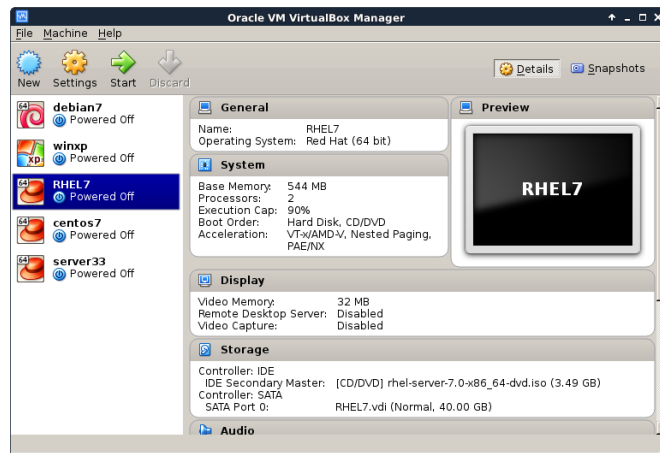
C. Alat dan Bahan

- Komputer
- Debian DVD Images + Virtual Box

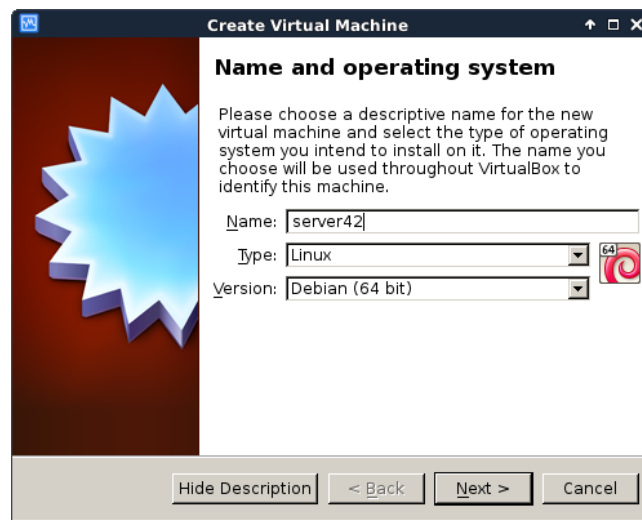
D. Praktik (Minggu 1 & 2)

1. Buat mesin virtualbox baru

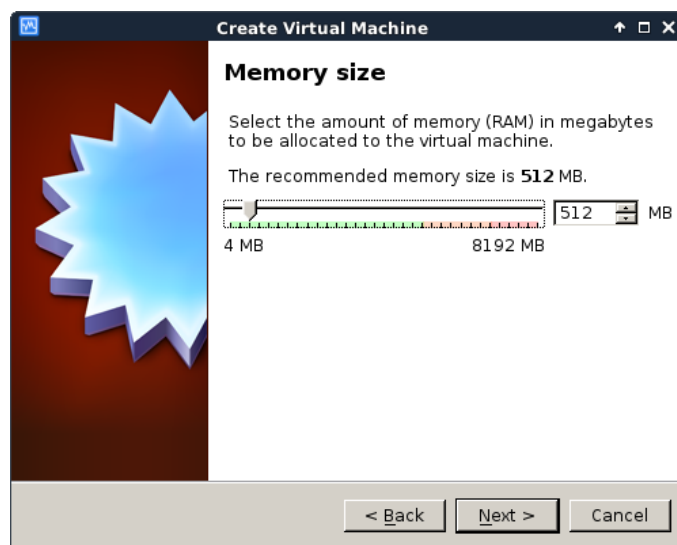
- Klik tombol New untuk memulai wizard yang akan membantu Anda membuat mesin virtual.

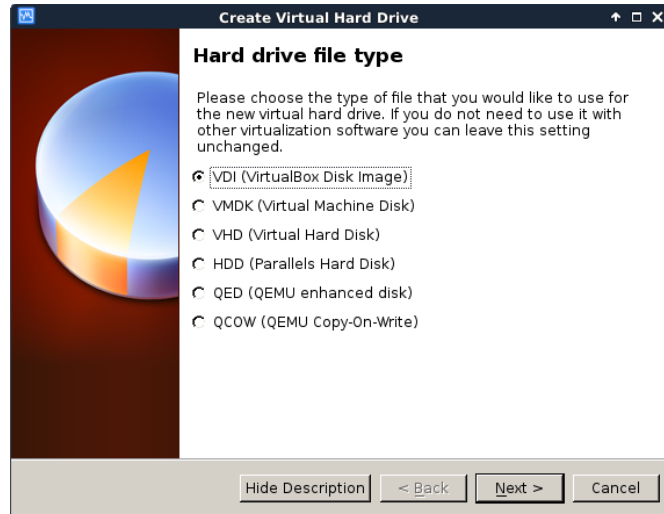


- Mesin memerlukan nama, screenshot ini menunjukkan bahwa menamakannya server42

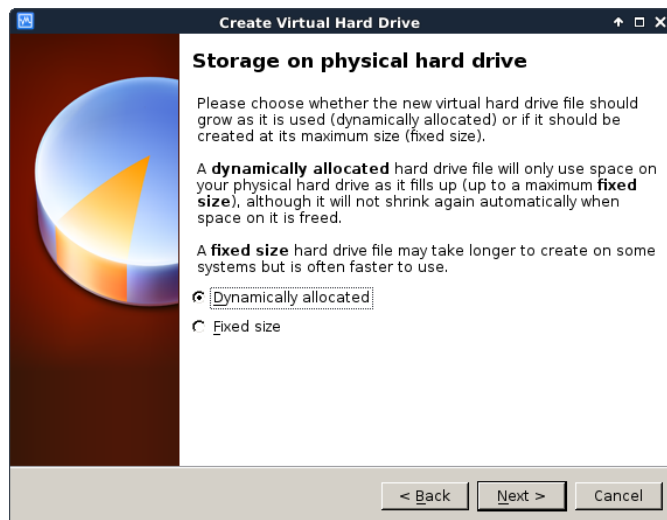


- 1 GB of RAM Cukup untuk melaksanakan praktik keseluruhan .

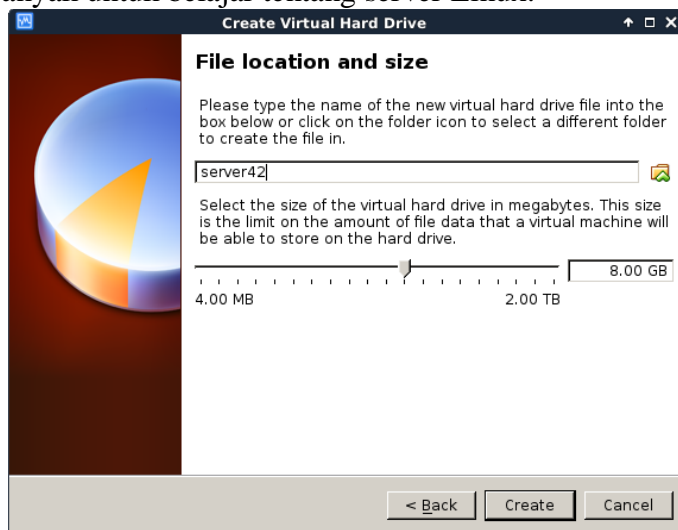




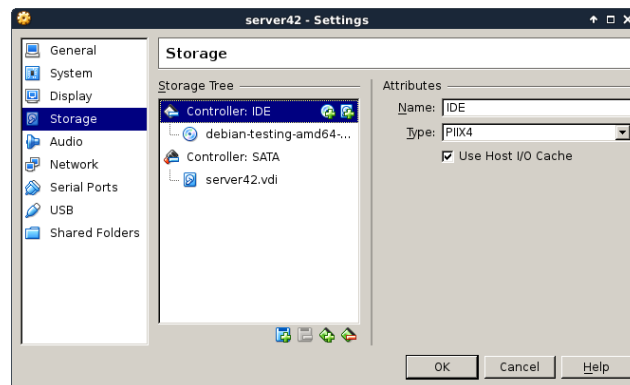
- Memilih **dynamically allocated** akan menghemat beberapa ruang disk (untuk kinerja kecil).



- 8GB harus banyak untuk belajar tentang server Linux.



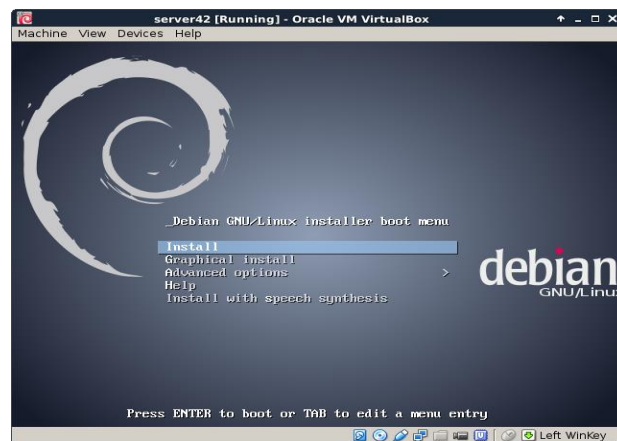
Ini akhiri wizard. Anda mesin virtual hampir siap untuk memulai instalasi. Pertama, pastikan Anda melampirkan image .iso yang didownload ke drive CD virtual. (dengan membuka **Settings**, **Storage** diikuti dengan klik mouse pada ikon CD bulat)



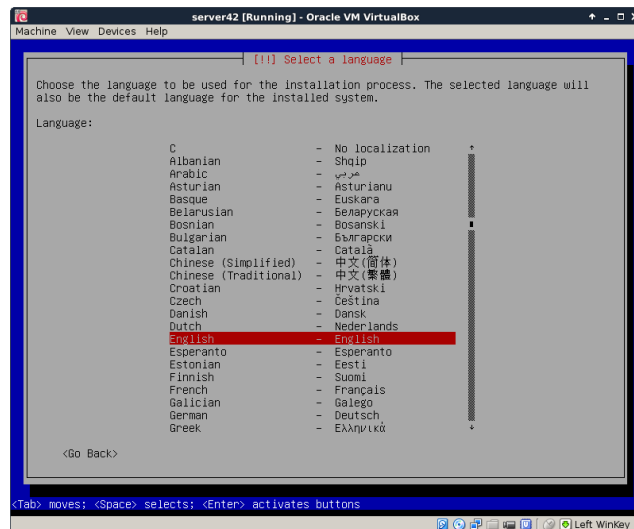
disarankan menonaktifkan suara dan usb, karena tidak pernah menggunakan fitur ini. Saya juga menghapus floppy disk dan menggunakan penunjuk mouse PS / 2. Ini mungkin tidak terlalu penting, tp itu ide untuk menghemat beberapa sumber daya.

2. Install Debian.

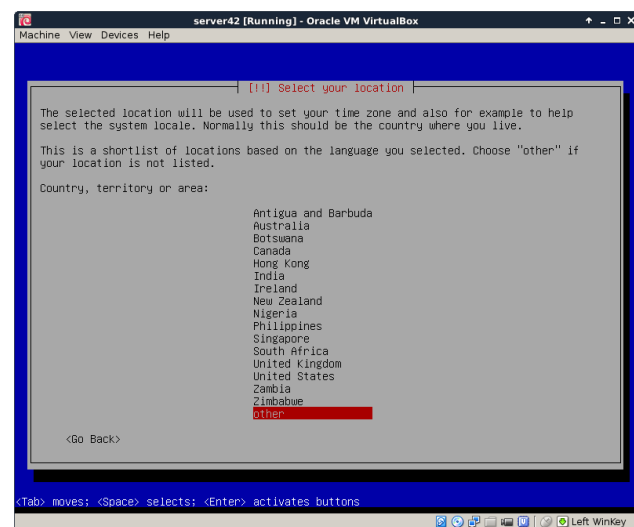
Sekarang boot mesin virtual dan mulai instalasi yang sebenarnya. Setelah beberapa detik Anda akan melihat layar yang serupa dengan ini. Pilih Instal untuk memulai instalasi Debian.



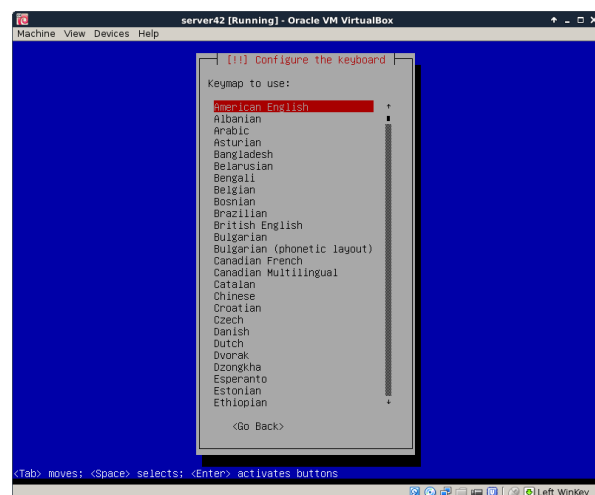
- Pertama pilihlah bahasa yang ingin digunakan.



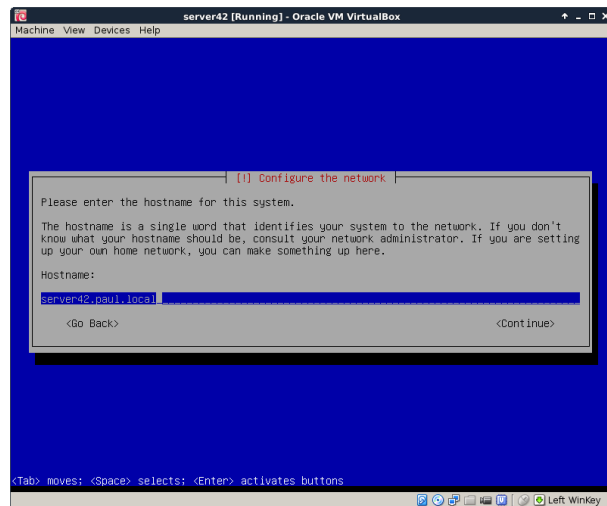
- Pilih Negara, Informasi ini akan digunakan untuk menyarankan unduh mirror



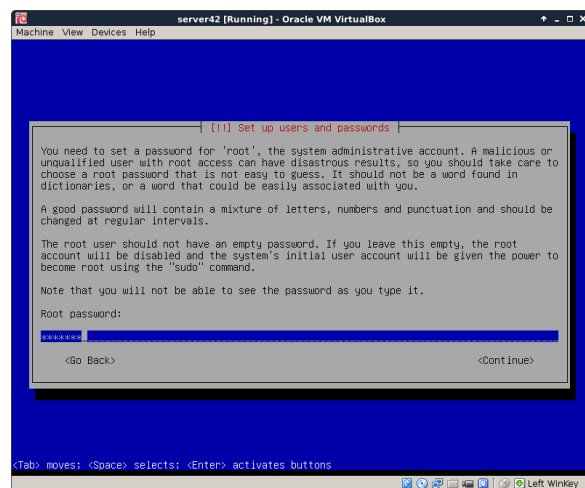
- Pilih keyboard yang benar. Pada server ini tidak penting karena sebagian besar server dikelola secara remote melalui ssh.



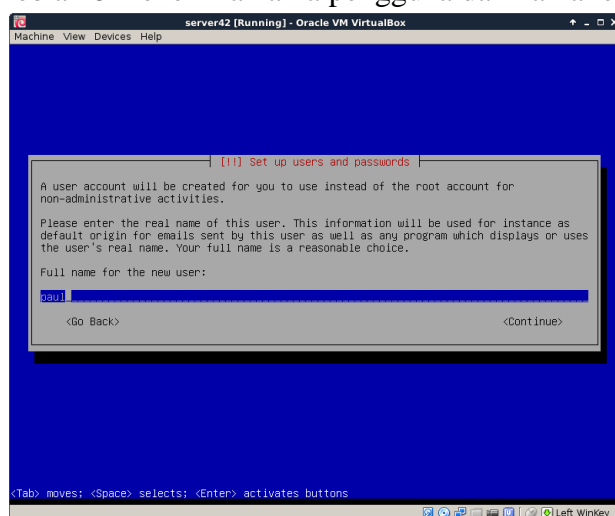
- Enter **hostname** (dengan **fqdn** untuk mengatur **dnsdomainname**)



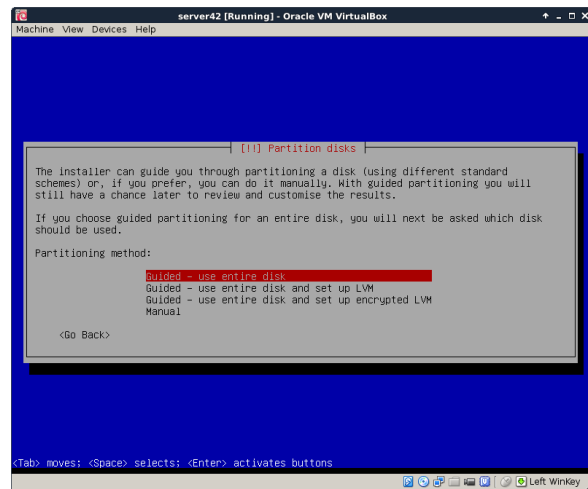
- Berikan root user sebuah password, mohon diingat password tersebut.



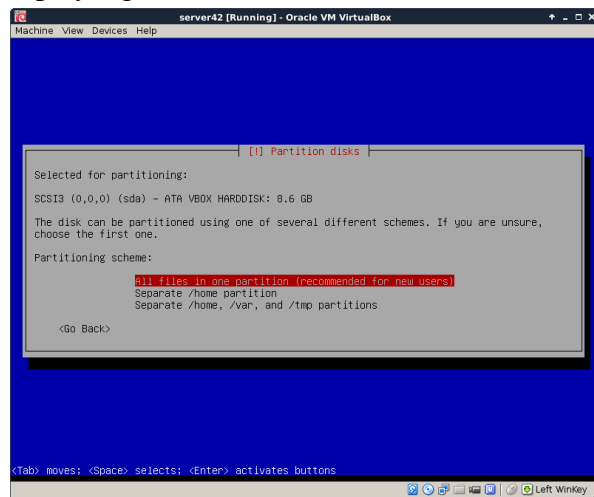
- Disarankan untuk juga membuat akun pengguna biasa. Saya tidak memberikan nama lengkap saya, Debian 8 menerima nama pengguna dan nama lengkap yang sama.



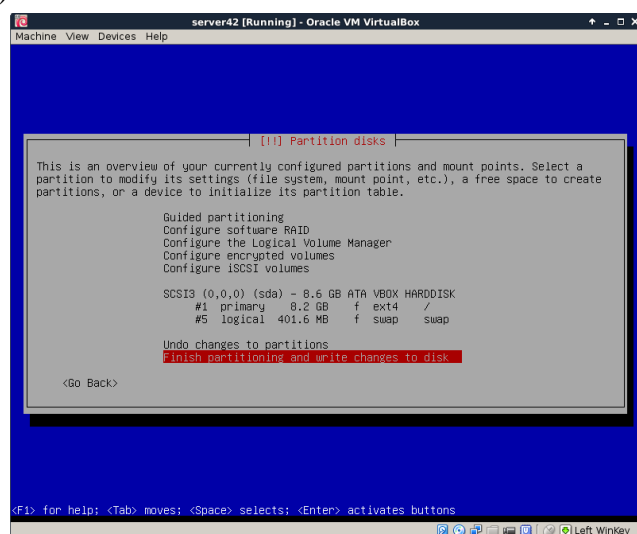
- **use entire disk** mengacu pada disk virtual yang Anda buat sebelumnya di Virtualbox.



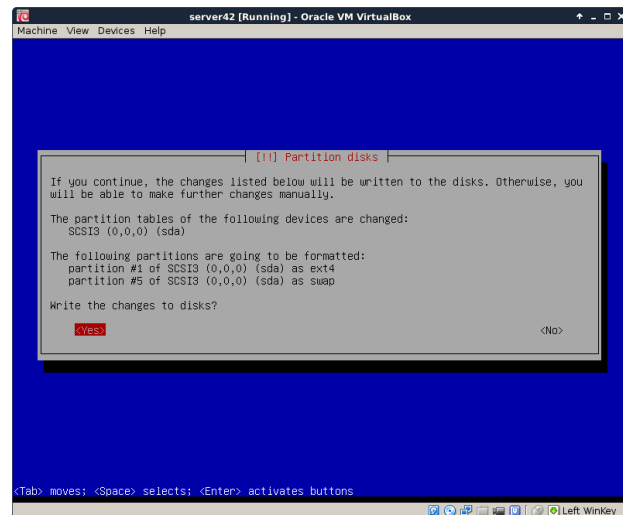
- Sekali lagi standarnya mungkin yang Anda inginkan. Hanya ubah partisi jika Anda benar-benar tahu apa yang Anda lakukan



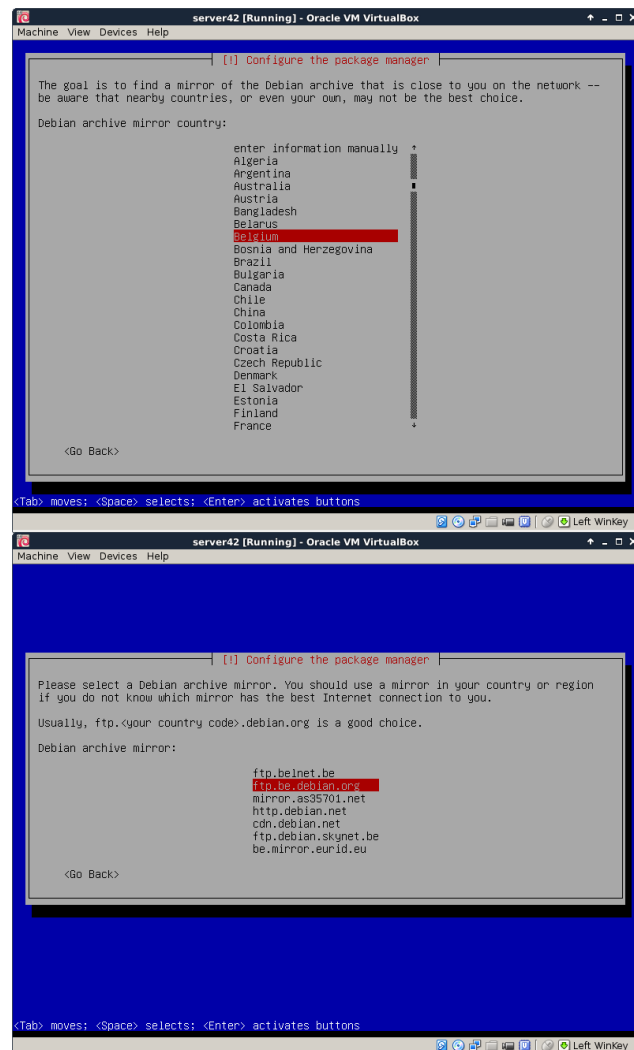
- Accept the partition layout (hanya berubah jika Anda benar-benar tahu apa yang Anda lakukan).



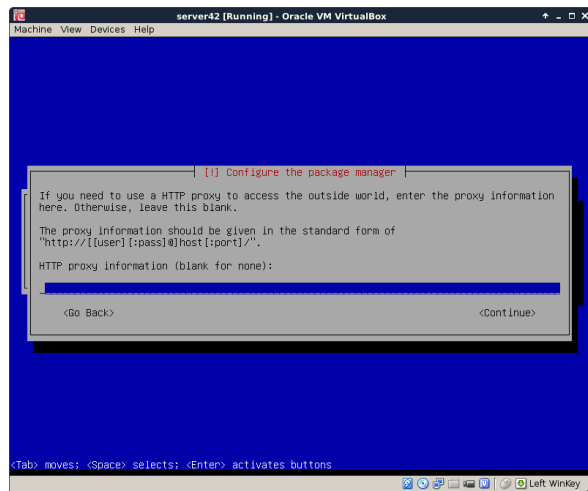
- Ini adalah point of no return, momen magis di mana menekan ya selamanya akan menghapus data pada komputer (virtual).



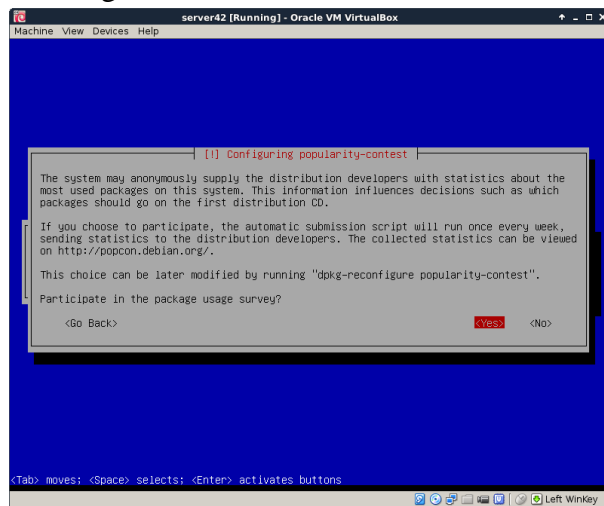
- Perangkat lunak diunduh dari mirror repository, sebaiknya pilih yang dekat (seperti di negara yang sama).



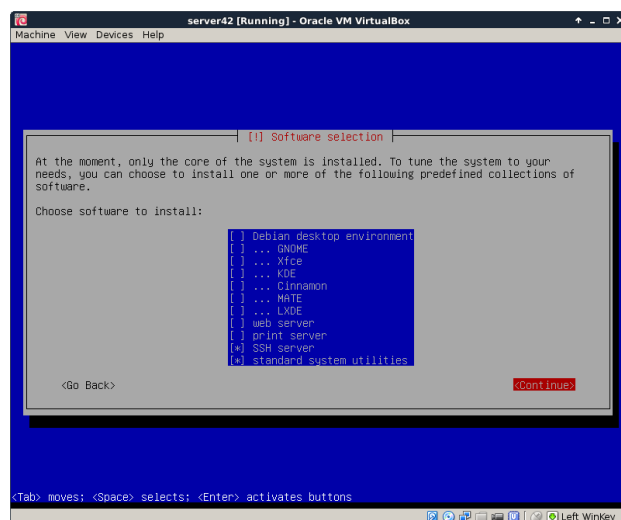
- Biarkan bidang proxy kosong (kecuali Anda yakin bahwa Anda berada di belakang server proxy).

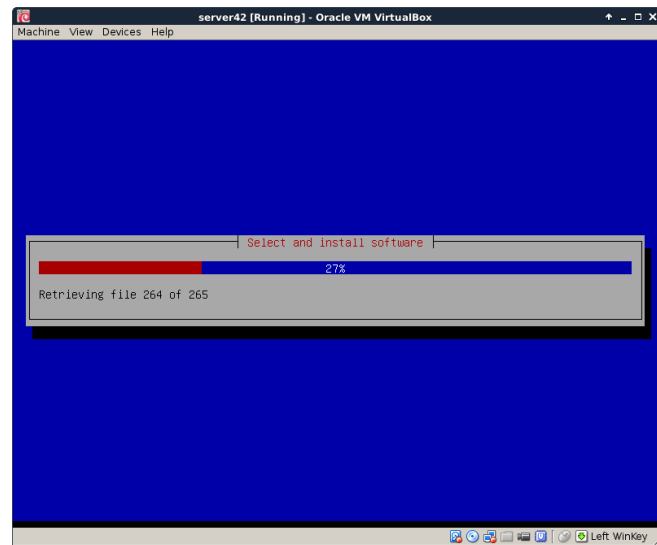


- Pilih apakah Anda ingin mengirim statistik anonim ke proyek Debian (mengumpulkan data tentang paket yang diinstal). Anda dapat melihat statistik di sini <http://popcon.debian.org/>.

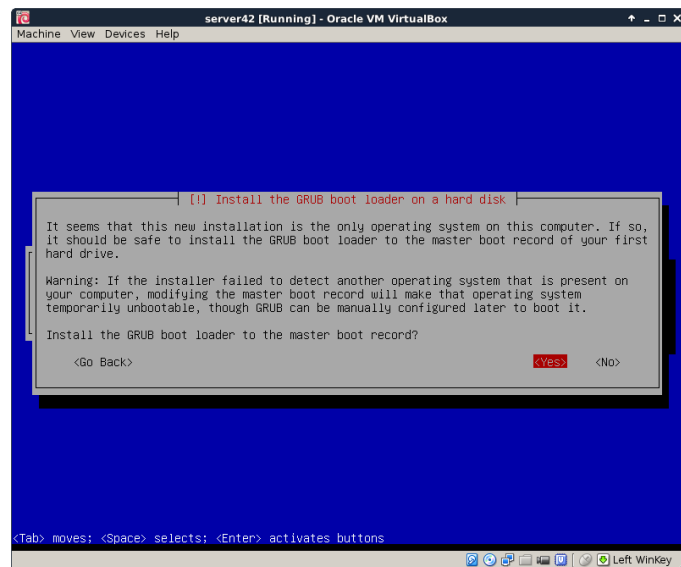


- Pilih perangkat lunak apa yang harus dipasang, kami tidak memerlukan barang grafis apa pun untuk pelatihan ini.

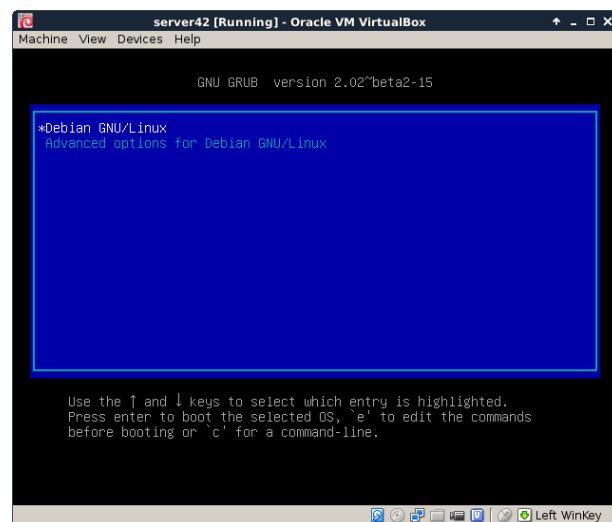




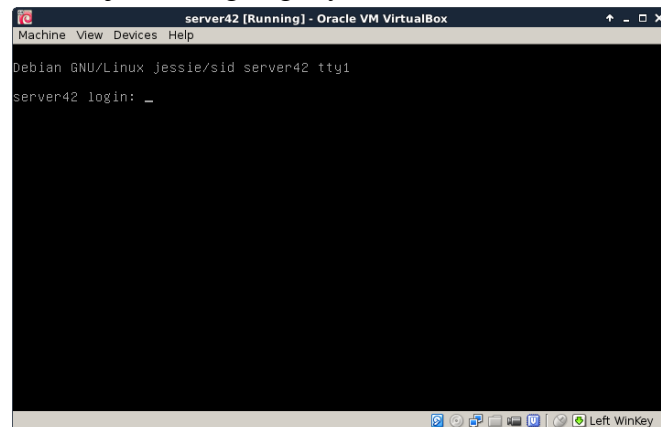
- Say yes untuk menginstal bootloader pada mesin virtual



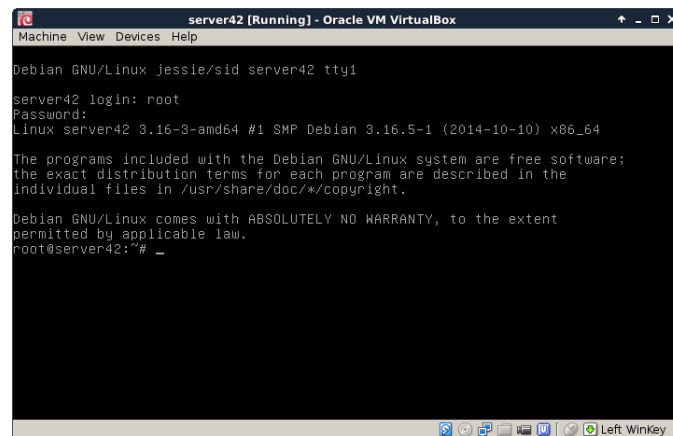
3. Booting untuk pertama kalinya menunjukkan layar grub



- Beberapa detik kemudian Anda akan melihat banyak teks bergulir di layar (dmesg). Setelah itu Anda disajikan dengan getty ini dan diizinkan masuk pertama Anda.



- Anda sekarang harus dapat masuk ke mesin virtual Anda dengan akun root. Apakah Anda ingat kata sandinya? Apakah itu hunter2?



```
root@server42:~# who am i
root      tty1      2014-11-10
18:21 root@server42:~# hostname
server42
root@server42:~#
date
Mon Nov 10 18:21:56 CET 2014
```

4. Virtualbox networking

Anda juga dapat log on dari remote (atau dari komputer host Windows / Mac / Linux) menggunakan **ssh** atau **putty**. Ubah pengaturan jaringan di mesin virtual untuk menjembatani. Ini akan memungkinkan mesin virtual Anda untuk menerima alamat ip dari server dhcp lokal Anda. Jaringan virtualbox default adalah melampirkan kartu jaringan virtual ke nat. Layar ini menunjukkan alamat ip 10.0.2.15 ketika di **nat**:

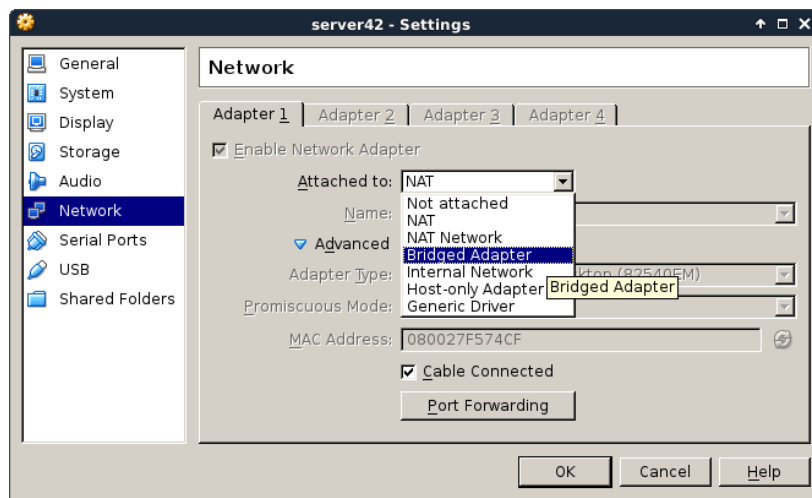
```
root@server42:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:f5:74:cf
          inet addr:10.0.2.15  Bcast:10.0.2.255
          Mask:255.255.255.0  inet6 addr:
          fe80::a00:27ff:fe5:74cf/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```

RX packets:11 errors:0 dropped:0 overruns:0 frame:0
TX packets:19 errors:0 dropped:0 overruns:0
carrier:0 collisions:0 txqueuelen:1000
RX bytes:2352 (2.2 KiB) TX bytes:1988 (1.9 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1
        Mask:255.0.0.0 inet6 addr:
        ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:65536 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0
        carrier:0 collisions:0 txqueuelen:0
        RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

```



Dengan shutting down network interface dan mengaktifkannya kembali, kami memaksa Debian untuk memperbarui alamat ip dari bridged network..

```

root@server42:~# # do not run ifdown while connected over ssh!
root@server42:~# ifdown eth0
Killed old client process
Internet Systems Consortium DHCP Client 4.3.1
Copyright 2004-2014 Internet Systems
Consortium. All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

```

```

Listening on LPF/eth0/08:00:27:f5:74:cf
Sending on   LPF/eth0/08:00:27:f5:74:c

```

```

Sending          on
Socket/fallback
DHCPRELEASE on eth0 to 10.0.2.2 port
67

```

```

root@server42:~# # now enable bridge in virtualbox
settings
root@server42:~# ifup
eth0
Internet Systems Consortium DHCP Client
4.3.1
Copyright 2004-2014 Internet Systems
Consortium. All rights reserved.
For info, please visit
https://www.isc.org/software/dhcp/

```

```

Listening                                on
LPF/eth0/08:00:27:f5:74:cf Sending on
LPF/eth0/08:00:27:f5:74:cf Sending on
Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval
8
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval
8
DHCPREQUEST on eth0 to 255.255.255.255 port
67
DHCPOFFER                                from
192.168.1.42
DHCPACK                                from
192.168.1.42
bound to 192.168.1.111 -- renewal in 2938
seconds. root@server42:~# ifconfig eth0
eth0                                Link      encap:Ethernet      HWaddr
08:00:27:f5:74:cf
        inet addr:192.168.1.111  Bcast:192.168.1.255
        Mask:255.255.255.0  inet6 addr: fe80::a00:27ff:fef5:74cf/64
        Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:15 errors:0 dropped:0 overruns:0 frame:0
        TX packets:31 errors:0 dropped:0 overruns:0
        carrier:0 collisions:0 txqueuelen:1000
        RX bytes:3156 (3.0 KiB)  TX bytes:3722 (3.6 KiB)

root@server42:
~#

```

Berikut ini contoh **ssh** untuk komputer yang baru dipasang ini. Perhatikan bahwa Debian 8 telah menonaktifkan remote root access, jadi saya harus menggunakan akun pengguna normal.

```

paul@debian8:~$ ssh paul@192.168.1.111
paul@192.168.1.111's password:

```

```

The programs included with the Debian GNU/Linux system are free
software; the exact distribution terms for each program are described
in the individual files in /usr/share/doc/*/copyright.

```

```

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the
extent permitted by applicable law.
paul@server42:~$
paul@server42:~$ su
~ Password:
root@server42:~#

```

5. Setting the hostname

Nama host server diminta selama instalasi, jadi tidak perlu mengkonfigurasi ini secara manual.

```

root@server42:~# hostname
server42
root@server42:~# cat /etc/hostname
server42
root@server42:~# dnsdomainname
paul.local
root@server42:~# grep server42 /etc/hosts
127.0.1.1      server42.paul.local
server42 root@server42:~#

```

6. Adding a static ip address

Contoh ini menunjukkan cara menambahkan alamat ip statis ke server Anda. Anda dapat menggunakan ifconfig untuk menetapkan alamat statis yang aktif hingga reboot berikutnya (atau hingga ifdown berikutnya).

```
root@server42:~# ifconfig eth0:0 10.104.33.39
```

Menambahkan beberapa baris ke file / etc / network / interfaces untuk mengaktifkan alamat ip ekstra selamanya.

```
root@server42:~# vi /etc/network/interfaces
root@server42:~# tail -4
/etc/network/interfaces auto eth0:0
iface eth0:0 inet
static address
10.104.33.39 netmask
255.255.0.0
root@server42:~#
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:f5:74:cf
          inet addr:192.168.1.111  Bcast:192.168.1.255
          Mask:255.255.255.0  inet6 addr: fe80::a00:27ff:fef5:74cf/64
          Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:528 errors:0 dropped:0 overruns:0 frame:0
          TX packets:333 errors:0 dropped:0 overruns:0
          carrier:0 collisions:0 txqueuelen:1000
          RX bytes:45429 (44.3 KiB)  TX bytes:48763 (47.6 KiB)

eth0:0    Link encap:Ethernet  HWaddr 08:00:27:f5:74:cf
          inet addr:10.104.33.39  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1
          Mask:255.0.0.0  inet6 addr:
          ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0
          carrier:0 collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@server42:~#
```

7. Debian package management

Untuk mendapatkan semua informasi tentang paket terbaru dari repositori online:

```
root@server42:~# aptitude update
Get: 1 http://ftp.be.debian.org jessie InRelease [191 kB]
Get: 2 http://security.debian.org jessie/updates
InRelease [84.1 kB] Get: 3 http://ftp.be.debian.org
jessie-updates InRelease [117 kB] Get: 4
http://ftp.be.debian.org jessie-backports InRelease [118
kB] Get: 5 http://security.debian.org jessie/updates/main
Sources [14 B]
Get: 6 http://ftp.be.debian.org jessie/main Sources/DiffIndex [7,876 B]
... (output truncated)
```

Untuk mengunduh dan menerapkan semua pembaruan untuk semua paket yang diinstal:

```
root@server42:~# aptitude upgrade
Resolving dependencies...
The following NEW packages will be installed:
```

```

firmware-linux-free{a} irqbalance{a} libnumal{a} linux-image-
3.16.0-4-amd64{a} The following packages will be upgraded:
busybox file libc-bin libc6 libexpat1 libmagic1 libpaper-utils libpaper1
libsqlite3-0 linux-image-amd64 locales multiarch-support
12 packages upgraded, 4 newly installed, 0 to remove and 0
not upgraded. Need to get 44.9 MB of archives. After
unpacking 161 MB will be used.
Do you want to continue? [Y/n/?]
... (output truncated)

```

Untuk menginstal perangkat lunak baru (vim dan tmux dalam contoh ini):

```

root@server42:~# aptitude
install vim tmux
The following NEW packages will be
installed:
tmux vim vim-runtime{a}
0 packages upgraded, 3 newly installed, 0 to remove and 0
not upgraded. Need to get 6,243 kB of archives. After
unpacking 29.0 MB will be used. Do you want to continue?
[Y/n/?]
Get: 1 http://ftp.be.debian.org/debian/ jessie/main tmux amd64
1.9-6 [245 kB]
Get: 2 http://ftp.be.debian.org/debian/ jessie/main vim-runtime all 2:7.4.488-1
[5,046 kB] Get: 3 http://ftp.be.debian.org/debian/ jessie/main vim amd64
2:7.4.488-1 [952 kB]

```

E. Latihan

- Install Ubuntu, CentOS, Ubuntu Mate On Raspberry Pi

Bagian 2 Bekerja dengan direktori dan file

A. Tujuan

- Mahasiswa memahami dan dapat bekerja dengan direktori
- Mahasiswa memahami dan dapat bekerja dengan berbagai file
- Mahasiswa dapat melakukan berbagai operasi file

B. Teori

Semua File case sensitive

File di Linux (atau Unix lainnya) adalah case sensitive. Ini berarti bahwa **FILE1** berbeda dari **file1**, dan `/ etc / hosts` berbeda dari `/ etc / Hosts` (yang terakhir tidak ada pada komputer Linux yang khas).

Everything is a **file**.

Direktori adalah jenis file khusus, tetapi masih merupakan file (case sensitive!). Setiap jendela terminal (misalnya `/ dev / pts / 4`), hard disk atau partisi apa pun (misalnya `/ dev / sdb1`) dan proses apa pun semuanya diwakili di suatu tempat di sistem file sebagai file. Ini akan menjadi jelas di seluruh kursus ini bahwa semua yang ada di Linux adalah sebuah file.

Utilitas file menentukan jenis file. Linux tidak menggunakan ekstensi untuk menentukan jenis file. Baris perintah tidak peduli apakah file berakhiran `.txt` atau `.pdf`. Sebagai administrator sistem, Anda harus menggunakan perintah file untuk menentukan jenis file

Modul ini adalah gambaran singkat dari perintah yang paling umum untuk bekerja dengan direktori: `pwd`, `cd`, `ls`, `mkdir` dan `rmdir`. Perintah-perintah ini tersedia di sistem Linux (atau Unix) apa pun. Modul ini juga membahas absolute dan relative paths serta penyelesaian jalur dalam bash Shell

C. Alat dan Bahan

- Komputer
- Virtual Box dengan Debian OS

D. Praktik (Minggu 3 & 4)

1. Work dengan direktori

- **pwd**

Tempat/direktori saat ini dapat ditampilkan dengan perintah **pwd** (Print Working Directory). Silakan, coba: Buka antarmuka baris perintah (juga disebut terminal, konsol atau xterm) dan ketik `pwd`. Alat ini menampilkan direktori Anda saat ini.

```
paul@debian8:~$ pwd
/home/paul
```

- **cd**

Anda dapat mengubah direktori Anda saat ini dengan perintah `cd` (Change Directory).


```
paul@debian8$ cd /etc
paul@debian8$ pwd
/etc
paul@debian8$ cd /bin
paul@debian8$ pwd
/bin
paul@debian8$ cd /home/paul/
paul@debian8$ pwd
/home/paul
```

- **cd ~**

CD juga merupakan jalan pintas untuk kembali ke direktori home Anda. Hanya mengetik cd tanpa direktori target, akan menempatkan Anda di direktori home Anda. Mengetik cd ~ memiliki efek yang sama.

```
paul@debian8$ cd /etc
paul@debian8$ pwd
/etc
paul@debian8$
cd
paul@debian8$
pwd
/home/paul
paul@debian8$ cd
~ paul@debian8$
pwd
/home/paul
```

- **cd ..**

Untuk masuk ke **parent directory** (yang baru saja di atas direktori Anda saat ini di pohon direktori), ketik cd ...

```
paul@debian8$ pwd
/usr/share/games
paul@debian8$ cd
.. paul@debian8$
pwd
/usr/share
```

*To stay in the current directory, type **cd .** ;-)* We will see useful use of the **.** character representing the current directory later.

- **cd -**

Cara pintas lain yang berguna dengan cd adalah hanya mengetik cd - untuk pergi ke direktori sebelumnya.

```
paul@debian8$ pwd
/home/paul
paul@debian8$ cd
/etc paul@debian8$
pwd
/etc
paul@debian8$ cd -
/home/paul
paul@debian8$ cd
-
/etc
```

- **absolute and relative paths**

Anda harus menyadari **absolute and relative paths** dalam pohon file. Ketika Anda mengetikkan jalur yang dimulai dengan **slash (/)**, maka akar dari pohon file diasumsikan. Jika Anda tidak memulai jalur Anda dengan **slash**, maka direktori saat ini adalah titik awal yang diasumsikan. Screenshot di bawah ini pertama menunjukkan direktori / rumah / paul saat ini. Dari dalam direktori ini, Anda harus mengetikkan `cd /` home alih-alih `cd home` untuk masuk ke direktori / home.

```
paul@debian8$ pwd
/home/paul
paul@debian8$ cd
home
bash: cd: home: No such file or
directory paul@debian8$ cd /home
paul@debian8$ pwd
/home
```

Ketika di dalam **/home**, Anda harus mengetikkan **/home** daripada `cd / paul` untuk memasuki subdirektori paul dari direktori / rumah saat ini.

```
paul@debian8$ pwd
/home
paul@debian8$ cd /paul
bash: cd: /paul: No such file or
directory paul@debian8$ cd paul
paul@debian8$ pwd
/home/paul
```

In case your current directory is the **root directory /**, then both `cd /home` and `cd home` will get you in the **/home** directory. Jika direktori Anda saat ini adalah **root directory /**, maka baik `cd /home` dan `cd home` akan membawa Anda ke direktori /home.

```
paul@debian8$ pwd
/
paul@debian8$ cd home
paul@debian8$ pwd
/home
paul@debian8$ cd /
paul@debian8$ cd
/home paul@debian8$
pwd
/home
```

- **Path Complation**

Kunci tab dapat membantu Anda mengetik jalur tanpa kesalahan. Menetik `cd /et` diikuti oleh tombol tab akan memperluas baris perintah ke `cd / etc /`. Saat mengetik `cd /Et` diikuti oleh tombol tab, tidak akan ada yang terjadi karena Anda mengetikkan jalan yang salah (huruf besar E).

Anda akan membutuhkan lebih sedikit masukkan kunci saat menggunakan kunci tab, dan Anda akan yakin bahwa Anda mengetik path yang benar!

- **ls**

Anda dapat daftar isi direktori dengan **ls**.

```
paul@debian8:~$
ls
```

```
allfiles.txt dmesg.txt services stuff
summer.txt paul@debian8:~$
```

- **ls -a**

Opsi yang sering digunakan dengan **ls** adalah **-a** untuk menampilkan semua file. Menampilkan semua file berarti termasuk file yang tersembunyi / **hidden files**. Ketika nama file pada sistem file Linux dimulai dengan sebuah titik, itu dianggap sebagai file tersembunyi dan itu tidak muncul dalam daftar file biasa.

```
paul@debian8:~$
ls
allfiles.txt dmesg.txt services stuff
summer.txt paul@debian8:~$ ls -a
. allfiles.txt .bash_profile dmesg.txt .lessht
stuff
.. .bash_history .bashrc services .ssh
summer.txt paul@debian8:~$
```

- **ls -l**

Banyak kali Anda akan menggunakan opsi dengan **ls** untuk menampilkan isi direktori dalam format yang berbeda atau untuk menampilkan bagian yang berbeda dari direktori. Mengetik hanya **ls** memberi Anda daftar file dalam direktori. Mengetik **ls-l** (yaitu huruf L, bukan angka 1) memberi Anda daftar panjang.

```
paul@debian8:~$ ls -l
total 17296
-rw-r--r-- 1 paul paul 17584442 Sep 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul 96650 Sep 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul 19558 Sep 17 00:04
services drwxr-xr-x 2 paul paul 4096 Sep 17
00:04 stuff
-rw-r--r-- 1 paul paul 0 Sep 17 00:04 summer.txt
```

- **ls -lh**

Pilihan **ls** lain yang sering digunakan adalah **-h**. Ini menunjukkan angka (ukuran file) dalam format yang lebih mudah dibaca manusia. Juga ditampilkan di bawah ini adalah beberapa variasi dalam cara Anda dapat memberikan opsi ke **ls**.

Note that we use the letter L as an option in this screenshot, not the number 1.

```
paul@debian8:~$ ls -l -h
total 17M
-rw-r--r-- 1 paul paul 17M Sep 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul 95K Sep 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul 20K Sep 17 00:04
services drwxr-xr-x 2 paul paul 4.0K Sep 17
00:04 stuff
-rw-r--r-- 1 paul paul 0 Sep 17 00:04
summer.txt paul@debian8:~$ ls -lh
total 17M
-rw-r--r-- 1 paul paul 17M Sep 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul 95K Sep 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul 20K Sep 17 00:04
services drwxr-xr-x 2 paul paul 4.0K Sep 17
00:04 stuff
-rw-r--r-- 1 paul paul 0 Sep 17 00:04
summer.txt paul@debian8:~$ ls -hl
total 17M
```

```

-rw-r--r-- 1 paul paul 17M Sep 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul 95K Sep 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul 20K Sep 17 00:04
services drwxr-xr-x 2 paul paul 4.0K Sep 17
00:04 stuff
-rw-r--r-- 1 paul paul 0 Sep 17 00:04
summer.txt paul@debian8:~$ ls -h -l
total 17M
-rw-r--r-- 1 paul paul 17M Sep 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul 95K Sep 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul 20K Sep 17 00:04
services drwxr-xr-x 2 paul paul 4.0K Sep 17
00:04 stuff
-rw-r--r-- 1 paul paul 0 Sep 17 00:04
summer.txt paul@debian8:~$

```

- **mkdir**

Berjalan di sekitar pohon file Unix itu menyenangkan, tetapi bahkan lebih menyenangkan untuk membuat direktori Anda sendiri dengan **mkdir**. Anda harus memberikan setidaknya satu parameter ke **mkdir**, nama direktori baru yang akan dibuat. Pikirkan sebelum Anda mengetik **/**.

```

paul@debian8:~$ mkdir
mydir paul@debian8:~$ cd
mydir
paul@debian8:~/mydir$ ls -
al total 8
drwxr-xr-x 2 paul paul 4096 Sep 17 00:07
. drwxr-xr-x 48 paul paul 4096 Sep 17
00:07 .. paul@debian8:~/mydir$ mkdir stuff
paul@debian8:~/mydir$ mkdir otherstuff
paul@debian8:~/mydir$ ls -l
total 8
drwxr-xr-x 2 paul paul 4096 Sep 17 00:08
otherstuff drwxr-xr-x 2 paul paul 4096 Sep 17
00:08 stuff paul@debian8:~/mydir$

```

- **mkdir -p**

Perintah berikut akan gagal, karena direktori induk **threedirsdeep** tidak ada.

```

paul@debian8:~$ mkdir mydir2/mysubdir2/threedirsdeep
mkdir: cannot create directory 'mydir2/mysubdir2/threedirsdeep': No such fi\
le or directory

```

Ketika diberi opsi **-p**, maka **mkdir** akan membuat **parent directories** sesuai kebutuhan.

```

paul@debian8:~$ mkdir -p mydir2/mysubdir2/threedirsdeep
paul@debian8:~$ cd mydir2
paul@debian8:~/mydir2$ ls
-l total 4
drwxr-xr-x 3 paul paul 4096 Sep 17 00:11
mysubdir2 paul@debian8:~/mydir2$ cd mysubdir2
paul@debian8:~/mydir2/mysubdir2$ ls -l
total 4
drwxr-xr-x 2 paul paul 4096 Sep 17 00:11
threedirsdeep paul@debian8:~/mydir2/mysubdir2$ cd
threedirsdeep/
paul@debian8:~/mydir2/mysubdir2/threedirsdeep$ pwd
/home/paul/mydir2/mysubdir2/threedirsdeep

```

- **rmdir**

Ketika sebuah direktori kosong, Anda dapat menggunakan **rmdir** untuk menghapus direktori.

```
paul@debian8:~/mydir$ ls -l
total 8
drwxr-xr-x 2 paul paul 4096 Sep 17 00:08
otherstuff drwxr-xr-x 2 paul paul 4096 Sep 17
00:08 stuff paul@debian8:~/mydir$ rmdir
otherstuff paul@debian8:~/mydir$ cd ..
paul@debian8:~$ rmdir mydir
rmdir: failed to remove 'mydir': Directory not
empty paul@debian8:~$ rmdir mydir/stuff
paul@debian8:~$ rmdir mydir
paul@debian8:~$
```

- **mkdir -p**

Dan mirip dengan opsi **mkdir -p**, Anda juga dapat menggunakan **rmdir** untuk menghapus direktori secara rekursif.

```
paul@debian8:~$ mkdir -p
test42/subdir paul@debian8:~$ rmdir
-p test42/subdir paul@debian8:~$
```

2. Work with file

Dalam bab ini kita belajar bagaimana mengenali, membuat, menghapus, menyalin dan memindahkan file menggunakan perintah seperti file, sentuh, rm, cp, mv dan rename. Screenshot ini menunjukkan perbedaan antara dua file, satu dengan huruf W, yang lain dengan huruf kecil w.

```
paul@laika:~/Linux$ ls
winter.txt Winter.txt
paul@laika:~/Linux$ cat
winter.txt It is cold.
paul@laika:~/Linux$ cat Winter.txt
It is very cold!
```

Berikut beberapa contoh pada sistem Linux yang khas.

```
paul@laika:~$ file pic33.png
pic33.png: PNG image data, 3840 x 1200, 8-bit/color RGBA, non-
interlaced paul@laika:~$ file /etc/passwd
/etc/passwd: ASCII text
paul@laika:~$ file HelloWorld.c
HelloWorld.c: ASCII C program
text
```

Perintah file menggunakan file ajaib yang berisi pola untuk mengenali jenis file. File magic terletak di **/usr / share / file / magic**. Ketikkan man 5 magic untuk informasi lebih lanjut.

Sangat menarik untuk menunjukkan **file -s** untuk file-file khusus seperti yang ada di **/ dev** dan **/ proc**.

```
root@debian6~# file /dev/sda
```

```

/dev/sda: block special
root@debian6~# file -s
/dev/sda
/dev/sda: x86 boot sector; partition 1: ID=0x83, active,
starthead... root@debian6~# file /proc/cpuinfo
/proc/cpuinfo: empty
root@debian6~# file -s /proc/cpuinfo
/proc/cpuinfo: ASCII C++ program text

```

- create an empty file

Salah satu cara mudah untuk membuat file kosong adalah dengan **touch**.

Screenshot ini dimulai dengan direktori kosong, membuat dua file dengan **touch** dan daftar file-file tersebut.

```

paul@debian7:~$ ls
-l
total
0
paul@debian7:~$ touch
file42 paul@debian7:~$
touch file33
paul@debian7:~$ ls -l
total
0
-rw-r--r-- 1 paul paul 0 Oct 15 08:57
file33
-rw-r--r-- 1 paul paul 0 Oct 15 08:56
file42 paul@debian7:~$

```

- **Touch -t**

Perintah **touch** dapat mengatur beberapa properti saat membuat file kosong. Dapatkah Anda menentukan apa yang ditetapkan dengan melihat tangkapan layar berikutnya? Jika tidak, periksa manual untuk **touch**.

```

paul@debian7:~$ touch -t 200505050000 SinkoDeMayo
paul@debian7:~$ touch -t 130207111630
BigBattle.txt paul@debian7:~$ ls -l
total 0
-rw-r--r-- 1 paul paul 0 Jul 11 1302 BigBattle.txt
-rw-r--r-- 1 paul paul 0 Oct 15 08:57 file33
-rw-r--r-- 1 paul paul 0 Oct 15 08:56 file42
-rw-r--r-- 1 paul paul 0 May 5 2005
SinkoDeMayo paul@debian7:~$

```

- Remove Forever

Saat Anda tidak lagi membutuhkan file, gunakan **rm** untuk menghapusnya. Tidak seperti beberapa antarmuka pengguna grafis, baris perintah pada umumnya tidak memiliki **tempat sampah** atau **tempat sampah** untuk memulihkan file. Ketika Anda menggunakan **rm** untuk menghapus file, file tersebut hilang. Karena itu, berhati-hatilah saat menghapus file!

```

paul@debian7:~$ ls
BigBattle.txt file33 file42
SinkoDeMayo paul@debian7:~$ rm
BigBattle.txt paul@debian7:~$ ls
file33 file42
SinkoDeMayo
paul@debian7:~$

```

- **rm -i**

Untuk mencegah diri Anda secara tidak sengaja menghapus file, Anda dapat mengetikkan **rm -i**.

```
paul@debian7:~$  
ls  
file33  file42  
SinkoDeMayo  
paul@debian7:~$ rm -i  
file33  
rm: remove regular empty file `file33'?  
yes paul@debian7:~$ rm -i SinkoDeMayo  
rm: remove regular empty file `SinkoDeMayo'?  
n paul@debian7:~$ ls  
file42  
SinkoDeMayo  
paul@debian7:~$
```

- **rm -rf**

Secara default, **rm -r** tidak akan menghapus direktori yang tidak kosong. Namun **rm** menerima beberapa opsi yang memungkinkan Anda menghapus direktori apa pun. Pernyataan **rm-rf** terkenal karena akan menghapus apa pun (asalkan Anda memiliki izin untuk melakukannya). Ketika Anda login sebagai root, berhati-hatilah dengan **rm -rf** (f berarti **memaksa** dan r berarti rekursif) karena menjadi root menyiratkan bahwa izin tidak berlaku untuk Anda. Anda benar-benar dapat menghapus seluruh sistem file Anda secara tidak sengaja.

```
paul@debian7:~$ mkdir test  
paul@debian7:~$ rm test  
rm: cannot remove `test': Is a  
directory paul@debian7:~$ rm -rf test  
paul@debian7:~$ ls test  
ls: cannot access test: No such file or  
directory paul@debian7:~$
```

- **copy one file**

Untuk menyalin file, gunakan cp dengan sumber dan argumen target.

```
paul@debian7:~$ ls  
file42  SinkoDeMayo  
paul@debian7:~$ cp file42 file42.copy  
paul@debian7:~$ ls  
file42  file42.copy  SinkoDeMayo
```

- **copy to another directory**

Jika targetnya adalah direktori, maka file sumber disalin ke direktori target itu.

```
paul@debian7:~$ mkdir dir42  
paul@debian7:~$ cp SinkoDeMayo  
dir42 paul@debian7:~$ ls dir42/  
SinkoDeMayo
```

- **cp -r**

Untuk menyalin direktori lengkap, gunakan cp -r (opsi -r memaksa menyalin rekursif dari semua file di semua subdirektori).

```
paul@debian7:~$ ls  
dir42  file42  file42.copy  
SinkoDeMayo paul@debian7:~$ cp -r  
dir42/ dir33 paul@debian7:~$ ls
```

```
dir33 dir42 file42 file42.copy
SinkoDeMayo paul@debian7:~$ ls dir33/
SinkoDeMayo
```

- **cp multiple files ke directory**

Anda juga dapat menggunakan cp untuk menyalin beberapa file ke dalam direktori. Dalam hal ini, argumen terakhir (a.k.a. target) harus berupa direktori.

```
paul@debian7:~$ cp file42 file42.copy SinkoDeMayo dir42/
paul@debian7:~$ ls dir42/
file42 file42.copy SinkoDeMayo
```

- **cp -i**

Untuk mencegah cp menimpa file yang ada, gunakan opsi -i (untuk interaktif).

```
paul@debian7:~$ cp SinkoDeMayo file42
paul@debian7:~$ cp SinkoDeMayo file42
paul@debian7:~$ cp -i SinkoDeMayo
file42 cp: overwrite `file42'? n
paul@debian7:~$
```

- **rename files dengan mv**

Gunakan mv untuk mengganti nama file atau memindahkan file ke direktori lain.

```
paul@debian7:~$ ls
dir33 dir42 file42 file42.copy
SinkoDeMayo paul@debian7:~$ mv file42 file33
paul@debian7:~$ ls
dir33 dir42 file33 file42.copy
SinkoDeMayo paul@debian7:~$
```

Ketika Anda perlu mengubah nama hanya satu file maka mv adalah perintah yang disukai untuk digunakan.

- **rename directories dengan mv**

Perintah mv yang sama dapat digunakan untuk mengganti nama direktori.

```
paul@debian7:~$ ls -l
total 8
drwxr-xr-x 2 paul paul 4096 Oct 15 09:36
dir33 drwxr-xr-x 2 paul paul 4096 Oct 15
09:36 dir42
-rw-r--r-- 1 paul paul 0 Oct 15 09:38 file33
-rw-r--r-- 1 paul paul 0 Oct 15 09:16 file42.copy
-rw-r--r-- 1 paul paul 0 May 5 2005
SinkoDeMayo paul@debian7:~$ mv dir33 backup
paul@debian7:~$ ls -l
total 8
drwxr-xr-x 2 paul paul 4096 Oct 15 09:36
backup drwxr-xr-x 2 paul paul 4096 Oct 15
09:36 dir42
-rw-r--r-- 1 paul paul 0 Oct 15 09:38 file33
-rw-r--r-- 1 paul paul 0 Oct 15 09:16 file42.copy
-rw-r--r-- 1 paul paul 0 May 5 2005
SinkoDeMayo paul@debian7:~$
```

- **mv -i**

Mv juga memiliki -i switch yang mirip dengan cp dan rm.

screenshot ini menunjukkan bahwa mv -i akan meminta izin untuk menimpa file yang ada.

```
paul@debian7:~$ mv -i file33
SinkoDeMayo mv: overwrite
`SinkoDeMayo'? no paul@debian7:~$
```


- **rename**

Perintah **rename** pada Debian menggunakan ekspresi reguler (ekspresi reguler atau shor regex dijelaskan dalam bab selanjutnya) untuk mengganti nama banyak file sekaligus

Di bawah ini **rename** contoh yang mengubah semua kemunculan txt menjadi png untuk semua nama file yang diakhiri dengan .txt.

```
paul@debian7:~/test42$ ls abc.txt file33.txt
file42.txt paul@debian7:~/test42$ rename
's/\.txt/\.png/' *.txt paul@debian7:~/test42$ ls
abc.png file33.png file42.png
```

Contoh kedua ini mengalihkan semua (pertama) kemunculan file ke dalam dokumen untuk semua nama file yang diakhiri dengan .png.

```
paul@debian7:~/test42$ ls
abc.png file33.png file42.png
paul@debian7:~/test42$ rename 's/file/document/' *.png
paul@debian7:~/test42$ ls
abc.png document33.png
document42.png paul@debian7:~/test42$
```

E. Latihan

Work dengan directories

1. Tampilkan direktori Anda saat ini.
2. Ubah ke direktori / etc.
3. Sekarang ubah ke direktori home Anda hanya dengan menekan tiga tombol.
4. Ubah ke direktori / boot / grub menggunakan hanya sebelas tombol yang ditekan.
5. Pergi ke direktori induk dari direktori saat ini.
6. Pergi ke direktori root.
7. Daftar isi direktori root.
8. Buat daftar panjang direktori root.
9. Tetap di mana Anda berada, dan daftar isi dari / etc.
10. Tetap di mana Anda berada, dan daftar isi / bin dan / sbin.
11. Tetap di mana Anda berada, dan daftar isi ~.
12. Buat daftar semua file (termasuk file tersembunyi) di direktori home Anda.
13. Buat daftar file dalam / boot dalam format yang dapat dibaca manusia.
14. Buat direktori testdir di direktori home Anda.
15. Ubah ke direktori / etc, tetap di sini dan buat direktori newdir di direktori home Anda.
16. Buat dalam satu perintah direktori ~ / dir1 / dir2 / dir3 (dir3 adalah subdirektori dari dir2, dan dir2 adalah subdirektori dari dir1).
17. Hapus direktori testdir.
18. Jika waktu mengizinkan (atau jika Anda menunggu siswa lain untuk menyelesaikan latihan ini), gunakan dan pahami pushd dan popd. Gunakan halaman manual bash untuk menemukan informasi tentang perintah ini.

Work dengan file

1. Buat daftar file dalam direktori / bin
2. Tampilkan jenis file / bin / cat, / etc / passwd dan / usr / bin / passwd.

3. Unduh wolf.jpg dan LinuxFun.pdf dari <http://linux-training.be> (wget <http://linux-training.be/files/studentfiles/wolf.jpg> dan wget <http://linux-training.be/files/books/LinuxFun.pdf>)

```
wget http://linux-training.be/files/studentfiles/wolf.jpg wget http://linux-training.be/files/studentfiles/wolf.png wget http://linux-training.be/files/books/LinuxFun.pdf
```

4. Tampilkan jenis file wolf.jpg dan LinuxFun.pdf
5. Ganti nama wolf.jpg menjadi wolf.pdf (gunakan mv).
6. Tampilkan jenis file wolf.pdf dan LinuxFun.pdf.
7. Buat direktori ~ / sentuh dan masukkan.
8. Buat file today.txt dan kemarin.txt dalam sentuhan.
9. Ubah tanggal pada kemarin.txt untuk mencocokkan tanggal kemarin.
10. Salin kemarin.txt ke copy.yesterday.txt
11. Ubah nama copy.yesterday.txt menjadi kim
12. Buat direktori bernama ~ / testbackup dan salin semua file dari ~ / sentuh ke dalamnya.
13. Gunakan satu perintah untuk menghapus direktori ~ / testbackup dan semua file ke dalamnya.
14. Buat direktori ~ / etcbackup dan salin semua file * .conf dari / etc ke dalamnya. Apakah Anda memasukkan semua subdirektori dari / etc?
15. Gunakan ganti nama untuk mengganti nama semua file * .conf menjadi * .backup. (jika Anda memiliki lebih dari satu distro, cobalah semuanya!)

Bagian 3

basic Unix tools

A. Tujuan

- Siswa memahami penggunaan perintah dasar (mencari, menemukan file dan mengompresi file) menggunakan mode perintah

B. Teori

Bab ini memperkenalkan perintah untuk mencari atau menemukan file dan untuk mengompres file, bersama dengan alat umum lainnya yang tidak dibahas sebelumnya. Sementara alat yang dibahas di sini secara teknis tidak dianggap **filters**, mereka dapat digunakan dalam **pipes**.

- **find**

Perintah **find** bisa sangat berguna pada awal sebuah pipe untuk mencari file. Inilah beberapa contoh. Anda mungkin ingin menambahkan **2> / dev / null** ke baris perintah untuk menghindari mengacaukan layar Anda dengan pesan kesalahan.

- **locate**

Perintah **locate** sangat berbeda dari **find** dalam menggunakan indeks untuk mencari file. Ini jauh lebih cepat daripada melintasi semua direktori. Jika indeks belum ada, maka Anda harus membuatnya (sebagai root pada Red Hat Enterprise Linux) dengan perintah **updatedb**.

C. Alat dan Bahan

- Komputer
- Virtual Box dengan Linuk OS

D. Praktik (Minggu 5)

- **find**

Temukan semua file di / etc dan masukkan daftar di etcfiles.txt

```
find /etc > etcfiles.txt
```

Temukan semua file dari seluruh sistem dan masukkan daftarnya ke allfiles.txt

```
find / > allfiles.txt
```

Temukan file yang berakhiran .conf di direktori saat ini (dan semua subdir).

```
find . -name "*.conf"
```

Temukan file jenis file (bukan direktori, pipe, dll.) Yang diakhiri dengan .conf.

```
find . -type f -name "*.conf"
```

Temukan file dari direktori jenis yang berakhiran .bak.

```
find /data -type d -name "*.bak"
```

Temukan file yang lebih baru daripada file42.txt

```
find . -newer file42.txt
```

Temukan juga dapat mengeksekusi perintah lain pada setiap file yang ditemukan. Contoh ini akan mencari *.odf files and copy them to /backup/.

```
find /data -name "*.odf" -exec cp {} /backup/ \;
```

Temukan juga dapat mengeksekusi, setelah konfirmasi Anda, perintah lain pada setiap file yang ditemukan. Contoh ini akan menghapus file * .odf jika Anda menyetujuinya untuk setiap file yang ditemukan.

```
find /data -name "*.odf" -ok rm {} \
```

- **locate**

```
[paul@RHEL4b ~]$ locate Samba
warning: locate: could not open database:
/var/lib/slocate/slocate.db:... warning: You need to run the 'updatedb'
command (as root) to create th... Please have a look at
/etc/updatedb.conf to enable the daily cron job. [paul@RHEL4b ~]$
updatedb
fatal error: updatedb: You are not authorized to create a default
sloc... [paul@RHEL4b ~]$ su -
Password:
[root@RHEL4b ~]# updatedb
[root@RHEL4b ~]#
```

Kebanyakan distribusi Linux akan menjadwalkan **updatedb** untuk dijalankan sekali setiap hari

- **date**

Perintah **date** dapat menampilkan tanggal, waktu, zona waktu, dan banyak lagi.

```
paul@rhel55 ~$ date
Sat Apr 17 12:44:30 CEST 2010
```

Sebuah **Date String** dapat disesuaikan untuk menampilkan format pilihan Anda. Periksa halaman manual untuk opsi lainnya.

```
paul@rhel55 ~$ date +%A %d-%m-%Y
Saturday 17-04-2010
```

Waktu pada setiap Unix dihitung dalam jumlah detik sejak 1969 (detik pertama menjadi detik pertama dari pertama Januari 1970). Gunakan **date +%s** untuk menampilkan waktu Unix dalam hitungan detik.

```
paul@rhel55 ~$ date +%s
1271501080
```

Kapan penghitungan ini mencapai dua ribu juta detik?

```
paul@rhel55 ~$ date -d '1970-01-01 + 2000000000
seconds' Wed May 18 04:33:20 CEST 2033
```

- **cal**

Perintah **cal** menampilkan bulan saat ini, dengan hari ini disorot.

```
paul@rhel55 ~$ cal
      April 2010
Su Mo Tu We Th Fr Sa
                1  2  3
    5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

Anda dapat memilih bulan apa pun di masa lalu atau masa depan.

```
paul@rhel55 ~$ cal 2 1970
      February 1970
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

- **sleep**

The **sleep** command is sometimes used in scripts to wait a number of seconds. This example shows a five second **sleep**.

Perintah **sleep** terkadang digunakan dalam skrip untuk menunggu beberapa detik. Contoh ini menunjukkan sleep lima detik

```
paul@rhel55 ~$ sleep
5 paul@rhel55 ~$
```

- **time**

Perintah **time** dapat menampilkan berapa lama waktu yang diperlukan untuk menjalankan perintah. Perintah **date** hanya membutuhkan sedikit waktu.

```
paul@rhel55 ~$ time
date
Sat Apr 17 13:08:27 CEST
2010
```

```
real    0m0.014s
user    0m0.008s
sys     0m0.006s
```

Perintah **Sleep 5** membutuhkan lima detik untuk dieksekusi, tetapi menghabiskan sedikit **cpu time**.

```
paul@rhel55 ~$ time sleep
5
```

```
real    0m5.018s
user    0m0.005s
sys     0m0.011s
```

Perintah **bzip2** ini mengompres file dan membutuhkan banyak **cpu time**

```
paul@rhel155 ~$ time bzip2
text.txt

real    0m2.368s
user    0m0.847s
sys      0m0.539s
```

- **gzip - gunzip**

Pengguna tidak pernah memiliki cukup ruang disk, jadi kompresi sangat berguna.

Perintah **gzip** dapat membuat file mengambil lebih sedikit ruang.

```
paul@rhel155 ~$ ls -lh text.txt
-rw-rw-r-- 1 paul paul 6.4M Apr 17 13:11
text.txt
paul@rhel155 ~$ gzip text.txt
paul@rhel155 ~$ ls -lh text.txt.gz
-rw-rw-r-- 1 paul paul 760K Apr 17 13:11 text.txt.gz
```

Anda bisa mendapatkan kembali asli dengan **gunzip**

```
paul@rhel155 ~$ gunzip
text.txt.gz
paul@rhel155 ~$ ls -
lh text.txt
-rw-rw-r-- 1 paul paul 6.4M Apr 17 13:11 text.txt
```

- **zcat - zmore**

File teks yang dikompresi dengan **gzip** dapat dilihat dengan **zcat** dan **zmore**.

```
paul@rhel155 ~$ head -4 text.txt
/
/opt
/opt/VBoxGuestAdditions-3.1.6
/opt/VBoxGuestAdditions-
3.1.6/routines.sh
paul@rhel155 ~$ gzip
text.txt
paul@rhel155 ~$ zcat text.txt.gz | head -4
/
/opt
/opt/VBoxGuestAdditions-3.1.6
/opt/VBoxGuestAdditions-3.1.6/routines.sh
```

- **bzip2 - bunzip2**

File juga dapat dikompres dengan **bzip2** yang membutuhkan sedikit lebih banyak waktu daripada **gzip**, tetapi kompres lebih baik.

```
paul@rhel155 ~$ bzip2 text.txt
paul@rhel155 ~$ ls -lh
text.txt.bz2
-rw-rw-r-- 1 paul paul 569K Apr 17 13:11 text.txt.bz2
```

File dapat uncompressed lagi dengan **bunzip2**.

```
paul@rhel155 ~$ bunzip2
text.txt.bz2
paul@rhel155 ~$ ls -
lh text.txt
-rw-rw-r-- 1 paul paul 6.4M Apr 17 13:11 text.txt
```

- **bzcat - bzmores**

Dan dengan cara yang sama **bzcat** dan **bzmores** dapat menampilkan file yang dikompres dengan **bzip2**.

```
paul@rhel155 ~$ bzip2 text.txt
paul@rhel155 ~$ bzcat text.txt.bz2 | head -4
/
```

```
/opt
/opt/VBoxGuestAdditions-3.1.6
/opt/VBoxGuestAdditions-3.1.6/routines.sh
```

E. Latihan

1. Jelaskan perbedaan antara dua perintah ini. Pertanyaan ini sangat penting. Jika Anda tidak tahu jawabannya, lihat kembali bab **shell**.

```
find /data -name "*.txt"
```

```
find /data -name *.txt
```

2. Jelaskan perbedaan antara dua pernyataan ini. Akankah mereka berdua bekerja ketika ada 200 **.odf** file di **/data**? Bagaimana kalau ada 2 juta file **.odf**?

```
find /data -name "*.odf" > data_odf.txt
```

```
find /data/*.odf > data_odf.txt
```

3. Tulis perintah **find** yang menemukan semua file yang dibuat setelah 30 Januari 2010
4. Tulis perintah **find** yang menemukan semua file ***.odf** yang dibuat pada bulan September 2009.
5. Hitung jumlah file di ***.conf** / **etc** dan semua subdirektornya.
6. Berikut adalah dua perintah yang melakukan hal yang sama: salin ***.odf** file ke **/backup** /. Apa yang menjadi alasan untuk mengganti perintah pertama dengan yang kedua? Sekali lagi, ini adalah pertanyaan penting.

```
cp -r /data/*.odf /backup/
```

```
find /data -name "*.odf" -exec cp {} /backup/ \;
```

7. Buat file bernama **loctest.txt**. Dapatkah Anda menemukan file ini dengan **locate**? Kenapa tidak ? Bagaimana Anda menemukan file ini?
8. Gunakan **find** dan **-exec** untuk mengganti nama semua file **.htm** menjadi **.html**.
9. Keluarkan perintah **tanggal**. Sekarang tampilkan tanggal dalam format **YYYY / MM / DD**.
10. Terbitkan perintah **cal**. Menampilkan kalender 1582 dan 1752. Perhatikan sesuatu yang istimewa?

Bagian 4

Standard File Permissions

A. Tujuan

- Mahasiswa memahami berbagai jenis kepemilikan file dan izin file di Sistem Operasi Linux
- Mahasiswa dapat memanipulasi izin file yang diperlukan pada Sistem Operasi Linux

B. Teori

Bab ini berisi detail tentang keamanan file dasar melalui **file ownership** dan **file permissions**.

Table Unix special files

first character	file type
-	normal file
d	directory
l	symbolic link
p	named pipe
b	block device
c	character device
s	socket

standard Unix file permissions

Sembilan karakter yang mengikuti jenis file menunjukkan izin dalam tiga triplet. Izin dapat berupa **r** untuk akses baca **w** untuk akses tulis, dan **x** untuk dieksekusi. Anda membutuhkan **r** izin untuk daftar (**ls**) isi dari suatu direktori. Anda memerlukan **x** izin untuk memasukkan (**cd**) direktori. Anda memerlukan **w** izin untuk membuat file di atau menghapus file dari direktori.

Tabel standard Unix file permissions

permission	on a file	on a directory
r (read)	read file contents (cat)	read directory contents (ls)
w (write)	change file contents (vi)	create files in (touch)
x (execute)	execute the file	enter the directory (cd)

```
paul@RHELv4u4:~/test$ ls -l proc42.bash
-rwxr-xr-- 1 paul proj 984 Feb  6 12:01 proc42.bash
```

Di bawah ini adalah tabel yang menjelaskan fungsi dari semua sepuluh

Tabel Unix file permissions position

position	characters	function
1	-	this is a regular file
2-4	rwX	permissions for the user owner
5-7	r-X	permissions for the group owner
8-10	r--	permissions for others

Ketika Anda adalah **user owner** file, maka the **user owner permissions** berlaku untuk Anda. Sisa izin tidak berpengaruh pada akses Anda ke file.

Ketika Anda termasuk **group** yang merupakan pemilik grup file, izin the **group owner** berlaku untuk Anda. Sisa izin tidak berpengaruh pada akses Anda ke file.

Ketika Anda bukan **user owner** file dan Anda bukan milik the **group owner**, maka, **others permissions** berlaku untuk Anda.

Pengaturan octal permissions

Most Unix administrators will use the **old school** octal system to talk about and set permissions. Look at the triplet bitwise, equating r to 4, w to 2, and x to 1.

Kebanyakan administrator Unix akan menggunakan sistem oktal **old school** untuk membicarakan dan mengatur izin. Lihatlah triplet bitwise, menyamakan r ke 4, w ke 2, dan x ke 1

Tabel Octal permissions

binary	octal	permission
000	0	---
001	1	--X
010	2	-W-
011	3	-WX
100	4	r--
101	5	r-X
110	6	rw-
111	7	rwX

Ini membuat 777 sama dengan rwxrwxrwx dan dengan logika yang sama, 654 rw-r-xr--. The **chmod** perintah akan menerima angka-angka ini.

C. Alat dan Bahan

- Komputer
- Virtual Box dengan Instalasi Linux OS

D. Praktik (minggu 6 & 7)

1. file ownership

- **user owner and group owner**

users dan **groups** sistem dapat dikelola secara lokal di **/etc/passwd** dan **/etc/group**, atau mereka dapat berada di NIS, LDAP, atau Samba domain. Pengguna dan grup ini dapat **memiliki** file. Sebenarnya, setiap file memiliki **user owner** dan **group owner**, seperti yang bisa dilihat di screenshot berikut.

```
paul@rhel65:~/owners$ ls -lh total 636K-rw-r--r--. 1 paul snooker 1.1K Apr 8 18:47 data.odt-rw-r--r--. 1 paul paul 626K Apr 8 18:46 file1-rw-r--r--. 1 root tennis 185 Apr 8 18:46 file2-rw-rw-r--. 1 root root 0 Apr 8 18:47 stuff.txt paul@rhel65:~/owners$
```

Pengguna paul memiliki tiga file; file1 paul sebagai **user owner** dan **group owner** paul sebagai **group owner**, data.odt adalah **grup yang dimiliki** oleh grup snooker, file2 oleh grup tenis. File terakhir disebut stuff.txt dan dimiliki oleh pengguna root dan grup root.

- **listing user accounts**

Anda dapat menggunakan perintah berikut untuk mendaftarkan semua akun pengguna lokal.

```
paul@debian7~$ cut -d: -f1 /etc/passwd | column
root          ntp          sam          bert         naomi
daemon        mysql        tom          rino
matthias2 bin      paul        wouter      antonio
bram
sys           maarten     robrecht    simon        fabrice
sync         kevin       bilal       sven         chimene
games        yuri        dimitri     wouter2
messagebus man    william     ahmed       tarik
roger
lp           yves        dylan      jan
frank mail    kris       robin       ian
toon news    hamid      matthias    ivan
rinus uucp    vladimir  ben         azeddine
eddy proxy    abiy       mike        eric
bram2 www-data  david      kevin2      kamel
keith backup  chahid     kenzo       ischa
jesse
list         stef        aaron       bart
frederick irc    joeri      lorenzo     omer
hans
gnats        glenn       jens        kurt
dries nobody  yannick    ruben       steve
steve2 libuuid    christof    jelle       constantin
tomas Debian-exim  george     stefaan     sam2
johan statd   joost      marc        bjorn
tom2 sshd     arno       thomas      ronald
```

- **chgrp**

Anda dapat mengubah pemilik grup file menggunakan perintah **chgrp**.

```

root@rhel65:/home/paul/owners# ls -l file2
-rw-r--r--. 1 root tennis 185 Apr  8 18:46 file2
root@rhel65:/home/paul/owners# chgrp snooker
file2 root@rhel65:/home/paul/owners# ls -l file2
-rw-r--r--. 1 root snooker 185 Apr  8 18:46
file2 root@rhel65:/home/paul/owners#

```

- **chown**

Pemilik pengguna file dapat diubah dengan perintah **chown**.

```

root@laika:/home/paul# ls -l FileForPaul
-rw-r--r-- 1 root paul 0 2008-08-06 14:11
FileForPaul root@laika:/home/paul# chown paul
FileForPaul root@laika:/home/paul# ls -l
FileForPaul
-rw-r--r-- 1 paul paul 0 2008-08-06 14:11 FileForPaul

```

Anda juga dapat menggunakan **chown** untuk mengubah pemilik pengguna dan pemilik grup

```

root@laika:/home/paul# ls -l FileForPaul
-rw-r--r-- 1 paul paul 0 2008-08-06 14:11 FileForPaul
root@laika:/home/paul# chown root:project42
FileForPaul root@laika:/home/paul# ls -l FileForPaul
-rw-r--r-- 1 root project42 0 2008-08-06 14:11 FileForPaul

```

2. Unix special files

berikut screenshot dari sebuah character device (console) dan sebuah block device (hard disk).

```

paul@debian6lt~$ ls -ld /dev/console /dev/sda
crw----- 1 root root  5, 1 Mar 15 12:45
/dev/console brw-rw---- 1 root disk  8, 0 Mar 15
12:45 /dev/sda

```

dan bisa dilihat sebuah director, regular file dan symbolic link.

```

paul@debian6lt~$ ls -ld /etc /etc/hosts
/etc/motd drwxr-xr-x 128 root root 12288 Mar 15
18:34 /etc
-rw-r--r-- 1 root root  372 Dec 10 17:36 /etc/hosts
lrwxrwxrwx 1 root root  13 Dec  5 10:36 /etc/motd -> /var/run/motd

```

3. Contoh permission

Beberapa contoh kombinasi pada file dan direktori terlihat pada tangkapan layar ini. Nama file menjelaskan izin

```

paul@laika:~/perms$ ls -
lh total 12K
drwxr-xr-x  2 paul paul  4.0K  2007-02-07  22:26
AllEnter_UserCreateDelete
-rwxrwxrwx  1 paul paul      0  2007-02-07  22:21
EveryoneFullControl.txt
-r--r----- 1 paul paul      0  2007-02-07  22:21
OnlyOwnersRead.txt
-rwxrwx--- 1 paul paul  0 2007-02-07 22:21
OwnersAll_RestNothing.txt dr-xr-x--- 2 paul paul 4.0K 2007-02-07
22:25 UserAndGroupEnter
dr-x----- 2 paul paul 4.0K 2007-02-07 22:25
OnlyUserEnter paul@laika:~/perms$

```

Untuk meringkas, triplet **rwX** pertama mewakili izin untuk **pemilik pengguna**. Triplet kedua sesuai dengan **pemilik grup**; itu menentukan izin untuk semua anggota grup itu.

Triplet ketiga mendefinisikan izin untuk semua pengguna **lain** yang bukan pemilik pengguna dan bukan anggota pemilik grup.

4. Pengaturan Permission

Permissions can be changed with **chmod**. The first example gives the user owner execute permissions. Izin dapat diubah dengan **chmod**. Contoh pertama memberi pemilik pengguna untuk izin execute.

```
paul@laika:~/perms$ ls -l permissions.txt
-rw-r--r-- 1 paul paul 0 2007-02-07 22:34
permissions.txt paul@laika:~/perms$ chmod u+x
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rwxr--r-- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

Contoh ini menghilangkan izin baca pemilik grup

```
paul@laika:~/perms$ chmod g-r
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rwx---r-- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

Contoh ini menghilangkan izin baca orang lain.

```
paul@laika:~/perms$ chmod o-r
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rwx----- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

Contoh ini memberi mereka semua izin menulis .

```
paul@laika:~/perms$ chmod a+w
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rwx-w--w- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

Anda bahkan tidak perlu mengetik a.

```
paul@laika:~/perms$ chmod +x
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rwx-wx-wx 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

Anda juga dapat mengatur izin eksplisit.

```
paul@laika:~/perms$ chmod u=rw
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rw--wx-wx 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

Jangan ragu untuk membuat kombinasi apa pun.

```
paul@laika:~/perms$ chmod u=rw,g=rw,o=r
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rw-rw-r-- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

Bahkan kombinasi amis diterima oleh chmod.

```
paul@laika:~/perms$ chmod u=rwx,ug+rw,o=r
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rwxrw-r-- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

- **Izin Oktal**

```
paul@laika:~/perms$ chmod 777
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rwxrwxrwx 1 paul paul 0 2007-02-07 22:34
permissions.txt paul@laika:~/perms$ chmod 664
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rw-rw-r-- 1 paul paul 0 2007-02-07 22:34
permissions.txt paul@laika:~/perms$ chmod 750
permissions.txt paul@laika:~/perms$ ls -l
permissions.txt
-rwxr-x--- 1 paul paul 0 2007-02-07 22:34 permissions.tx
```

- **umask**

Saat membuat file atau direktori, seperangkat izin default diterapkan. Izin default ini ditentukan oleh **umask**. **Umask** menetapkan izin yang Anda tidak ingin diaktifkan secara default. Anda dapat menampilkan **umask** dengan perintah **umask**.

```
[Harry@RHEL4b ~]$
umask
00
02
[Harry@RHEL4b ~]$ touch
test
[Harry@RHEL4b ~]$ ls -l
test
-rw-rw-r-- 1 Harry Harry 0 Jul 24 06:03
test
[Harry@RHEL4b ~]$
```

Seperti yang Anda lihat, file juga tidak dapat dijalankan secara default. Ini adalah fitur keamanan umum di antara Unix; file yang baru dibuat tidak pernah dapat dieksekusi secara default. Anda harus secara eksplisit melakukan **chmod + x** untuk membuat file dapat dieksekusi. Ini juga berarti bahwa 1 bit dalam **umask** tidak memiliki arti - **umask** 0022 sama dengan 0033.

- **mkdir -m**

Saat membuat direktori dengan **mkdir**, Anda dapat menggunakan opsi **-m** untuk mengatur **mode**. Screenshot ini menjelaskan.

```
paul@debian5~$ mkdir -m 700 MyDir
paul@debian5~$ mkdir -m 777 Public
paul@debian5~$ ls -dl MyDir/
Public/
drwx----- 2 paul paul 4096 2011-10-16 19:16 MyDir/
drwxrwxrwx 2 paul paul 4096 2011-10-16 19:16 Public/
```

- **cp -p**

To preserve permissions and time stamps from source files, use **cp -p**.

```

paul@laika:~/perms$ cp file* cp
paul@laika:~/perms$ cp -p file*
cpp paul@laika:~/perms$ ll *
-rwx----- 1 paul paul    0 2008-08-25 13:26 file33
-rwxr-x--- 1 paul paul    0 2008-08-25 13:26 file42

cp:
total 0
-rwx----- 1 paul paul  0 2008-08-25 13:34 file33
-rwxr-x--- 1 paul paul  0 2008-08-25 13:34 file42

cpp:
total 0
-rwx----- 1 paul paul  0 2008-08-25 13:26 file33
-rwxr-x--- 1 paul paul  0 2008-08-25 13:26 file42

```

- **Advanced permissions file**

1. **sticky bit pada directory**

Anda dapat **sticky bit** yang melekat pada direktori untuk mencegah pengguna menghapus file yang tidak mereka miliki sebagai pemilik pengguna. sticky bit ditampilkan di lokasi yang sama dengan izin x untuk yang lain. sticky bit diwakili oleh **t** (artinya x juga ada) atau **T** (ketika tidak ada x untuk yang lain).

```

root@RHELv4u4:~# mkdir
/project55 root@RHELv4u4:~# ls -
ld /project55
drwxr-xr-x  2 root root 4096 Feb  7 17:38
/project55 root@RHELv4u4:~# chmod +t /project55/
root@RHELv4u4:~# ls -ld /project55
drwxr-xr-t  2 root root 4096 Feb  7 17:38
/project55 root@RHELv4u4:~#

```

sticky bit juga dapat diatur dengan octal permissions, itu adalah biner 1 dalam yang pertama dari empat triplets.

```

root@RHELv4u4:~# chmod 1775 /project55/
root@RHELv4u4:~# ls -ld /project55
drwxrwxr-t  2 root root 4096 Feb  7 17:38
/project55 root@RHELv4u4:~#

```

Anda biasanya akan menemukan **sticky** pada direktori / **tmp**

```

root@barry:~# ls -ld
/tmp
drwxrwxrwt 6 root root 4096 2009-06-04 19:02
/tmp

```

2. **setgid bit pada directory**

setgid dapat digunakan pada direktori untuk memastikan bahwa semua file di dalam direktori dimiliki oleh pemilik grup direktori. Bit **setgid** ditampilkan di lokasi yang sama dengan izin x untuk pemilik grup. Bit **setgid** diwakili oleh **s** (artinya x juga ada) atau **S** (ketika tidak ada x untuk pemilik grup). Seperti contoh ini menunjukkan, meskipun **root** bukan milik grup proj55, file yang dibuat oleh root di / project55 akan menjadi milik proj55 karena **setgid** ditetapkan

```

root@RHELv4u4:~# groupadd proj55
root@RHELv4u4:~# chown root:proj55
/project55/ root@RHELv4u4:~# chmod 2775
/project55/ root@RHELv4u4:~# touch
/project55/fromroot.txt root@RHELv4u4:~# ls
-l /project55/
drwxrwsr-x  2 root proj55 4096 Feb  7 17:45 /project55/
root@RHELv4u4:~# ls -l /project55/
total 4
-rw-r--r--  1 root proj55 0 Feb  7 17:45
fromroot.txt root@RHELv4u4:~#

```

You can use the **find** command to find all **setgid** directories.

```

paul@laika:~$ find / -type d -perm -2000 2> /dev/null
/var/log/mysql
/var/log/news
/var/local
...

```

3. setgid dan setuid pada regular files

Kedua izin ini menyebabkan file eksekusi dieksekusi dengan izin dari **pemilik file** alih-alih **pemilik eksekusi**. Ini berarti bahwa jika ada pengguna yang menjalankan program yang menjadi milik **pengguna root**, dan bit **setuid** diatur pada program itu, maka program berjalan sebagai **root**. Ini bisa berbahaya, tetapi terkadang ini baik untuk keamanan.

Ambil contoh kata sandi; mereka disimpan di **/etc / shadow** yang hanya dapat dibaca oleh

akar. (Pengguna **root** tidak pernah membutuhkan izin.)

```

root@RHELv4u4:~# ls -l /etc/shadow
-r-----  1 root root 1260 Jan 21 07:49 /etc/shadow

```

Mengubah kata sandi Anda membutuhkan pembaruan file ini, jadi bagaimana pengguna non-root normal dapat melakukan ini? Mari kita lihat izin di **/usr/bin/passwd**.

```

root@RHELv4u4:~# ls -l /usr/bin/passwd
-r-s--x--x  1 root root 21200 Jun 17  2005 /usr/bin/passwd

```

Saat menjalankan program **passwd**, Anda mengeksekusinya dengan root. Anda dapat menggunakan perintah **find** untuk menemukan semua program **setuid**.

```

paul@laika:~$ find /usr/bin -type f -perm -04000
/usr/bin/arping
/usr/bin/kgrantpty
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/fping6
/usr/bin/passwd
/usr/bin/gpasswd
..

```

Dalam banyak kasus, pengaturan **setuid** bit pada executable sudah cukup. Mengatur **setuid** bit akan menghasilkan program-program ini untuk dijalankan dengan kredensial pemilik grup mereka.

4. **setuid on sudo**

Sudo binary memiliki setuid bit set, sehingga setiap pengguna dapat menjalankannya dengan userid root yang efektif.

```
paul@rhel65:~$ ls -l $(which sudo)
---s--x--x. 1 root root 123832 Oct  7 2013
/usr/bin/sudo paul@rhel65:~$
```

➤ **access control lists**

Perizinan Standar Unix mungkin tidak cukup untuk beberapa organisasi. Bab ini memperkenalkan **access control lists** atau akses untuk melindungi file dan direktori lebih lanjut.

- **acl in /etc/fstab**

Sistem file yang mendukung daftar kontrol akses, atau ACL, harus dipasang dengan opsi acl yang terdaftar di / etc / fstab. Pada contoh di bawah ini, Anda dapat melihat bahwa sistem file root memiliki dukungan acl, sedangkan / home / data tidak.

```
root@laika:~# tail -4 /etc/fstab
/dev/sda1 / ext3 acl,relatime 0 1
/dev/sdb2 /home/data auto noacl,defaults 0 0
pasha:/home/r /home/pasha nfs defaults 0 0
wolf:/srv/data /home/wolf nfs defaults 0 0
```

- **getfacl**

Membaca acl dapat dilakukan dengan / usr / bin / getfacl. Screenshot ini menunjukkan cara membaca acl file33 dengan getfacl.

```
paul@laika:~/test$ getfacl file33
# file: file33
# owner: paul
# group:
paul
user::rw-
group::r--
mask::rwx
other::r--
```

- **setfacl**

Menulis atau mengubah acl dapat dilakukan dengan / usr / bin / setfacl. Screenshot ini menunjukkan bagaimana mengubah acl file33 dengan setfacl.

Pertama kita tambahkan sandra pengguna dengan izin oktal 7 ke acl.

```
paul@laika:~/test$ setfacl -m u:sandra:7 file33
```

Kemudian kita tambahkan tenis kelompok dengan izin oktal 6 ke acl dari file yang sama.

```
paul@laika:~/test$ setfacl -m g:tennis:6 file33
```

Hasilnya terlihat dengan getfacl.

```
paul@laika:~/test$ getfacl file33
# file: file33
```



```
# owner: paul
# group: paul
user::rw-
user:sandra:rw
x group::r--
group:tennis:r
w- mask::rwx
other::r--
```

- **remove an acl entry**

Opsi -x dari perintah setfacl akan menghapus entri acl dari file yang ditargetkan.

```
paul@laika:~/test$ setfacl -m u:sandra:7
file33 paul@laika:~/test$ getfacl file33 |
grep sandra user:sandra:rwx
paul@laika:~/test$ setfacl -x sandra file33
paul@laika:~/test$ getfacl file33 | grep
sandra
```

Perhatikan bahwa menghilangkan u atau g saat mendefinisikan acl untuk akun akan menjadikannya default ke akun pengguna.

- **remove the complete acl**

Opsi -b dari perintah setfacl akan menghapus acl dari file yang ditargetkan.

```
paul@laika:~/test$ setfacl -b
file33 paul@laika:~/test$ getfacl
file33
# file: file33
# owner: paul
# group:
paul
user::rw-
group::r--
other::r--
```

- **the acl mask**

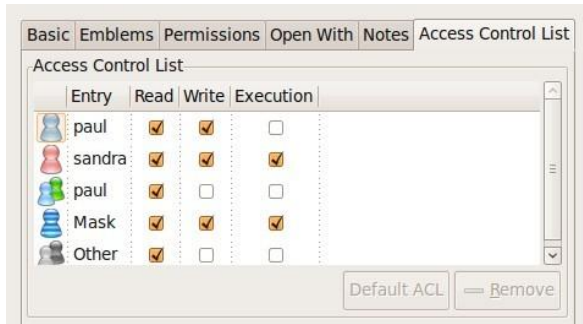
Mask acl mendefinisikan izin efektif maksimum untuk setiap entri di acl. Ini mask dihitung setiap kali Anda menjalankan perintah setfacl atau chmod.

Anda dapat mencegah perhitungan dengan menggunakan --no-mask switch.

```
paul@laika:~/test$ setfacl --no-mask -m u:sandra:7
file33 paul@laika:~/test$ getfacl file33
# file: file33
# owner: paul
# group:
paul
user::rw-
user:sandra:rwx
#effective:rw- group::r--
mask::rw
-
other::r
--
```

- **eiciel**

Pengguna desktop mungkin ingin menggunakan eiciel untuk mengelola acl dengan alat grafis.



Anda perlu menginstal eiciel dan nautilus-action untuk memiliki tab ekstra di nautilus untuk mengelola acls.

```
paul@laika:~$ sudo aptitude install eiciel nautilus-actions
```

➤ file links

Komputer rata-rata yang menggunakan Linux memiliki sistem file dengan banyak **hard links** dan **symbolic links**.

Untuk memahami tautan dalam sistem file, pertama-tama Anda harus memahami apa itu **inode** is.

1. inodes

• inode contents

The **ls -l** command will display some of the inode contents, as seen in this screenshot.

Inode adalah struktur data yang berisi metadata tentang file. Ketika sistem file menyimpan file baru di hard disk, menyimpan tidak hanya isi (data) file, tetapi juga properti tambahan seperti nama file, tanggal pembuatan, izinnya, pemilik file, dan banyak lagi. Semua informasi ini (kecuali nama file dan isi file) disimpan di inode file.

Perintah **ls-l** akan menampilkan beberapa isi inode, seperti yang terlihat pada screenshot ini

```
root@rhel53 ~# ls -ld
/home/project42/
drwxr-xr-x  4 root  pro42  4.0K  Mar  27  14:29
/home/project42/
```

• inode table

Tabel inode berisi semua inode dan dibuat ketika Anda membuat sistem file (dengan **mkfs**). Anda dapat menggunakan perintah **df -i** untuk melihat berapa banyak inode yang digunakan dan gratis pada sistem file yang di-mount.

```
root@rhel53 ~# df -i
Filesystem          Inodes    IUsed    IFree IUse% Mounted on
/dev/mapper/VolGroup00-LogVol100
4947968  115326  4832642    3% /
/dev/hda1            26104      45   26059    1% /boot
tmpfs                64417        1   64416    1% /dev/shm
/dev/sda1            262144    2207   259937    1% /home/project42
/dev/sdb1            74400    5519   68881    8% /home/project33
/dev/sdb5              0         0      0    - /home/sales
/dev/sdb6           100744     11  100733    1% /home/research
```

Dalam screenshot `df -i` di atas Anda dapat melihat penggunaan inode untuk beberapa sistem file yang dipasang. Anda tidak melihat angka untuk `/dev/sdb5` karena ini adalah sistem file yang gemuk.

- **inode number**

Setiap inode memiliki nomor unik (nomor inode). Anda dapat melihat nomor inode dengan perintah `ls -li`.

```
paul@RHELv4u4:~/test$ touch file1
paul@RHELv4u4:~/test$ touch file2
paul@RHELv4u4:~/test$ touch file3
paul@RHELv4u4:~/test$ ls -li
total 12
817266 -rw-rw-r-- 1 paul paul 0 Feb  5 15:38 file1
817267 -rw-rw-r-- 1 paul paul 0 Feb  5 15:38 file2
817268 -rw-rw-r-- 1 paul paul 0 Feb  5 15:38 file3
paul@RHELv4u4:~/test$
```

Ketiga file ini dibuat satu demi satu dan mendapat tiga inode berbeda (kolom pertama). Semua informasi yang Anda lihat dengan perintah `ls` ini berada di inode, kecuali untuk nama file (yang terkandung dalam direktori).

- **inode dan file contents**

Mari letakkan beberapa data di salah satu file.

```
paul@RHELv4u4:~/test$ ls -li
total 16
817266 -rw-rw-r-- 1 paul paul 0 Feb  5 15:38 file1
817270 -rw-rw-r-- 1 paul paul 92 Feb  5 15:42 file2
817268 -rw-rw-r-- 1 paul paul 0 Feb  5 15:38 file3
paul@RHELv4u4:~/test$ cat file2
It is winter now and it is very cold.
We do not like the cold, we prefer hot summer
nights.
paul@RHELv4u4:~/test$
```

Data yang ditampilkan oleh perintah `cat` tidak di inode, tetapi di tempat lain di disk. Inode berisi pointer ke data itu.

2. Tentang direktori

- **directory adalah sebuah tabel**

Direktori adalah jenis file khusus yang berisi tabel yang memetakan nama file ke inode. Daftar direktori kami saat ini dengan `ls -ali` akan menampilkan isi dari file direktori.

```
paul@RHELv4u4:~/test$ ls -ali
total 32
817262 drwxrwxr-x 2 paul paul 4096 Feb  5 15:42 .
800768 drwx----- 16 paul paul 4096 Feb  5 15:42 ..
817266 -rw-rw-r-- 1 paul paul 0 Feb  5 15:38 file1
817270 -rw-rw-r-- 1 paul paul 92 Feb  5 15:42 file2
817268 -rw-rw-r-- 1 paul paul 0 Feb  5 15:38 file3
paul@RHELv4u4:~/test$
```

3. hard links

- **Membuat hard links**

Saat membuat **hard link** ke file dengan `ln`, entri tambahan ditambahkan di direktori. Nama file baru dipetakan ke inode yang ada.

```
paul@RHELv4u4:~/test$ ln file2
hardlink_to_file2 paul@RHELv4u4:~/test$ ls -li
total
24
817266 -rw-rw-r-- 1 paul paul 0 Feb 5 15:38
file1
817270 -rw-rw-r-- 2 paul paul 92 Feb 5 15:42
file2
817268 -rw-rw-r-- 1 paul paul 0 Feb 5 15:38
file3
817270 -rw-rw-r-- 2 paul paul 92 Feb 5 15:42
hardlink_to_file2 paul@RHELv4u4:~/test$
```

Kedua file memiliki inode yang sama, sehingga mereka akan selalu memiliki izin yang sama dan pemilik yang sama. Kedua file akan memiliki konten yang sama. Sebenarnya, kedua file sekarang sama, artinya Anda dapat menghapus file asli dengan aman, file yang sudah di-hardlink akan tetap ada. Inode berisi penghitung, menghitung jumlah hard links ke dirinya sendiri. Ketika konter turun menjadi nol, maka inode dikosongkan.

- **Menemukan hard links**

Anda dapat menggunakan perintah `find` untuk mencari file dengan inode tertentu. Screenshot di bawah ini menunjukkan bagaimana mencari semua nama file yang mengarah ke inode 817270. Ingat bahwa nomor inode adalah unik untuk partisi-nya.

```
paul@RHELv4u4:~/test$ find / -inum 817270 2> /dev/null
/home/paul/test/file2
/home/paul/test/hardlink_to_file2
```

symbolic links

Symbolic links (kadang-kadang disebut **soft links**) tidak tertaut ke inode, tetapi buat nama untuk pemetaan nama. Tautan simbolis dibuat dengan `ln -s`. Seperti yang Anda lihat di bawah, Symbolic links mendapatkan inode sendiri.

```
paul@RHELv4u4:~/test$ ln -s file2
symlink_to_file2 paul@RHELv4u4:~/test$ ls -li
total
32
817273 -rw-rw-r-- 1 paul paul 13 Feb 5 17:06
file1
817270 -rw-rw-r-- 2 paul paul 106 Feb 5 17:04
file2
817268 -rw-rw-r-- 1 paul paul 0 Feb 5 15:38
file3
817270 -rw-rw-r-- 2 paul paul 106 Feb 5 17:04
hardlink_to_file2
817267 lrwxrwxrwx 1 paul paul 5 Feb 5 16:55 symlink_to_file2 ->
file2 paul@RHELv4u4:~/test$
```

Izin pada symbolic link tidak memiliki arti, karena izin dari target berlaku. Hard links terbatas pada partisi mereka sendiri (karena mereka menunjuk ke sebuah inode), symbolic link dapat menghubungkan di mana saja (sistem file lain, bahkan jaringan).

- **removing links**

Links dapat dihapus dengan **rm**.

```
paul@laika:~$ touch data.txt
paul@laika:~$ ln -s data.txt
sl_data.txt paul@laika:~$ ln data.txt
hl_data.txt paul@laika:~$ rm
sl_data.txt paul@laika:~$ rm
hl_data.txt
```

E. Latihan

Standard file permissions

1. Sebagai pengguna biasa, buat direktori ~/permissions. Buat file milik Anda sendiri di sana.
2. Salin file yang dimiliki oleh root dari /etc/ ke direktori perizinan Anda, siapa yang memiliki file ini sekarang?
3. Sebagai root, buat file di direktori pengguna ~ / permissions.
4. Sebagai pengguna biasa, lihat siapa yang memiliki file ini yang dibuat oleh root.
5. Ubah kepemilikan semua file di ~ / izin untuk diri Anda sendiri.
6. Pastikan Anda memiliki semua hak atas file-file ini, dan yang lain hanya bisa membaca.
7. Dengan chmod, apakah 770 sama dengan rwxrwx ---?
8. Dengan chmod, apakah 664 sama dengan r-xr-xr--?
9. Dengan chmod, apakah 400 sama dengan r -----?
10. Dengan chmod, apakah 734 sama dengan rwxr-xr--?
11. Tampilkan umask dalam oktal dan dalam bentuk simbolis
12. Atur umask ke 077, tetapi gunakan format simbolik untuk mengaturnya. Verifikasi bahwa ini berfungsi.
13. Buat file sebagai root, berikan hanya baca kepada orang lain. Bisakah pengguna normal membaca file ini? Tes menulis ke file ini dengan vi.
14. Buat file sebagai pengguna biasa, berikan hanya baca kepada orang lain. Dapatkah pengguna normal lain membaca file ini? Tes menulis ke file ini dengan vi.
15. Bisakah root membaca file ini? Bisakah root menulis ke file ini dengan vi?
16. Buat direktori milik grup, di mana setiap anggota grup itu dapat membaca dan menulis ke file, dan membuat file. Pastikan bahwa orang hanya dapat menghapus file mereka sendiri

sticky, setuid and setgid bits

1. Atur direktori, yang dimiliki oleh grup olahraga.
2. Anggota grup olahraga harus dapat membuat file di direktori ini.
3. Semua file yang dibuat di direktori ini harus dimiliki grup oleh grup olahraga.
4. Pengguna harus dapat menghapus hanya file milik pengguna mereka sendiri.
5. Uji bahwa ini berhasil!
6. Verifikasi izin di /usr/bin/passwd. Hapus setuid, lalu coba ubah kata sandi Anda sebagai pengguna biasa. Setel ulang kembali izin dan coba lagi.

7. Jika waktu mengizinkan (atau jika Anda menunggu siswa lain untuk menyelesaikan praktik ini), baca tentang atribut file di halaman manual `chattr` dan `lsattr`. Coba atur atribut `i` pada file dan uji bahwa itu berfungsi.

Links

1. Buat dua file bernama `winter.txt` dan `summer.txt`, masukkan beberapa teks di dalamnya.
2. Buat hard links ke `winter.txt` bernama `hlwinter.txt`.
3. Tampilkan nomor inoda dari ketiga file ini, hard links harus memiliki inode yang sama.
4. Gunakan perintah `find` untuk mendaftarkan dua file yang terhubung dengan hard disk
5. Segala sesuatu tentang file ada di inode, kecuali dua hal: beri nama!
6. Buat tautan simbolis ke `summer.txt` yang disebut `slsummer.txt`.
7. Temukan semua file dengan nomor inode 2. Apa informasi ini memberitahu Anda?
8. Lihatlah direktori `/etc/init.d/` `/etc/rc2.d/` `/etc/rc3.d/` ... apakah Anda melihat tautannya?
9. Lihat di `/lib` dengan `ls-l` ...
10. Gunakan `find` untuk melihat di direktori home Anda untuk file biasa yang tidak (!) Memiliki satu hard links.

Bagian 5

Bekerja dengan File Contents

A. Tujuan

- Mahasiswa memahami penggunaan perintah manipulasi konten file menggunakan mode perintah

B. Teori

Dalam bab ini kita akan melihat isi file teks dengan **head**, **tail**, **cat**, **tac**, **more**, **less** dan **strings**.

Kami juga akan mendapatkan sekilas kemungkinan alat-alat seperti **cat** pada baris perintah.

- **cat**

Perintah **cat** adalah salah satu alat yang paling universal, namun yang dilakukannya hanyalah menyalin **standard input** ke **standard output**. Dalam kombinasi dengan cangkang ini bisa sangat kuat dan beragam. Beberapa contoh akan memberi sekilas kemungkinan. Contoh pertama adalah sederhana, Anda dapat menggunakan **cat** untuk menampilkan file di layar. Jika file lebih panjang dari layar, itu akan bergulir ke ujung.

- custom end marker

Anda dapat memilih penanda akhir untuk **cat** dengan **<<** seperti yang ditunjukkan pada tangkapan layar ini. Konstruksi ini disebut **here directive** dan akan mengakhiri perintah **cat**.

- more and less

Perintah **more** berguna untuk menampilkan file yang mengambil lebih dari satu layar. **more** akan memungkinkan Anda untuk melihat isi dari file halaman demi halaman. Gunakan bilah spasi untuk melihat halaman berikutnya, atau **q** untuk keluar. Sebagian orang lebih menyukai perintah **less** untuk **more**.

- **strings**

Dengan perintah **string** Anda dapat menampilkan string ascii yang dapat dibaca yang ditemukan dalam file (binary). Contoh ini menempatkan biner **ls** kemudian menampilkan string yang dapat dibaca dalam file biner (output terpotong).

C. Alat dan Bahan

- Komputer
- Virtual Box dengan Instalasi Linux OS

D. Praktik (Minggu 8)

- **head**

Anda dapat menggunakan **head** untuk menampilkan sepuluh baris pertama file.

```
paul@debian7~$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
```

```
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/
sh root@debian7~#
```

Perintah **head** juga dapat menampilkan baris n pertama file.

```
paul@debian7~$ head -4 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/
sh bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
paul@debian7~$
```

Dan **head** juga bisa menampilkan n byte pertama.

```
paul@debian7~$ head -c14 /etc/passwd
root:x:0:0:roopaul@debian7~$
```

- **Tail**

Mirip dengan **head**, perintah **tail** akan menampilkan sepuluh baris terakhir dari sebuah file.

```
paul@debian7~$ tail /etc/services
vboxd                20012/udp
binkp                 24554/tcp           # binkp fidonet
protocol asp         27374/tcp           # Address
Search Protocol asp  27374/udp
csync2               30865/tcp           # cluster synchronization
tool dircproxy       57000/tcp           # Detachable IRC Proxy
tfido                60177/tcp           # fidonet EMSI over
telnet fido          60179/tcp           # fidonet EMSI
over TCP

# Local
services
paul@debian7~$
```

Anda dapat memberi **tail** jumlah garis yang ingin Anda lihat.

```
paul@debian7~$ tail -3 /etc/services
fido                60179/tcp           # fidonet EMSI over TCP

# Local
services
paul@debian7~$
```

Perintah **tail** memiliki opsi berguna lainnya, beberapa di antaranya akan kita gunakan.

- **cat**

```
paul@debian8:~$ cat
/etc/resolv.conf
domain                linux-
training.be          search
linux-training.be
nameserver
192.168.1.42
```

- **concatenate**

Cat adalah kependekan dari **concatenate**. Salah satu kegunaan dasar cat adalah menggabungkan file menjadi lebih besar (atau lengkap) file.


```

paul@debian8:~$ echo one >part1
paul@debian8:~$ echo two >part2
paul@debian8:~$ echo three
>part3 paul@debian8:~$ cat
part1
one
paul@debian8:~$ cat part2
two
paul@debian8:~$ cat part3
three
paul@debian8:~$ cat part1 part2 part3
one two three
paul@debian8:~$ cat part1 part2 part3 >all
paul@debian8:~$ cat all
one two three
paul@debian8:~$

```

- create files

Anda dapat menggunakan **cat** untuk membuat file teks datar. Ketik perintah **cat> winter.txt** seperti yang ditunjukkan pada gambar di bawah. Kemudian ketik satu atau lebih baris, finishing setiap baris dengan tombol enter. Setelah baris terakhir, ketik dan tahan tombol Control (Ctrl) dan tekan d.

```

paul@debian8:~$ cat >
winter.txt It is very cold
today! paul@debian8:~$ cat
winter.txt
It is very cold today!
paul@debian8:~$

```

Kombinasi tombol Ctrl d akan mengirim EOF (End of File) ke proses yang sedang berjalan yang mengakhiri perintah cat.

- custom end marker

```

paul@debian8:~$ cat > hot.txt <<stop
> It is hot today!
> Yes it is summer.
> stop
paul@debian8:~$ cat hot.txt
It is hot
today! Yes it
is summer.
paul@debian8:~$

```

- copy files

Pada contoh ketiga Anda akan melihat bahwa cat dapat digunakan untuk menyalin file. Kami akan menjelaskan secara detail apa yang terjadi di sini di bab shell bash.

```

paul@debian8:~$ cat winter.txt
It is very cold today!
paul@debian8:~$ cat winter.txt > cold.txt
paul@debian8:~$ cat
cold.txt It is very cold
today! paul@debian8:~$

```

- tac

Hanya satu contoh yang akan menunjukkan kepada Anda tujuan tac (cat backwards).

```

paul@debian8:~$ cat count
one two three four
paul@debian8:~$ tac count

```

four three two one

- **strings**

```
paul@laika:~$ which ls
/bin/ls
paul@laika:~$ strings /bin/ls
/lib/ld-
linux.so.2
librt.so.1
    gmon_start__
_Jv_RegisterClass
es clock_gettime
libacl.so.1
...
```

Exercise

1. Tampilkan 12 baris pertama dari / etc / services.
2. Tampilkan baris terakhir dari / etc / passwd.
3. Gunakan cat untuk membuat file bernama count.txt yang terlihat seperti ini:

One Two Three Four Five

4. Gunakan cp untuk membuat cadangan file ini ke cnt.txt.
5. Gunakan cat untuk membuat cadangan file ini ke catcnt.txt.
6. Tampilkan catcnt.txt, tetapi dengan semua baris dalam urutan terbalik (baris terakhir lebih dulu).
7. Gunakan lebih banyak untuk menampilkan / etc / services.
8. Tampilkan string karakter yang dapat dibaca dari perintah / usr / bin / passwd.
9. Gunakan ls untuk mencari file terbesar di / etc.
10. Buka dua terminal windows (atau tab) dan pastikan Anda berada di direktori yang sama di keduanya. Ketik gema ini adalah baris pertama> tailing.txt di terminal pertama, lalu keluarkan tail-tailing.txt di terminal kedua. Sekarang kembali ke terminal pertama dan ketik echo Ini adalah baris lain >> tailing.txt (perhatikan tanda double >>), verifikasi bahwa tail -f di terminal kedua menunjukkan kedua garis. Hentikan ekor -f dengan Ctrl-C.
11. Gunakan cat untuk membuat file bernama tailing.txt yang berisi isi tailing.txt diikuti oleh isi / etc / passwd.
12. Gunakan cat untuk membuat file bernama tailing.txt yang berisi isi tailing.txt

Bagian 6

Hierarki Filesystem

A. Tujuan

- Siswa memahami fungsi berbagai direktori pada sistem linux

B. Teori

Standart Hierarki Filesystem

Banyak distribusi Linux sebagian mengikuti **Filesystem Hierarchy Standard**. FHS dapat membantu membuat lebih banyak pohon sistem file Unix / Linux lebih baik di masa depan. FHS tersedia online di <http://www.pathname.com/fhs/> di mana kita membaca: "Standar hirarki sistem file telah dirancang untuk digunakan oleh pengembang distribusi Unix, pengembang paket, dan pelaksana sistem. Namun, itu terutama dimaksudkan sebagai referensi dan bukan tutorial tentang cara mengelola filesystem Unix atau hirarki direktori. "

Konfigurasi direktori

/boot

Direktori / **boot** berisi semua file yang diperlukan untuk mem-boot komputer. File-file ini tidak sering berubah. Pada sistem Linux Anda biasanya menemukan direktori / boot / grub di sini. / **boot** / **grub** berisi **/boot/grub/grub.cfg** (sistem yang lebih lama mungkin masih memiliki **/boot/grub/grub.conf**) yang mendefinisikan menu boot yang ditampilkan sebelum kernel dimulai.

/etc

Semua file konfigurasi mesin-spesifik harus ditempatkan di / etc. Secara historis / etc berdiri untuk sebagainya, hari ini orang sering menggunakan **Editable Text Configuration** backronym.

/etc/init.d/

Banyak distribusi Unix / Linux memiliki direktori /etc/init.d yang berisi skrip untuk memulai dan menghentikan daemon. Direktori ini dapat menghilang ketika Linux bermigrasi ke sistem yang menggantikan cara init lama untuk memulai semua daemon.

/etc/X11/

Tampilan grafis (alias X Window System atau just **X**) digerakkan oleh perangkat lunak dari X.org foundation. File konfigurasi untuk tampilan grafis Anda adalah **/etc/X11/xorg.conf**.

/etc/skel/

Direktori **skeleton** / etc / skel disalin ke direktori home dari pengguna yang baru dibuat. Biasanya berisi file yang tersembunyi seperti skrip .bashrc.

/etc/sysconfig/

Direktori ini, yang tidak disebutkan dalam FHS, berisi banyak file dari **Red Hat Enterprise Linux**. Kami akan membahas beberapa di antaranya secara lebih rinci. Screenshot di bawah adalah direktori **/ etc / sysconfig** dari RHELv4u4 dengan semua yang terinstal

/root

Pada banyak sistem / root adalah lokasi default untuk data pribadi dan profil pengguna root. Jika tidak ada secara default, maka beberapa administrator membuatnya

/srv

You may use **/srv** for data that is **served by your system**. The FHS allows locating cvs, rsync, ftp and www data in this location. The FHS also approves administrative naming Anda dapat menggunakan / srv untuk data yang dilayani oleh sistem Anda. FHS memungkinkan menemukan data cvs, rsync, ftp dan www di lokasi ini. FHS juga menyetujui penamaan administratif pada / srv, seperti /srv/project55/ftp dan /srv/sales/www.

Pada Sun Solaris (atau Oracle Solaris) / digunakan untuk tujuan ini.

/sys Linux 2.6 hot plugging

Direktori / sys dibuat untuk kernel Linux 2.6. Sejak 2.6, Linux menggunakan sysfs untuk mendukung perangkat hot plug USB dan IEEE 1394 (FireWire). Lihat halaman manual udev (8) (penerus devfs) dan hotplug (8) untuk info lebih lanjut (atau kunjungi <http://linux-hotplug.sourceforge.net/>).

Pada dasarnya direktori / sys berisi informasi kernel tentang perangkat keras.

C. Alat dan Bahan

- Komputer
- Virtual Box dengan Instalasi Linux OS

D. Praktik (minggu 9)

- **man hier**

Ada beberapa perbedaan dalam sistem file antara **distribusi Linux**. Untuk bantuan tentang mesin Anda, enter **man hier** manusia untuk menemukan informasi tentang hirarki sistem file. Panduan ini akan menjelaskan struktur direktori di komputer Anda.

- **root directory (/)**

Semua sistem Linux memiliki struktur direktori yang dimulai pada direktori root. Direktori root diwakili oleh garis miring ke depan / **forward slash**, seperti ini: /. Segala sesuatu yang ada di sistem Linux Anda dapat ditemukan di bawah direktori root ini. Mari kita lihat sekilas isi direktori root.

```
[paul@RHELv4u3 ~]$ ls /  
bin  dev  home  media  mnt  proc  sbin      srv  tftpboot  
usr  boot  etc   lib    misc  opt   root      selinux  sys  tmp  
var
```

- **binary directories (/bin)**

Binari adalah file yang berisi kode sumber yang dikompilasi (atau kode mesin). Binar bisa

dieksekusi di komputer. Terkadang binari disebut **executable**

Direktori / bin berisi binari untuk digunakan oleh semua pengguna. Menurut FHS si / bin

direktori harus berisi / bin / cat dan / bin / date (antara lain).

Pada screenshot di bawah ini Anda melihat perintah umum Unix / Linux seperti cat, cp, cpio, tanggal, dd, echo, grep, dan seterusnya. Banyak dari ini akan dibahas dalam buku ini.

```
paul@laika:~$ ls /bin
archdetect      egrep           mt
setupcon        autopartition   false          mt-gnu
sh
bash            fgconsole      mv
sh.distrib      bunzip2        fgrep          nano
sleep
bzcat           fuser          nc
stralign        bzcmp          fusermount     nc.traditional
stty            bzdiff         get_mouptions  netcat
su              bzegrep        grep           netstat
sync            bzexe          gunzip         ntfs-3g
sysfs           bzfgrep        gzexe          ntfs-3g.probe
tailf           bzgrep         gzip           parted_devices
tar
bzip2           hostname       parted_server
tempfile        bzip2recover   hw-detect      partman
touch           bzless         ip             partman-commit
true            bzmre          kbd_mode       perform_recipe
unlockmgr       cat            kill           pidof
umount
...
```

- **other /bin directories**

Anda dapat menemukan subdirektori /bin di banyak direktori lainnya. Seorang pengguna bernama serena dapat menempatkan programnya sendiri di / **home / serena / bin**.

Beberapa aplikasi, seringkali ketika dipasang langsung dari sumber akan menempatkan diri di / opt. Instalasi **server samba** dapat menggunakan / **opt / samba / bin** untuk menyimpan binari-nya.

- **system binaries**

/ sbin berisi binari untuk mengkonfigurasi sistem operasi. Banyak sistem binari membutuhkan hak istimewa root untuk melakukan tugas-tugas tertentu.

Di bawah screenshot berisi binari sistem untuk mengubah alamat ip, mempartisi disk dan membuat sistem file ext4.

```
paul@ubu1010:~$ ls -l /sbin/ifconfig /sbin/fdisk /sbin/mkfs.ext4
-rwxr-xr-x 1 root root 97172 2011-02-02 09:56 /sbin/fdisk
-rwxr-xr-x 1 root root 65708 2010-07-02 09:27 /sbin/ifconfig
-rwxr-xr-x 5 root root 55140 2010-08-18 18:01 /sbin/mkfs.ext4
```

- **libraries & shared libraries**

Binari yang ditemukan di / **bin** dan / **sbin** sering menggunakan pustaka bersama yang terletak di / **lib**. Di bawah ini adalah screenshot dari sebagian konten / lib.

```
paul@laika:~$ ls
/lib/libc*
/lib/libc-2.5.so /lib/libcfont.so.0.0.0
/lib/libcom_err.so.2.1
/lib/libcap.so.1 /lib/libcidn-2.5.so
/lib/libconsole.so.0
/lib/libcap.so.1.10 /lib/libcidn.so.1
/lib/libconsole.so.0.0.0
/lib/libcfont.so.0 /lib/libcom_err.so.2 /lib/libcrypt-
2.5.so
```

Biasanya, **kernel Linux** memuat modul kernel dari **/lib/modules/\$kernel-version/**. Saat ini kami berada dalam transisi antara sistem 32-bit dan 64-bit. Oleh karena itu, Anda mungkin menemukan direktori bernama **/lib32** dan **/lib64** yang memperjelas ukuran register yang digunakan selama waktu penyusunan pustaka. Komputer 64-bit mungkin memiliki beberapa binari 32-bit dan pustaka untuk kompatibilitas dengan aplikasi warisan. Screenshot ini menggunakan utilitas file untuk menunjukkan perbedaannya.

```
paul@laika:~$ file /lib32/libc-2.5.so
/lib32/libc-2.5.so: ELF 32-bit LSB shared object, Intel 80386, \
version 1 (SYSV), for GNU/Linux 2.6.0, \
stripped
paul@laika:~$ file /lib64/libcap.so.1.10
/lib64/libcap.so.1.10: ELF 64-bit LSB shared object, AMD x86-64, \
version 1 (SYSV), stripped
```

ELF (Executable and Linkable Format) digunakan di hampir setiap sistem operasi mirip Unix sejak System V.

- **optional**

Tujuan dari **/opt** adalah untuk menyimpan perangkat lunak opsional. Dalam banyak kasus, ini adalah perangkat lunak dari luar repositori distribusi. Anda mungkin menemukan direktori kosong **/opt** pada banyak sistem.

Paket besar dapat menginstal semua file di **/bin**, **/lib**, **/etc** subdirektori dalam **/opt/\$packagename/**. Jika misalnya paket disebut **wp**, maka menginstal di **/opt/wp**, meletakkan binari di **/opt/wp/bin** dan manpages di **/opt/wp/man**.

- **Etcetera (/etc) Editable Text Configuration**

Banyak kali nama file konfigurasi sama dengan aplikasi, daemon, atau protokol dengan **.conf** ditambahkan sebagai ekstensi.

```
paul@laika:~$ ls /etc/*.conf
```

/etc/adduser.conf	/etc/ld.so.conf	/etc/scrollkeeper.conf
/etc/brltty.conf	/etc/lftp.conf	/etc/sysctl.conf
/etc/ccertificates.conf	/etc/libao.conf	/etc/syslog.conf
/etc/cvs-cron.conf	/etc/logrotate.conf	/etc/ucf.conf
/etc/ddclient.conf	/etc/ltrace.conf	/etc/uniconf.conf
/etc/debconf.conf	/etc/mke2fs.conf	/etc/updatedb.conf
/etc/deluser.conf	/etc/netscsid.conf	/etc/usplash.conf
/etc/fdmount.conf	/etc/nsswitch.conf	/etc/uswsusp.conf
/etc/hdparm.conf	/etc/pam.conf	/etc/vnc.conf
/etc/host.conf	/etc/pnm2ppa.conf	/etc/wodim.conf

```

/etc/inetd.conf      /etc/povray.conf    /etc/wvdial.conf
/etc/kernel-img.conf /etc/resolv.conf
paul@laika:~$

```

```

paul@RHELv4u4:~$ ls /etc/sysconfig/
apmd          firstboot      irda           network        saslauthd
apm-scripts   grub           irqbalance     networking     selinux
authconfig    hidd           keyboard       ntpd
spamassassin  autofs        httpd          kudzu
openib.conf   squid          bluetooth      hwconf         lm_sensors    pand
syslog
clock          il8n           mouse          pcmcia         sys-config-sec
console        init           mouse.B        pgsql          sys-config-
users crond      installinfo    named          prelink        sys-
logviewer      desktop        ipmi           netdump        rawdevices    tux
diskdump        iptables       netdump_id_dsa rhn
vncservers      dund           iptables-cfg   netdump_id_dsa.p samba
xinetd
paul@RHELv4u4:~$

```

File / etc / sysconfig / firstboot memberitahu Red Hat Setup Agent untuk tidak dijalankan pada saat boot. Jika Anda ingin menjalankan Agen Pengaturan Red Hat di reboot berikutnya, maka cukup hapus file ini, dan jalankan chkconfig - tingkat 5 firstboot aktif. Red Hat Setup Agent memungkinkan Anda menginstal pembaruan terbaru, membuat akun pengguna, bergabung dengan Red Hat Network dan banyak lagi. Kemudian akan membuat file / etc / sysconfig / firstboot lagi.

```

paul@RHELv4u4:~$ cat /etc/sysconfig/firstboot
RUN_FIRSTBOOT=
NO

```

File / etc / sysconfig / hddisks berisi beberapa parameter untuk menyesuaikan hard disk. File itu menjelaskan dirinya sendiri.

Anda dapat melihat perangkat keras yang terdeteksi oleh kudzu di / etc / sysconfig / hwconf. Kudzu adalah perangkat lunak dari Red Hat untuk penemuan otomatis dan konfigurasi perangkat keras.

Jenis keyboard dan tabel keymap diatur dalam file / etc / sysconfig / keyboard. Untuk informasi keyboard konsol lebih lanjut, periksa halaman buku panduan keymap (5), dumpkeys (1), loadkeys (1) dan direktori / lib / kbd / keymaps /.

```

root@RHELv4u4:/etc/sysconfig# cat keyboard
KEYBOARDTYPE="p
c"
KEYTABLE="us"

```

- home directory

Pengguna dapat menyimpan data pribadi atau proyek di bawah / di rumah. Ini adalah praktik umum (tetapi tidak wajib oleh fhs) untuk memberi nama direktori home pengguna setelah nama pengguna dalam format / home /

\$ USERNAME. Sebagai contoh:

```

paul@ubu606:~$ ls /home

```

```
geert   annik   sandra   paul  
tom
```

Selain memberikan setiap pengguna (atau setiap proyek atau grup) lokasi untuk menyimpan file pribadi, direktori home pengguna juga berfungsi sebagai lokasi untuk menyimpan profil pengguna. Profil pengguna Unix yang khas berisi banyak file tersembunyi (file yang nama filenya dimulai dengan sebuah titik). File-file yang tersembunyi dari profil pengguna Unix berisi pengaturan khusus untuk pengguna itu.

```
paul@ubu606:~$ ls -d /home/paul/.*  
/home/paul/.                               /home/paul/.bash_profile  
/home/paul/.ssh                             /home/paul/.bashrc  
/home/paul/.  
/home/paul/.viminfo  
/home/paul/.bash_history  
/home/paul/.lessht
```

- **media**

Direktori / **media** berfungsi sebagai titik mount untuk **removable media devices** seperti CD-ROM, kamera digital, dan berbagai perangkat yang dipasang dengan usb. Karena / media agak baru di dunia Unix, Anda bisa sangat menghadapi sistem yang berjalan tanpa direktori ini. Solaris 9 tidak memilikinya, Solaris 10 tidak. Kebanyakan distribusi Linux saat ini me-mount semua media yang dapat dilepas di / media.

```
paul@debian5:~$ ls /media/  
cdrom  cdrom0  usbdisk
```

- **mount /mnt**

Direktori / **mnt** harus kosong dan seharusnya hanya digunakan untuk titik mount sementara (sesuai dengan FHS).

Administrator Unix dan Linux digunakan untuk membuat banyak direktori di sini, seperti / mnt / something /. Anda mungkin akan menemukan banyak sistem dengan lebih dari satu direktori dibuat dan / atau dipasang di dalam / mnt untuk digunakan untuk berbagai filesystem lokal dan remote.

- **Temp /tmp**

Aplikasi dan pengguna harus menggunakan / tmp untuk menyimpan data sementara saat diperlukan. Data yang disimpan di / tmp dapat menggunakan ruang disk atau RAM. Keduanya dikelola oleh sistem operasi. Jangan pernah gunakan / tmp untuk menyimpan data yang penting atau yang ingin Anda arsipkan

memory directories

- **Device /dev**

File perangkat di / dev tampak seperti file biasa, tetapi sebenarnya tidak ada di hard disk. Direktori / dev diisi dengan file karena kernel mengenali perangkat keras.

- **common physical devices**

Perangkat keras umum seperti perangkat hard disk diwakili oleh file perangkat di / dev. Di bawah screenshot file perangkat SATA pada laptop dan kemudian drive IDE yang terpasang di desktop

```
#
# SATA or SCSI or
USB
#
paul@laika:~$ ls
/dev/sd*
/dev/sda   /dev/sda1   /dev/sda2   /dev/sda3   /dev/sdb   /dev/sdb1
/dev/sdb2

#
# IDE or
ATAPI
#
paul@barry:~$ ls
/dev/hd*
/dev/hda   /dev/hda1   /dev/hda2   /dev/hdb    /dev/hdb1   /dev/hdb2
/dev/hdc
```

Selain merepresentasikan perangkat fisik, beberapa file perangkat bersifat khusus. Perangkat khusus ini bisa sangat berguna.

/dev/tty and /dev/pts

Misalnya, / dev / **tty1** mewakili terminal atau konsol yang melekat pada sistem. (Jangan merusak kepala Anda pada terminologi 'terminal' atau 'konsol' yang tepat, yang kami maksudkan di sini adalah antarmuka baris perintah.) Saat mengetik perintah di terminal yang merupakan bagian dari antarmuka grafis seperti Gnome atau KDE, maka terminal Anda akan direpresentasikan sebagai / dev / **pts / 1** (1 dapat berupa angka lain).

/dev/null

Di Linux Anda akan menemukan perangkat khusus lainnya seperti / dev / null yang dapat dianggap sebagai lubang hitam; ini memiliki penyimpanan tidak terbatas, tetapi tidak ada yang dapat diambil dari itu. Secara teknis, apa pun yang ditulis ke / dev / null akan dibuang. / dev / null dapat berguna untuk membuang keluaran yang tidak diinginkan dari perintah. / dev / null bukan lokasi yang baik untuk menyimpan backup Anda ;-).

/proc conversation with the kernel

/ proc adalah direktori khusus lain, muncul sebagai file biasa, tetapi tidak menggunakan ruang disk. Ini sebenarnya adalah tampilan kernel, atau lebih baik, apa yang kernel kelola, dan merupakan sarana untuk berinteraksi dengannya secara langsung. / proc adalah filesystem proc.

```
paul@RHELv4u4:~$ mount -t proc
none on /proc type proc (rw)
```

Ketika daftar direktori / proc Anda akan melihat banyak angka (pada Unix apa saja) dan beberapa file menarik (di Linux)

```
mul@laika:~$ ls /proc
```

1	2339	4724	5418	6587	7201	cmdline	mounts
10175	2523	4729	5421	6596	7204	cpuinfo	mtrr
10211	2783	4741	5658	6599	7206	crypto	net
10239	2975	4873	5661	6638	7214	devices	pagetypeinfo
141	29775	4874	5665	6652	7216	diskstats	partitions
15045	29792	4878	5927	6719	7218	dma	sched_debug
1519	2997	4879	6	6736	7223	driver	scsi
1548	3	4881	6032	6737	7224	execdomains	self
1551	30228	4882	6033	6755	7227	fb	slabinfo
1554	3069	5	6145	6762	7260	filesystems	stat
1557	31422	5073	6298	6774	7267	fs	swaps
1606	3149	5147	6414	6816	7275	ide	sys
180	31507	5203	6418	6991	7282	interrupts	sysrq-trigger
181	3189	5206	6419	6993	7298	iomem	sysvipc
182	3193	5228	6420	6996	7319	ioports	timer_list
18898	3246	5272	6421	7157	7330	irq	timer_stats
19799	3248	5291	6422	7163	7345	kallsyms	tty
19803	3253	5294	6423	7164	7513	kcore	uptime
19804	3372	5356	6424	7171	7525	key-users	version
1987	4	5370	6425	7175	7529	kmsg	version_signature
1989	42	5379	6426	7188	9964	loadavg	vmcore
2	45	5380	6430	7189	acpi	locks	vmnet
20845	4542	5412	6450	7191	asound	meminfo	vmstat
221	46	5414	6551	7192	buddyinfo	misc	zoneinfo
2338	4704	5416	6568	7199	bus	modules	

Mari kita selidiki properti file di dalam / proc. Melihat tanggal dan waktu akan menampilkan tanggal dan waktu saat ini yang menunjukkan file terus diperbarui (tampilan pada kernel).

```
paul@RHELv4u4:~$ date
Mon Jan 29 18:06:32 EST 2007
paul@RHELv4u4:~$ ls -al
/proc/cpuinfo
-r--r--r-- 1 root root 0 Jan 29 18:06
/proc/cpuinfo paul@RHELv4u4:~$
paul@RHELv4u4:~$ ..time
passes... paul@RHELv4u4:~$
paul@RHELv4u4:~$ date
Mon Jan 29 18:10:00 EST 2007
paul@RHELv4u4:~$ ls -al
/proc/cpuinfo
-r--r--r-- 1 root root 0 Jan 29 18:10 /proc/cpuinfo
```

Sebagian besar file dalam / proc adalah 0 byte, namun mereka berisi data - terkadang banyak data. Anda dapat melihat ini dengan mengeksekusi cat pada file seperti / proc / cpuinfo, yang berisi informasi tentang CPU.

```
paul@RHELv4u4:~$ file /proc/cpuinfo
/proc/cpuinfo: empty
paul@RHELv4u4:~$ cat
/proc/cpuinfo processor      :
0
vendor_id      : AuthenticAMD
cpu family     :
15 model
: 43

model name     : AMD Athlon(tm) 64 X2 Dual Core Processor 4600+
```

```

stepping      : 1
cpu MHz       :
2398.628 cache size
: 512 KB fdiv_bug
: no hlt_bug   :
no f00f_bug   : no
coma_bug      : no
fpu           :
yes    fpu_exception
: yes  cpuid level
: 1
wp       : yes
flags    : fpu vme de pse tsc msr pae mce cx8 apic mtrr
pge... bogomips      : 4803.54

```

Just for fun, here is /proc/cpuinfo on a Sun Sunblade 1000...

```

paul@pasha:~$ cat /proc/cpuinfo
cpu : TI UltraSparc III (Cheetah)
fpu : UltraSparc III integrated
FPU promlib : Version 3 Revision
2
prom : 4.2.2
type : sun4u
ncpus probed :
2 ncpus active
: 2
Cpu0Bogo : 498.68
Cpu0ClkTck : 000000002cb41780
Cpu1Bogo : 498.68
Cpu1ClkTck : 000000002cb41780
MMU Type : Cheetah
State:
CPU0: online
CPU1: online

```

Sebagian besar file di / proc hanya dapat dibaca, beberapa memerlukan hak akses root, beberapa file dapat ditulis, dan banyak file di / proc / sys dapat ditulis. Mari kita bahas beberapa file di / proc.

/proc/interrupts

Pada arsitektur x86, / proc / interrupts menampilkan interupsi.

```
paul@RHELv4u4:~$ cat /proc/interrupts
```

CPU0

```

0:   13876877   IO-APIC-edge timer
1:       15    IO-APIC-edge i8042
8:       1     IO-APIC-edge rtc
9:       0     IO-APIC-level acpi
12:      67     IO-APIC-edge i8042
14:     128     IO-APIC-edge ide0
15:    124320     IO-APIC-edge ide1
169:   111993   IO-APIC-level ioc0
177:    2428   IO-APIC-level eth0
NMI:          0
LOC:   13878037
ERR:          0
MIS:          0

```

Pada mesin dengan dua CPU, file terlihat seperti ini.

```

paul@laika:~$ cat
/proc/interrupts

```

CPU0
CPU1

```
0:      860013      0 IO-APIC-edge timer
1:      4533      0 IO-APIC-edge i8042
7:        0      0 IO-APIC-edge parport0
8:    6588227      0 IO-APIC-edge rtc
10:     2314      0 IO-APIC-fasteoi acpi
12:      133      0 IO-APIC-edge i8042
14:        0      0 IO-APIC-edge libata
15:    72269      0 IO-APIC-edge libata
18:        1      0 IO-APIC-fasteoi yenta
19:    115036      0 IO-APIC-fasteoi eth0
20:    126871      0 IO-APIC-fasteoi libata, ohci1394
21:    30204      0 IO-APIC-fasteoi ehci_hcd:usb1, uhci_hcd:usb2
22:     1334      0 IO-APIC-fasteoi saa7133[0], saa7133[0]
24:    234739      0 IO-APIC-fasteoi nvidia
NMI:        72      42
LOC:    860000    859994
ERR:        0
```

/proc/kcore

Memori fisik diwakili dalam / **proc** / **kcore**. Jangan mencoba untuk cat file ini, bukan menggunakan debugger. Ukuran / proc / kcore sama dengan memori fisik Anda, ditambah empat byte.

```
paul@laika:~$ ls -lh /proc/kcore
-r----- 1 root root 2.0G 2007-01-30 08:57
/proc/kcore paul@laika:~$
```

- **Unix System Resources**

/usr

Meskipun / usr diucapkan seperti pengguna, ingat bahwa itu singkatan dari Sumber Daya Sistem Unix. Hierarki / usr harus berisi **shareable**, **read only** data. Sebagian orang memilih untuk melakukan mount

/ usr sebagai hanya baca. Ini dapat dilakukan dari partisi sendiri atau dari hanya berbagi NFS(NFS dibahas nanti).

/usr/bin

The **/usr/bin** directory berisi banyak perintah-perintah.

```
paul@deb508:~$ ls /usr/bin | wc -l
1395
```

(On Solaris the **/bin** directory adalah sebuah symbolic link pada **/usr/bin**.)

/usr/include

Direktori / usr / include berisi penggunaan umum termasuk file untuk C.

```
paul@ubu1010:~$ ls /usr/include/
```

aalib.h	expat_config.h	math.h	search.h
af_vfs.h	expat_external.h	mcheck.h	semaphore.h
aio.h	expat.h	memory.h	setjmp.h
AL	fcntl.h	menu.h	sgtty.h
aliases.h	features.h	mntent.h	shadow.h
...			

/usr/lib

The **/usr/lib** direktori berisi librari yang tidak dapat dieksekusi secara langsung oleh pengguna atau scripts.

```
paul@deb508:~$ ls /usr/lib | head -7
4Sui
te
ao
apt
arj
aspe
ll
avah
i
bono
bo
```

/usr/local

Direktori **/usr/local** dapat digunakan oleh administrator untuk menginstal perangkat lunak secara lokal.

```
paul@deb508:~$ ls /usr/local/
bin etc games include lib man sbin share
src paul@deb508:~$ du -sh /usr/local/
128K /usr/local/
```

/usr/share

Direktori **/usr/share** berisi data arsitektur independen. Seperti yang Anda lihat, ini adalah direktori yang cukup besar

```
paul@deb508:~$ ls /usr/share/ | wc -l
263
paul@deb508:~$ du -sh /usr/share/
1.3G /usr/share/
```

Direktori ini biasanya berisi **/usr/share/man** untuk manual pages

```
paul@deb508:~$ ls /usr/share/man
cs fr hu it.UTF-8 man2 man6 pl.ISO8859-2 sv
de fr.ISO8859-1 id ja man3 man7 pl.UTF-8
tr es fr.UTF-8 it ko man4 man8 pt_BR
zh_CN
fi gl it.ISO8859-1 man1 man5 pl ru zh_TW
```

Dan ini berisi **/usr/share/games** untuk semua data permainan statis (jadi tidak ada skor tinggi atau memutar log).

```
paul@ubu1010:~$ ls /usr/share/games/
openttd wesnoth
```

/usr/src

Direktori **/usr/src** adalah lokasi yang direkomendasikan untuk file sumber kernel.

```
paul@deb508:~$ ls -l /usr/src/
total 12
drwxr-xr-x  4 root root 4096 2011-02-01 14:43 linux-headers-2.6.26-2-686
drwxr-xr-x 18 root root 4096 2011-02-01 14:43 linux-headers-2.6.26-2-
common drwxr-xr-x  3 root root 4096 2009-10-28 16:01 linux-kbuild-2.6.26
```

- **variable data**

/var

File yang tidak dapat diprediksi dalam ukuran, seperti file log, cache, dan spool, harus ditempatkan di / var.

/var/log

Direktori / var / log berfungsi sebagai titik sentral untuk memuat semua file log.

```
[paul@RHEL4b ~]$ ls /var/log
acpid          cron.2      maillog.2    quagga
secure.4       amanda      cron.3       maillog.3    radius
spooler.1      anaconda.log cron.4       maillog.4    rpmpkgs
spooler.1      anaconda.syslog cups         mailman      rpmpkgs.1
spooler.2      anaconda.xlog dmesg        messages     rpmpkgs.2
spooler.3      audit       exim         messages.1   rpmpkgs.3
spooler.4      boot.log    gdm          messages.2   rpmpkgs.4
squid boot.log.1      httpd        messages.3   sa
uucp boot.log.2      iiim         messages.4   samba
vbox
boot.log.3      iptraf      mysqld.log   scrollkeeper.log vmware-tools-
guestd boot.log.4      lastlog     news         secure         wtmp
canna          mail        pgsql       secure.1       wtmp.1
cron           maillog     ppp         secure.2       Xorg.0.log
cron.1         maillog.1   prelink.log secure.3
Xorg.0.log.old
```

/var/log/messages

Sebuah file pertama yang tipikal untuk diperiksa ketika troubleshooting pada Red Hat (dan turunannya) adalah file / var / log / messages. Secara default file ini akan berisi informasi tentang apa yang baru saja terjadi pada sistem. File tersebut bernama / var / log / syslog pada Debian dan Ubuntu.

```
[root@RHEL4b ~]# tail /var/log/messages
Jul 30 05:13:56 anacron: anacron startup succeeded
Jul 30 05:13:56 atd: atd startup succeeded
Jul 30 05:13:57 messagebus: messagebus startup succeeded
Jul 30 05:13:57 cups-config-daemon: cups-config-daemon startup succeeded
Jul 30 05:13:58 haldaemon: haldaemon startup succeeded
Jul 30 05:14:00 fstab-sync[3560]: removed all generated mount points
Jul 30 05:14:01 fstab-sync[3628]: added mount point /media/cdrom
for... Jul 30 05:14:01 fstab-sync[3646]: added mount point
/media/floppy for... Jul 30 05:16:46 sshd(pam_unix)[3662]: session
opened for user paul by... Jul 30 06:06:37 su(pam_unix)[3904]: session
opened for user root by paul
```

/var/cache

Direktori / var / cache dapat berisi data cache untuk beberapa aplikasi.

```
paul@ubul010:~$ ls /var/cache/
```

apt	dictionaries-common	gdm	man	software-center
binfmts	flashplugin-installer	hald	pm-utils	
cups	fontconfig	jockey	pppconfig	
debconf	fonts	ldconfig	samba	

/var/spool

Direktori / var / spool biasanya berisi direktori spool untuk mail dan cron, tetapi juga berfungsi sebagai direktori induk untuk file spool lainnya (misalnya file spool cetak)

/ var / lib

Direktori / var / lib berisi informasi status aplikasi.

Red Hat Enterprise Linux misalnya menyimpan file yang berkaitan dengan rpm di / var / lib / rpm /.

/ var / ...

/ var juga berisi file Process ID di / var / run (segera diganti dengan / dijalankan) dan file sementara yang bertahan dengan reboot di / var / tmp dan informasi tentang kunci file di / var / lock.

E. Latihan

1. Apakah file / **bin** / **cat** ada? Bagaimana dengan / **bin** / **dd** dan / **bin** / **echo**. Apa jenis file-file ini?
2. Berapa ukuran file kernel Linux (vmlinu *) di / **boot**?
3. Buat direktori ~ / **test**. Kemudian jalankan perintah berikut:

```
cd ~/test
```

```
dd if=/dev/zero of=zeros.txt count=1 bs=100
```

```
od zeroes.txt
```

dd akan di copy sekali (count=1) ukuran blok 100 bytes (bs=100) dari file /**dev/zero** sampai ~/test/zeros.txt. dapat kah anda mendeskripsikan fungsi dari /**dev/zero** ?

4. Sekarang jalankan perintah berikut:

```
dd if=/dev/random of=random.txt count=1 bs=100 ; od random.txt
```

dd akan di copy sekali (count=1) ukuran blok 100 bytes (bs=100) dari file /**dev/random** sampai ~/test/random.txt. dapat kah anda mendeskripsikan fungsi dari /**dev/random** ?

5. Terbitkan dua perintah berikut, dan lihat karakter pertama dari setiap baris output

```
ls -l /dev/sd* /dev/hd*
```

```
ls -l /dev/tty*  
/dev/input/mou*
```

Ls pertama akan menampilkan blok (b) perangkat, ls kedua menunjukkan karakter (c) perangkat. Bisakah Anda membedakan antara perangkat blok dan karakter?

6. Gunakan cat untuk menampilkan / etc / hosts dan /etc/resolv.conf. Apa ide Anda tentang tujuan dari file-file ini?
7. Apakah ada file di / etc / skel /? Periksa juga file-file tersembunyi.
8. Tampilkan / proc / cpuinfo. Pada arsitektur apa Linux Anda berjalan?
9. Tampilkan / proc / interrupts. Berapa ukuran file ini? Di mana file ini disimpan?
10. Bisakah Anda masuk direktori / root? Apakah ada file (tersembunyi)?
11. Apakah ifconfig, fdisk, parted, shutdown dan grub-install ada di / sbin? Mengapa biner ini di / sbin dan tidak di / bin?
12. Apakah / var / log file atau direktori? Bagaimana dengan / var / spool?
13. Buka dua perintah perintah (Ctrl-Shift-T di gnome-terminal) atau terminal (Ctrl-Alt-F1, Ctrl-Alt-F2, ...) dan keluarkan siapa saya di keduanya. Kemudian coba gema kata dari satu terminal ke terminal lainnya
14. Baca halaman manual dan jelaskan perbedaan antara / dev / random dan / dev / urandom.

Bagian 7

Manajemen Pengguna Lokal

A. Tujuan

- Mahasiswa memahami penggunaan perintah untuk mengelola pengguna linux
- Mahasiswa dapat mempraktikkan berbagai perintah yang berkaitan dengan pengguna, grup dan kata sandi

B. Teori

Manajemen Pengguna

Manajemen pengguna di Linux dapat dilakukan dengan tiga cara yang saling melengkapi. Anda dapat menggunakan alat grafis yang disediakan oleh distribusi Anda. Alat-alat ini memiliki tampilan dan nuansa yang bergantung pada distribusi. Jika Anda pengguna Linux pemula di sistem rumah Anda, kemudian gunakan alat grafis yang disediakan oleh distribusi Anda. Ini akan memastikan bahwa Anda tidak mengalami masalah. Pilihan lainnya adalah menggunakan alat baris perintah seperti **useradd**, **usermod**, **gpasswd**, **passwd** dan lainnya. Administrator server cenderung menggunakan alat ini, karena mereka akrab dan sangat mirip di banyak distribusi yang berbeda. Bab ini akan fokus pada alat-alat baris perintah ini. Cara ketiga dan lebih ekstrim adalah mengedit file konfigurasi lokal secara langsung menggunakan vi (atau vipw / vigr). Jangan coba ini sebagai pemula dalam sistem produksi!

groups

Pengguna dapat terdaftar dalam grup. Grup memungkinkan Anda menetapkan izin pada tingkat grup alih-alih harus menyetel izin untuk setiap pengguna.

Setiap distribusi Unix atau Linux akan memiliki alat grafis untuk mengelola grup. Pengguna pemula disarankan untuk menggunakan alat grafis ini. Pengguna yang lebih berpengalaman dapat menggunakan alat baris perintah untuk mengelola pengguna, tetapi hati-hati: Beberapa distribusi tidak mengizinkan penggunaan gabungan alat GUI dan CLI untuk mengelola grup (YaST dalam Novell Suse). Administrator senior dapat mengedit file yang relevan secara langsung dengan vi atau vigr.

C. Alat dan Bahan

- Komputer
- Virtual Box dengan Instalasi Linux OS

D. Praktik (Minggu 10 & 11)

- whoami

Perintah **whoami** digunakan untuk menampilkan username anda.

```
[paul@centos7 ~]$ whoami
paul
[paul@centos7 ~]$
```


- **who**

Perintah Who yang akan memberi Anda informasi tentang siapa yang login pada sistem.

```
[paul@centos7 ~]$ who
root      pts/0          2014-10-10 23:07
(10.104.33.101) paul      pts/1          2014-10-10
23:30 (10.104.33.101) laura pts/2          2014-10-
10 23:34 (10.104.33.96) tania pts/3          2014-
10-10 23:39 (10.104.33.91) [paul@centos7 ~]$
```

- **who am i**

Dengan perintah **who am i**, hanya akan menampilkan garis yang menunjuk ke sesi Anda saat ini.

```
[paul@centos7 ~]$ who am i
paul      pts/1          2014-10-10 23:30
(10.104.33.101) [paul@centos7 ~]$
```

- **w**

Perintah w menunjukkan kepada Anda siapa yang login dan apa yang mereka lakukan.

```
[paul@centos7 ~]$ w
 23:34:07 up 31 min,  2 users,  load average: 0.00, 0.01, 0.02
USER      TTY      LOGIN@  IDLE   JCPU   PCPU
WHAT root    pts/0    23:07   15.00s  0.01s
0.01s top paul    pts/1    23:30    7.00s  0.00s
0.00s w [paul@centos7 ~]$
```

- **id**

Perintah id akan memberi Anda ID pengguna Anda, ID grup utama, dan daftar grup tempat Anda bergabung.

```
paul@debian7:~$ id
uid=1000(paul) gid=1000(paul) groups=1000(paul)
```

- **su to another user**

Perintah su memungkinkan pengguna untuk menjalankan shell sebagai pengguna lain.

```
laura@debian7:~$ su
tania Password:
tania@debian7:/home/laura$
```

- **su to root**

Anda juga bisa menggunakan **su** untuk menjadi root, ketika Anda tahu kata sandi root.

```
laura@debian7:~$ su
root Password:
root@debian7:/home/laura#
```

- **su as root**

Anda perlu mengetahui kata sandi pengguna yang ingin Anda gantikan, kecuali Anda masuk sebagai root. Pengguna root dapat menjadi pengguna yang ada tanpa mengetahui kata sandi pengguna itu.

```
root@debian7:~# id
uid=0(root) gid=0(root)
groups=0(root) root@debian7:~# su -
valentina valentina@debian7:~$
```

- **su - \$username**

Secara default, perintah su mempertahankan lingkungan shell yang sama. Untuk menjadi pengguna lain dan juga mendapatkan lingkungan pengguna target, berikan su - perintah yang diikuti oleh nama pengguna target.

```
root@debian7:~# su
laura
laura@debian7:/root$
exit exit
root@debian7:~# su - laura
laura@debian7:~$ pwd
/home/laura
```

- **su -**

Ketika tidak ada nama pengguna yang disediakan untuk su atau su -, perintah akan mengasumsikan root adalah target.

```
tania@debian7:~$ su
- Password:
root@debian7:~#
```

- **Menjalankan sebuah Program sebagai Pengguna lain**

Program sudo memungkinkan pengguna untuk memulai program dengan kredensial pengguna lain. Sebelum ini berfungsi, administrator sistem harus mengatur file / **etc / sudoers**. Ini dapat berguna untuk mendelegasikan tugas administratif ke pengguna lain (tanpa memberikan kata sandi root).

Screenshot di bawah ini menunjukkan penggunaan sudo. Pengguna paul menerima hak untuk menjalankan useradd dengan kredensial root. Ini memungkinkan paul untuk membuat pengguna baru di sistem tanpa menjadi root dan tanpa mengetahui kata sandi root.

```
paul@debian7:~$ /usr/sbin/useradd -m valentina
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
```

Tapi dengan sudo dapat dijalankan

```
paul@debian7:~$ sudo /usr/sbin/useradd -m valentina
[sudo] password for paul:
paul@debian7:~$
```

- **sudo su -**

Pada beberapa sistem Linux seperti Ubuntu dan Xubuntu, pengguna root tidak memiliki kumpulan kata sandi. Ini berarti bahwa tidak mungkin login sebagai root (keamanan ekstra). Untuk melakukan tugas sebagai root, pengguna pertama diberikan semua **sudo rights** melalui / **etc / sudoers**. Bahkan semua pengguna yang menjadi anggota grup admin dapat menggunakan sudo untuk menjalankan semua perintah sebagai root.

```

root@laika:~#      grep      admin
/etc/sudoers
# Members of the admin group may gain root
privileges
%admin    ALL=(ALL)
ALL

```

Hasil akhir dari ini adalah bahwa pengguna dapat mengetikkan `sudo su` - dan menjadi root tanpa harus memasukkan kata sandi root. Perintah `sudo` memang mengharuskan Anda memasukkan kata sandi Anda sendiri. Jadi kata sandi prompt di screenshot di bawah ini untuk `sudo`, bukan untuk `su`.

```

paul@laika:~$ sudo su -
Password:
root@laika:~#

```

- **sudo logging**

Menggunakan `sudo` tanpa otorisasi akan menghasilkan peringatan yang berat:

```

paul@rhel65:~$ sudo su -

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for paul:
paul is not in the sudoers file. This incident will be
reported. paul@rhel65:~$

```

Pengguna root dapat melihat ini di `/var/log/secure` di Red Hat dan di `/var/log/auth.log` pada Debian).

```

root@rhel65:~# tail /var/log/secure | grep sudo | tr -s ' '
Apr 13 16:03:42 rhel65 sudo: paul : user NOT in sudoers ; TTY=pts/0 ; PWD=\
/home/paul ; USER=root ; COMMAND=/bin/su
- root@rhel65:~#

```

1. Manajemen Pengguna

- **/etc/passwd**

Database untuk local User pada Linux (dan hampir disemua sistem Unixes) adalah **/etc/passwd**.

```

[root@RHEL5 ~]# tail /etc/passwd
inge:x:518:524:art
dealer:/home/inge:/bin/ksh
ann:x:519:525:flute
player:/home/ann:/bin/bash
frederik:x:520:526:rubius
poet:/home/frederik:/bin/bash
steven:x:521:527:roman
emperor:/home/steven:/bin/bash
pascale:x:522:528:artist:/home/pascale:/bin/ksh
geert:x:524:530:kernel developer:/home/geert:/bin/bash
wim:x:525:531:master damuti:/home/wim:/bin/bash
sandra:x:526:532:radish stresser:/home/sandra:/bin/bash
annelies:x:527:533:sword
fighter:/home/annelies:/bin/bash
laura:x:528:534:art
dealer:/home/laura:/bin/ksh

```

Seperti yang Anda lihat, file ini berisi tujuh kolom yang dipisahkan oleh titik dua. Kolom berisi nama pengguna, x, id pengguna, id grup utama, deskripsi, nama direktori home, dan shell login.

Informasi lebih lanjut dapat ditemukan dengan mengetikkan `man 5 passwd`.

```
[root@RHEL5 ~]# man 5
passwd
```

- **root**

Pengguna root juga disebut superuser adalah akun paling kuat di sistem Linux Anda. Pengguna ini dapat melakukan hampir semua hal, termasuk pembuatan pengguna lain. Pengguna root selalu memiliki userid 0 (terlepas dari nama akun).

```
[root@RHEL5 ~]# head -1 /etc/passwd
```

- **useradd**

Anda dapat menambahkan pengguna dengan perintah **useradd**. Contoh di bawah ini menunjukkan cara menambahkan pengguna bernama yanina (parameter terakhir) dan pada saat yang sama memaksa pembuatan direktori home (-m), pengaturan nama direktori home (-d), dan pengaturan deskripsi (-c).

```
[root@RHEL5 ~]# useradd -m -d /home/yanina -c "yanina wickmayer"
yanina
[root@RHEL5 ~]# tail -1
/etc/passwd
yanina:x:529:529:yanina
wickmayer:/home/yanina:/bin/bash
```

Pengguna bernama yanina menerima 529 userid dan primary group id 529.

- **/etc/default/useradd**

Baik Red Hat Enterprise Linux dan Debian / Ubuntu memiliki file bernama **/etc/default/useradd** yang berisi beberapa opsi pengguna default. Selain menggunakan cat untuk menampilkan file ini, Anda juga dapat menggunakan **useradd -D**.

```
[root@RHEL4 ~]# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/ba
sh
SKEL=/etc/ske
l
```

- **userdel**

Anda dapat menghapus pengguna yanina dengan **userdel**. Opsi -r dari **userdel** juga akan menghapus direktori home.

```
[root@RHEL5 ~]# userdel -r yanina
```

- **usermod**

Anda dapat memodifikasi properti pengguna dengan perintah **usermod**. Contoh ini menggunakan **Usermod** untuk mengubah deskripsi dari pengguna harry.

```
[root@RHEL4 ~]# tail -1 /etc/passwd
harry:x:516:520:harry
potter:/home/harry:/bin/bash [root@RHEL4 ~]#
usermod -c 'wizard' harry [root@RHEL4 ~]# tail -
1 /etc/passwd
harry:x:516:520:wizard:/home/harry:/bin/bash
```

2. Membuat Home Direktori

Cara termudah untuk membuat direktori home adalah menyediakan opsi **-m** dengan **useradd** (kemungkinan ditetapkan sebagai opsi default di Linux).

Cara yang lebih mudah adalah dengan membuat direktori home secara manual dengan **mkdir** yang juga membutuhkan pengaturan pemilik dan izin pada direktori dengan **chmod** dan **chown** (kedua perintah tersebut dibahas secara mendetail dalam bab lain).

```
[root@RHEL5 ~]# mkdir /home/laura
[root@RHEL5 ~]# chown laura:laura
/home/laura [root@RHEL5 ~]# chmod 700
/home/laura [root@RHEL5 ~]# ls -ld
/home/laura/
drwx----- 2 laura laura 4096 Jun 24 15:17 /home/laura/
```

• /etc/skel/

Ketika menggunakan **useradd** opsi **-m**, direktori **/etc/skel/** disalin ke direktori home yang baru dibuat. Direktori **/etc/skel/** berisi beberapa file (biasanya tersembunyi) yang berisi pengaturan profil dan nilai default untuk aplikasi. Dengan cara ini **/etc/skel/** berfungsi sebagai direktori home default dan sebagai profil pengguna default.

```
[root@RHEL5 ~]# ls -la /etc/skel/
total 48
drwxr-xr-x 2 root root 4096 Apr 1 00:11
. drwxr-xr-x 97 root root 12288 Jun 24
15:36 ..
-rw-r--r-- 1 root root 24 Jul 12 2006 .bash_logout
-rw-r--r-- 1 root root 176 Jul 12 2006 .bash_profile
-rw-r--r-- 1 root root 124 Jul 12 2006 .bashrc
```

3. Menghapus home direktori

Opsi **-r** dari **userdel** akan memastikan bahwa direktori home dihapus bersama dengan akun pengguna.

```
[root@RHEL5 ~]# ls -ld /home/wim/
drwx----- 2 wim wim 4096 Jun 24 15:19
/home/wim/ [root@RHEL5 ~]# userdel -r wim
[root@RHEL5 ~]# ls -ld /home/wim/
ls: /home/wim/: No such file or directory
```

4. login shell

File **/etc/passwd** menentukan shell login untuk pengguna. Dalam screenshot di bawah ini Anda dapat melihat bahwa annelies pengguna akan masuk dengan shell **/bin/bash**, dan laura pengguna dengan shell **/bin/ksh**.

```
[root@RHEL5 ~]# tail -2 /etc/passwd
annelies:x:527:533:sword
fighter:/home/annelies:/bin/bash laura:x:528:534:art
dealer:/home/laura:/bin/ksh
```

Anda dapat menggunakan perintah **usermod** untuk mengubah shell untuk pengguna.

```
[root@RHEL5 ~]# usermod -s /bin/bash laura
[root@RHEL5 ~]# tail -1 /etc/passwd
laura:x:528:534:art
dealer:/home/laura:/bin/bash
```

- **chsh**

Pengguna dapat mengubah shell login mereka dengan perintah **chsh**. Pertama, pengguna **harry** memperoleh daftar shell yang tersedia (dia juga bisa melakukan **cat / etc / shells**) dan kemudian mengubah shell loginnya ke shell Korn (**/ bin / ksh**). Pada login berikutnya, **harry** akan menjadi default **ksh** daripada **bash**.

```
[laura@centos7 ~]$ chsh -l
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/bin/ksh
/bin/tcsh
/bin/csh
[laura@centos7 ~]$
```

Perhatikan bahwa opsi **-l** tidak ada pada **Debian** dan bahwa screenshot di atas mengasumsikan itu shell **ksh** dan **csh** diinstal.

Screenshot di bawah ini menunjukkan bagaimana **laura** dapat mengubah shell default-nya (aktif pada login berikutnya).

```
[laura@centos7 ~]$ chsh -s /bin/ksh
Changing shell for
laura. Password:
Shell changed.
```

Password Pengguna

- **passwd**

Kata sandi pengguna dapat diatur dengan perintah **passwd**. Pengguna harus memberikan kata sandi lama mereka sebelum dua kali memasukkan yang baru.

```
[tania@centos7 ~]$ passwd
Changing password for user
tania. Changing password for
tania. (current) UNIX password:
New password:
BAD PASSWORD: The password is shorter than 8 characters
New password:
BAD PASSWORD: The password is a palindrome
New password:
BAD PASSWORD: The password is too similar to the old one
passwd: Have exhausted maximum number of retries for
service
```

Seperti yang Anda lihat, alat **passwd** akan melakukan beberapa verifikasi dasar untuk mencegah pengguna menggunakan kata sandi yang terlalu sederhana. Pengguna root tidak harus mengikuti aturan ini (akan ada peringatan sekalipun). Pengguna root juga tidak perlu memberikan kata sandi lama sebelum memasukkan kata sandi baru dua kali.

```
root@debian7:~# passwd
tania Enter new UNIX
password: Retype new UNIX
password:
passwd: password updated successfully
```

- **shadow file**

Kata sandi pengguna dienkripsi dan disimpan di **/etc/shadow**. File **/etc/shadow** hanya bisa dibaca dan hanya bisa dibaca oleh root. Kita akan melihat di bagian izin file bagaimana mungkin bagi pengguna untuk mengubah kata sandinya. Untuk saat ini, Anda harus tahu bahwa pengguna dapat mengubah kata sandi mereka dengan perintah **usr/bin/passwd**.

```
[root@centos7 ~]# tail -4 /etc/shadow
paul:$6$ikp2Xta5BT.Tml.p$2TzjNnOYNNQKpwLJqoGJbVsZG5/Fti8ovBRd.VzRbiDS17TE
q\ IaSMH.TeBKNTS/SjlMrUW8qffc0JNORW.BTW1:16338:0:99999:7:::
tania:$6$8Z/zovxj$9qvoqT8i9KIrmN.k4EQwAF5ryz5yzNwEvYjAa9L5XVXQu.z4D1pvMRE
H\ eQpQzvRnqFdKkVj17H5ST.c79HDZw0:16356:0:99999:7:::
laura:$6$glDuTY5e$/NYYLxfHgZFWeoujaXSMcR.Mz.lGOxtcxFocFVJNb98nbTPhWFXfKW
G\ SyYh1WCv6763Wq54.w24Yr3uAZBOM/:16356:0:99999:7:::
valentina:$6$jRZa6PVI$1uQgqR6En9mZB6mKJ3LXRB4CnFko6LRhbh.v4iqUk9MVreui1lv
7\ GxHOUDSKA0N55ZRNhGHa6T2ouFnVno/0o1:16356:0:99999:7:::
[root@centos7 ~]#
```

Berkas **/etc/shadow** berisi sembilan kolom yang dipisahkan dengan titik dua. Sembilan bidang berisi (dari kiri ke kanan) nama pengguna, kata sandi terenkripsi (perhatikan bahwa hanya inge dan laura memiliki kata sandi terenkripsi), hari ketika kata sandi terakhir diubah (hari 1 adalah 1 Januari 1970), jumlah hari kata sandi harus dibiarkan tidak berubah, masa berlaku kata sandi, nomor peringatan hari sebelum kata sandi berakhir, jumlah hari setelah kedaluwarsa sebelum menonaktifkan akun, dan hari ketika akun dinonaktifkan (sekali lagi, sejak tahun 1970). Bidang terakhir belum memiliki arti.

Semua kata sandi dalam screenshot di atas adalah bagian hash **hunter2**

- **encryption dengan passwd**

Kata sandi disimpan dalam format terenkripsi. Enkripsi ini dilakukan oleh fungsi **crypt**. Cara termudah (dan disarankan) untuk menambahkan pengguna dengan kata sandi ke sistem adalah dengan menambahkan pengguna dengan perintah pengguna **useradd -m**, dan kemudian mengatur kata sandi pengguna dengan **passwd**.

```
[root@RHEL4 ~]# useradd -m
xavier [root@RHEL4 ~]# passwd
xavier Changing password for
user xavier. New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated
successfully. [root@RHEL4 ~]#
```

- **encryption dengan openssl**

Cara lain untuk membuat pengguna dengan kata sandi adalah dengan menggunakan opsi -p dari useradd, tetapi opsi itu memerlukan kata sandi terenkripsi. Anda dapat menghasilkan kata sandi terenkripsi ini dengan perintah openssl passwd.

Perintah **passwd openssl** akan menghasilkan beberapa hash yang berbeda untuk kata sandi yang sama, karena ini menggunakan salt.

```
paul@rhel65:~$ openssl passwd hunter2
86jcUNlnGDFpY
paul@rhel65:~$ openssl passwd hunter2
Yj7mDO90Anvq6
paul@rhel65:~$ openssl passwd hunter2
YqDcJeGoDbzKA
paul@rhel65:~$
```

Salt ini dapat dipilih dan terlihat sebagai dua karakter pertama dari hash.

```
paul@rhel65:~$ openssl passwd -salt 42 hunter2
42ZrbtPlZe8G.
paul@rhel65:~$ openssl passwd -salt 42 hunter2
42ZrbtPlZe8G.
paul@rhel65:~$ openssl passwd -salt 42 hunter2
42ZrbtPlZe8G
.
paul@rhel65:
~$
```

Contoh ini menunjukkan cara membuat pengguna dengan kata sandi.

```
root@rhel65:~# useradd -m -p $(openssl passwd hunter2) mohamed
```

Perhatikan bahwa perintah ini menempatkan kata sandi dalam sejarah perintah Anda

- **/etc/login.defs**

File **/etc/login.defs** berisi beberapa pengaturan default untuk kata sandi pengguna seperti pengaturan kata sandi **aging and length**. (Anda juga akan menemukan batas numerik dari id pengguna dan ID grup dan apakah direktori home harus dibuat secara default).

```
root@rhel65:~# grep ^PASS /etc/login.defs
PASS_MAX_DAYS    99999
PASS_MIN_DAYS    0
PASS_MIN_LEN     5
PASS_WARN_AGE    7
```

Debian juga memiliki file ini.

```
root@debian7:~# grep PASS /etc/login.defs
# PASS_MAX_DAYS    Maximum number of days a password may be
used.
# PASS_MIN_DAYS    Minimum number of days allowed between password
changes.
# PASS_WARN_AGE    Number of days warning given before a password
expires. PASS_MAX_DAYS    99999
PASS_MIN_DAYS
0
PASS_WARN_AGE
7
```



```
#PASS_CHANGE_TRI
ES
#PASS_ALWAYS_WA
RN
#PASS_MIN_L
EN
#PASS_MAX_L
EN
#
NO_PASSWORD_CONSOLE
root@debian7:
~#
```

- **chage**

Perintah **chage** dapat digunakan untuk menetapkan tanggal kedaluwarsa untuk akun pengguna (-E), mengatur usia kata sandi minimum (-m) dan maksimum (-M), tanggal kedaluwarsa kata sandi, dan mengatur jumlah hari peringatan sebelum tanggal kedaluwarsa kata sandi. Sebagian besar fungsi ini juga tersedia dari perintah **passwd**. Opsi -l dari **chage** akan menampilkan pengaturan ini untuk pengguna.

```
root@rhel65:~# chage -l
paul
Last password change                : Mar 27,
2014
Password expires                    :
never                               Password                inactive
: never                             Account              expires
: never Minimum number of days between password change
: 0
Maximum number of days between password change      :
99999
Number of days of warning before password expires   :
7 root@rhel65:~#
```

- **disabling a password**

Kata sandi di **/etc/shadow** tidak dapat dimulai dengan tanda seru. Ketika kolom kedua di **/etc/passwd** dimulai dengan tanda seru, maka kata sandi tidak dapat digunakan.

Menggunakan fitur ini sering disebut "**disabling, suspending, or locking**" akun pengguna. Selain **vi** (atau **vipw**) Anda juga dapat melakukannya dengan **usermod**.

Perintah pertama di screenshot berikutnya akan menunjukkan password hash dari **laura** di **/etc/shadow**. Perintah berikutnya menonaktifkan kata sandi **laura**, sehingga mustahil bagi **Lucille** untuk mengotentikasi menggunakan kata sandi ini.

```
root@debian7:~# grep laura /etc/shadow | cut -c1-70
laura:$6$JYj4JZqp$stwwWACp3OtE1R2aZuE87j.nbW.puDkNUYV7mCHfCVMa3CoDU
JV root@debian7:~# usermod -L laura
```

Seperti yang Anda lihat di bawah ini, hash kata sandi hanya didahului dengan tanda seru.

```
root@debian7:~# grep laura /etc/shadow | cut -c1-70
laura:!!$6$JYj4JZqp$stwwWACp3OtE1R2aZuE87j.nbW.puDkNUYV7mCHfCVMa3CoD
UJ root@debian7:~#
```

Pengguna root (dan pengguna dengan hak **sudo** pada su) masih akan dapat su ke akun laura (karena kata sandi tidak diperlukan di sini). Perhatikan juga bahwa **laura** masih bisa masuk jika ia telah mengatur ssh tanpa password!

```
root@debian7:~# su - laura
laura@debian7:~$
```

Anda dapat membuka kembali akun dengan **usermod -U**.

```
root@debian7:~# usermod -U
laura
root@debian7:~# grep laura /etc/shadow | cut -c1-
70
laura:$6$JYj4JZqp$stwwWACp3OtE1R2aZuE87j.nbW.puDkNUYV7mCHfCVMa3CoDU
JV
```

Perhatikan perbedaan kecil dalam opsi baris perintah passwd, usermod, dan **useradd** pada distribusi Linux yang berbeda. Verifikasi file lokal saat menggunakan fitur seperti "**disabling, suspending, or locking**" pada akun pengguna dan kata sandinya.

- mengedit file lokal

Jika Anda masih ingin mengedit / **etc** / **passwd** atau / **etc** / **shadow** secara manual, setelah mengetahui perintah-perintah ini untuk manajemen kata sandi, maka gunakan **vipw** bukan **vi** (**m**) secara langsung. Alat **vipw** akan melakukan penguncian file dengan benar.

```
[root@RHEL5 ~]# vipw /etc/passwd
vipw: the password file is busy (/etc/ptmp present)
```

1. system profile

Baik **bash** dan shell **ksh** akan memverifikasi keberadaan / **etc** / **profile** dan **source** jika ada.

Ketika membaca skrip ini, Anda akan melihat (baik pada Debian dan Red Hat Enterprise Linux) yang membangun variabel lingkungan **PATH** (antara lain). Skrip mungkin juga mengubah variabel **PS1**, mengatur **HOSTNAME** dan mengeksekusi lebih banyak skrip seperti / **etc** / **inputrc**

Screenshot ini menggunakan **grep** untuk menunjukkan manipulasi **PATH** di / **etc** / **profile** di Debian.

```
root@debian7:~# grep PATH /etc/profile
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
export PATH
root@debian7:~#
```

Screenshot ini menggunakan **grep** untuk menunjukkan manipulasi **PATH** di / **etc** / **profile** pada RHEL7 / CentOS7

```
[root@centos7 ~]# grep PATH /etc/profile
case ":${PATH}:" in
    PATH=$PATH:$1
    PATH=$1:$PATH
```

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE
HISTCONTROL [root@centos7 ~]#
```

Pengguna root dapat menggunakan skrip ini untuk mengatur alias, fungsi, dan variabel untuk setiap pengguna pada sistem.

- **~/.bash_profile**

Ketika file ini ada di direktori home, maka **bash** akan memberinya sumber. Pada Debian Linux 5/6/7 file ini tidak ada secara default.

RHEL7 / CentOS7 menggunakan kecil ~ / **.bash_profile** di mana ia memeriksa keberadaan

~ / **.bashrc** lalu sumbernya. Ia juga menambahkan \$ HOME / bin ke variabel \$ PATH.

```
[root@rhel7 ~]# cat /home/paul/.bash_profile
# .bash_profile

# Get the aliases and
functions if [ -f ~/.bashrc
]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/b

in export PATH
[root@rhel7 ~]#
```

- **~/.bash_login**

Ketika **.bash_profile** tidak ada, maka bash akan memeriksa ~ / **.bash_login** dan sumbernya. Baik Debian maupun Red Hat tidak memiliki file ini secara default.

- **~/.profile**

Ketika tidak ada ~ / **.bash_profile** dan ~ / **.bash_login** ada, maka bash akan memverifikasi keberadaan ~ / **.profile** dan menjalankannya. File ini tidak ada secara default di Red Hat.

Pada Debian skrip ini dapat mengeksekusi ~ / **.bashrc** dan akan menambahkan \$ HOME / bin ke \$ PATH variabel.

```
root@debian7:~# tail -11 /home/paul/.profile
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it
    exists if [ -f
"$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it
exists if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi
```

RHEL / CentOS tidak memiliki file ini secara default.

- **~/.bashrc**

Skrip `~ / .bashrc` sering kali bersumber dari skrip lain. Mari kita lihat apa yang dilakukannya secara default.

Red Hat menggunakan sangat sederhana `~ / .bashrc`, memeriksa `/ etc / bashrc` dan mengambilnya. Ini juga menyisakan ruang untuk alias dan fungsi khusus.

```
[root@rhel7 ~]# cat /home/paul/.bashrc
# .bashrc

# Source global
definitions if [ -f
/etc/bashrc ]; then
    . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging
feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
```

On Debian this script is quite a bit longer and configures `$PS1`, some history variables and a number of active and inactive aliases.

```
root@debian7:~# wc -l /home/paul/.bashrc
110 /home/paul/.bashrc
```

- **`~/.bash_logout`**

Ketika keluar dari **bash**, ia dapat mengeksekusi `~ / .bash_logout`. Debian menggunakan kesempatan ini untuk mengosongkan layar konsol.

```
serena@deb503:~$ cat ~/.bash_logout
# ~/.bash_logout: executed by bash(1) when login shell exits.

# when leaving the console clear the screen to increase

privacy if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
```

Red Hat Enterprise Linux 5 akan dengan mudah memanggil perintah `/ usr / bin / clear` dalam skrip ini.

```
[serena@rhel53 ~]$ cat ~/.bash_logout
# ~/.bash_logout

/usr/bin/clear
```

Red Hat Enterprise Linux 6 dan 7 membuat file ini, tetapi biarkan kosong (kecuali untuk komentar).

```
paul@rhel65:~$ cat ~/.bash_logout
# ~/.bash_logout
```

Groups

- **`groupadd`**

Grup dapat dibuat dengan perintah **groupadd**. Contoh di bawah ini menunjukkan pembuatan lima grup (kosong).

```

root@laika:~# groupadd tennis
root@laika:~# groupadd
football root@laika:~#
groupadd snooker
root@laika:~# groupadd
formula1 root@laika:~#
groupadd salsa

```

- **group file**

Pengguna dapat menjadi anggota beberapa grup. Keanggotaan grup ditentukan oleh **/etc/group** mengajukan.

```

root@laika:~# tail -5
/etc/group tennis:x:1006:
football:x:100
7:
snooker:x:1008
:
formula1:x:100
9:
salsa:x:1010:
root@laika:~#

```

Bidang pertama adalah nama grup. Bidang kedua adalah kata sandi grup (terenkripsi) (bisa kosong). Bagian ketiga adalah identifikasi kelompok atau GID. Bidang keempat adalah daftar anggota, kelompok-kelompok ini tidak memiliki anggota.

- **groups**

Pengguna dapat mengetikkan perintah grup untuk melihat daftar grup tempat pengguna berada.

```

[harry@RHEL4b ~]$
groups harry sports
[harry@RHEL4b ~]$

```

- **usermod**

Keanggotaan grup dapat dimodifikasi dengan perintah **useradd** atau **usermod**.

```

root@laika:~# usermod -a -G tennis inge
root@laika:~# usermod -a -G tennis katrien
root@laika:~# usermod -a -G salsa katrien
root@laika:~# usermod -a -G snooker sandra
root@laika:~# usermod -a -G formula1
annelies root@laika:~# tail -5 /etc/group
tennis:x:1006:inge,katrien
football:x:1007:
snooker:x:1008:sandra
formula1:x:1009:anneli
es
salsa:x:1010:katrien
root@laika:~#

```

Hati-hati saat menggunakan **usermod** untuk menambahkan pengguna ke grup. Secara default, perintah **usermod** akan menghapus pengguna dari setiap grup di mana dia adalah anggota jika grup tersebut tidak terdaftar dalam perintah! Menggunakan **-a** (append) akan mencegah perilaku ini.

- **groupmod**

Anda dapat mengubah nama grup dengan perintah **groupmod**.

```
root@laika:~# groupmod -n darts
snooker root@laika:~# tail -5
/etc/group tennis:x:1006:inge,katrien
football:x:1007:
formula1:x:1009:annelies
salsa:x:1010:katrien
darts:x:1008:sandra
```

- **groupdel**

Anda dapat secara permanen menghapus grup dengan perintah **groupdel**.

```
root@laika:~# groupdel
tennis root@laika:~#
```

- **gpasswd**

Anda dapat mendelegasikan kontrol keanggotaan grup ke pengguna lain dengan perintah **gpasswd**. Dalam contoh di bawah ini kami mendelegasikan izin untuk menambahkan dan menghapus anggota grup ke serena untuk grup **sports**. Kemudian kami su untuk serena dan tambahkan harry ke grup **sports**

```
[root@RHEL4b ~]# gpasswd -A serena
sports [root@RHEL4b ~]# su - serena
[serena@RHEL4b ~]$ id harry
uid=516(harry) gid=520(harry)
groups=520(harry) [serena@RHEL4b ~]$ gpasswd
-a harry sports Adding user harry to group
sports
[serena@RHEL4b ~]$ id harry
uid=516(harry) gid=520(harry)
groups=520(harry),522(sports) [serena@RHEL4b ~]$ tail -1
/etc/group sports:x:522:serena,venus,harry
[serena@RHEL4b ~]$
```

Administrator grup tidak harus menjadi anggota grup. Mereka dapat menghapus diri mereka sendiri dari grup, tetapi ini tidak memengaruhi kemampuan mereka untuk menambah atau menghapus anggota.

```
[serena@RHEL4b ~]$ gpasswd -d serena
sports Removing user serena from group
sports [serena@RHEL4b ~]$ exit
```

Informasi tentang administrator grup disimpan di file / etc / **gshadow**.

```
[root@RHEL4b ~]# tail -1
/etc/gshadow
sports:!:serena:venus,harry
[root@RHEL4b ~]#
```

Untuk menghapus semua grup administrator dari grup, gunakan perintah **gpasswd** untuk mengatur daftar administrator kosong.

```
[root@RHEL4b ~]# gpasswd -A "" sports
```

- **newgrp**

Anda dapat memulai shell anak dengan grup utama sementara baru menggunakan **newgrp** perintah.

```

root@rhel65:~/prigroup# touch
standard.txt
root@rhel65:~/prigroup# ls
-l total 0
-rw-r--r--. 1 root root 0 Apr 13 17:49
standard.txt root@rhel65:~/prigroup# echo $SHLVL
1
root@rhel65:~/prigroup# newgrp
tennis root@rhel65:~/prigroup# echo
$SHLVL
2
root@rhel65:~/prigroup# touch
newgrp.txt root@rhel65:~/prigroup# ls
-l
tota
l 0
-rw-r--r--. 1 root tennis 0 Apr 13 17:49
newgrp.txt
-rw-r--r--. 1 root root 0 Apr 13 17:49
standard.txt root@rhel65:~/prigroup# exit
exit
root@rhel65:~/prigrou
p#

```

E. Latihan

1. Jalankan perintah yang hanya menampilkan nama pengguna Anda yang sedang login.
2. Tampilkan daftar semua pengguna yang masuk.
3. Tampilkan daftar semua pengguna yang masuk termasuk perintah yang sedang mereka jalankan saat ini.
4. Tampilkan nama pengguna Anda dan identifikasi pengguna unik Anda (userid).
5. Gunakan su untuk beralih ke akun pengguna lain (kecuali Anda adalah root, Anda akan memerlukan kata sandi akun yang lain). Dan kembali ke akun sebelumnya.
6. Sekarang gunakan su - untuk beralih ke pengguna lain dan perhatikan perbedaannya. Perhatikan bahwa su - membuat Anda masuk ke direktori home **Tania**.
7. Cobalah untuk membuat akun pengguna baru (saat menggunakan akun pengguna normal Anda). ini harus gagal. (Detail tentang menambahkan akun pengguna dijelaskan dalam bab berikutnya.)
8. Sekarang coba yang sama, tetapi dengan **sudo** sebelum perintah Anda.

Manajemen pengguna

1. Buat akun pengguna bernama serena, termasuk direktori home dan deskripsi (atau komentar) yang bertuliskan Serena Williams. Lakukan semua ini dalam satu perintah tunggal.
2. Buat pengguna bernama venus, termasuk direktori home, bash shell, deskripsi yang dibaca Venus Williams semuanya dalam satu perintah tunggal.
3. Pastikan bahwa kedua pengguna memiliki entri yang benar di / etc / passwd, / etc / shadow dan / etc / group.
4. Verifikasi bahwa direktori home mereka telah dibuat.
5. Buat pengguna bernama einstime dengan / bin / date sebagai shell logon default-nya.
6. Apa yang terjadi ketika Anda masuk dengan pengguna einstime? Dapatkah Anda memikirkan contoh dunia nyata yang berguna untuk mengubah shell login pengguna menjadi aplikasi?

7. Buat file bernama welcome.txt dan pastikan setiap pengguna baru akan melihat file ini di direktori home mereka.
8. Verifikasi pengaturan ini dengan membuat (dan menghapus) akun pengguna uji.
9. Ubah shell login default untuk pengguna serena ke / bin / bash. Verifikasi sebelum dan sesudah Anda melakukan perubahan ini.

Password pengguna

1. Atur kata sandi untuk serena to hunter2.
2. Juga mengatur kata sandi untuk venus dan kemudian mengunci akun pengguna venus dengan usermod. Verifikasi penguncian di / etc / shadow sebelum dan sesudah Anda menguncinya.
3. Gunakan passwd -d untuk menonaktifkan kata sandi serena. Verifikasi garis serena di / etc / shadow sebelum dan sesudah penonaktifan.
4. Apa perbedaan antara mengunci akun pengguna dan menonaktifkan kata sandi akun pengguna seperti yang baru saja kita lakukan dengan usermod -L dan passwd -d?
5. Coba ubah kata sandi serena menjadi serena sebagai serena.
6. Pastikan serena harus mengubah kata sandinya dalam 10 hari.
7. Pastikan setiap pengguna baru perlu mengubah kata sandi setiap 10 hari.
8. Ambil cadangan sebagai root dari / etc / shadow. Gunakan vi untuk menyalin hash pemburu yang terenkripsi dari venus ke serena. Bisakah serena sekarang masuk dengan hunter2 sebagai kata sandi?
9. Mengapa menggunakan vipw bukan vi? Apa yang bisa menjadi masalah saat menggunakan vi atau vim?
10. Gunakan chsh untuk mendaftar semua shell (hanya berfungsi di RHEL / CentOS / Fedora), dan bandingkan dengan cat / dll / cangkang. Opsi useradd mana yang memungkinkan Anda untuk menamai direktori home?
11. Bagaimana Anda bisa melihat apakah kata sandi pengguna serena terkunci atau tidak terkunci? Berikan solusi dengan grep dan solusi dengan passwd

user profiles

1. Buat daftar semua file profil di sistem Anda.
2. Baca isi masing-masing ini, sering kali mereka mencari skrip tambahan.
3. Masukkan variabel unik, alias dan fungsi di masing-masing file tersebut.
4. Coba beberapa cara berbeda untuk mendapatkan shell (su, su -, ssh, tmux, gnome-terminal, Ctrl- alt-F1, ...) dan verifikasi mana dari variabel kustom Anda, alias dan fungsi ada di lingkungan Anda .
5. Apakah Anda juga tahu urutan di mana mereka dieksekusi?
6. Ketika aplikasi tergantung pada pengaturan di \$ HOME / .profile, apakah itu penting \$ HOME / .bash_profile ada atau tidak

groups

1. Buat grup tennis, football dan sports
2. Dalam satu perintah, buat venus anggota tennis dan sports
3. Ganti nama grup sepakbola menjadi foot.
4. Gunakan vi untuk menambahkan serena ke grup tennis.
5. Gunakan perintah id untuk memverifikasi bahwa serena adalah anggota tennis.
6. Buat seseorang yang bertanggung jawab untuk mengelola keanggotaan grup foot dan sports. Uji bahwa itu berhasil.

Bagian 8

Sistem Manajemen Paket Perangkat Lunak

A. Tujuan

- Mahasiswa dapat menginstal menggunakan apt pada sistem linux
- Mahasiswa dapat memperbarui repositori pada sistem linux

B. Teori

Paket perangkat lunak datang dalam berbagai format. Ubuntu menggunakan **dpkg** yang merupakan sistem kemasan Linux pertama untuk memasukkan informasi dependency (ketergantungan); itu mendapat namanya dari distribusi Linux yang dikembangkan pada (Debian). Format lain termasuk rpm (digunakan pada Red Hat, SuSE, dan sistem lainnya), yum, Konfigurasi GNU dan Sistem Build dan tar terkompresi. Format seperti tar terkompresi, yang populer sebelum pengenalan dpkg, digunakan lebih jarang hari ini karena mereka memerlukan lebih banyak pekerjaan pada bagian dan tidak menyediakan ketergantungan dan pengecekan kompatibilitas yang ditawarkan dpkg.

dpkg (Sistem manajemen paket Debian) disebut sebagai sistem manajemen dpkg, atau hanya dpkg. Sistem ini adalah kumpulan lebih dari 20 utilitas yang mengelola dan melaporkan paket dpkg, baik yang diinstal pada sistem dan yang tersedia dari repositori online. Berikan perintah `dpkg-query -f='${Package} ${Version} ${Architecture}\n'` atau `dpkg-query -f='${Package} ${Version} ${Architecture}\n'` untuk menampilkan daftar utilitas dpkg. file deb Utilitas dpkg bekerja dengan file yang namanya diakhiri dengan .deb dan disebut sebagai file deb atau paket (perangkat lunak).

APT (Advanced Package Tool) adalah kumpulan utilitas yang, bersama dengan dpkg, bekerja dengan paket perangkat lunak. APT mengunduh paket perangkat lunak, sementara dpkg menginstal, menghapus, memelihara, mengatur dependensi, dan melaporkan paket perangkat lunak. Berikan perintah `dpkg-query -f='${Package} ${Version} ${Architecture}\n'` atau `dpkg-query -f='${Package} ${Version} ${Architecture}\n'` yang tepat untuk menampilkan daftar utilitas APT (dan beberapa hal lainnya).

APT: Keeps the System Up-to-Date

APT (Advanced Package Tool) adalah kumpulan utilitas yang mengunduh, menginstal, menghapus, meningkatkan, dan melaporkan paket perangkat lunak. Paket utilitas APT mengunduh paket dan memanggil utilitas dpkg untuk memanipulasi paket begitu mereka berada di sistem lokal. Untuk informasi lebih lanjut, lihat www.debian.org/doc/manuals/apt-howto.

Repositori

Repositori menyimpan koleksi paket perangkat lunak dan informasi terkait, termasuk header yang menggambarkan setiap paket dan memberikan informasi tentang paket lain yang tergantung pada paket. Ubuntu mengelola repositori untuk setiap rilisnya. Kategori paket perangkat lunak

Paket perangkat lunak dari repositori Ubuntu dibagi menjadi beberapa kategori, termasuk yang berikut:

- **main** Ubuntu-supported open-source software
- **universe** Community-maintained open-source software

- **multiverse** Software restricted by copyright or legal issues
- **restricted** Proprietary device drivers
- Paket **backports** dari rilis Ubuntu selanjutnya yang tidak tersedia untuk rilis sebelumnya.

APT memilih paket dari repositori yang dicari berdasarkan kategori yang ditentukan dalam file `sources.list` (selanjutnya). Anda tidak perlu mengkonfigurasi ulang APT untuk menginstal Menentukan Repositori untuk APT ke Pencarian

File `/etc/apt/sources.list` menentukan repositori yang digunakan APT saat Anda memintanya untuk mencari atau menginstal paket. Anda harus memodifikasi file `sources.list` untuk mengaktifkan APT untuk mengunduh perangkat lunak dari repositori nondefault. Anda dapat menggunakan perangkat lunak `property-gtk` untuk menampilkan jendela Sumber Perangkat Lunak untuk mengubah `sources.list` atau Anda dapat menggunakan editor untuk mengubahnya.

C. Alat dan Bahan

- Komputer
- Virtual Box dengan Instalasi Linux OS

D. Praktik (Minggu 12)

1. aptitude untuk menginstal shell `tcsh`, yang merupakan bagian dari paket `tcsh`:

```
$ sudo aptitude install tcsh
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading extended state information
Initializing package states... Done
Building tag database... Done
The following NEW packages will be installed:
tcsh
0 packages upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0B/338kB of archives. After unpacking 709kB will be used.
Writing extended state information... Done
Selecting previously deselected package tcsh.
(Reading database ... 119619 files and directories currently installed.)
Unpacking tcsh (from .../tcsh_6.14.00-7_i386.deb) ...
Setting up tcsh (6.14.00-7) ...
...
```

2. `apt-get update`

Digunakan untuk update informasi paket dari sumber file instalasi yang telah didaftarkan .

Format penggunaan: `apt-get update`

`#apt-get update`

3. `apt-get install`

Digunakan untuk menambahkan / menginstall paket baru .

Format penggunaan: `apt-get install package_name`

`# apt-get install mc`

4. `apt-get remove`

Digunakan untuk menghapus paket yang sudah ada.

Format penggunaan: `apt-get remove package_name`.

`# apt-get remove mc`

5. `apt-get-u upgrade`

Digunakan untuk mengupgrade dari paket yang sudah ada.

Format penggunaan: `apt-get-u upgrade`.

`# apt-get-u upgrade`

6. `apt-get clean`

Digunakan untuk menghapus paket-paket yang tidak digunakan.

Format penggunaan: `apt-get clean`

`# apt-get clean`

7. `apt-get autoclean`

Digunakan untuk menghapus paket-paket yang tidak digunakan.

Format penggunaan: `apt-get autoclean`

`# apt-get autoclean`

8. `apt-show-versions`

Digunakan untuk menampilkan versi dari paket yang ada .

Format penggunaan: `apt-show-versions package name`

`# apt-show-versions ls`

9. `apt-cache -search`

Digunakan untuk mencari nama paket yang ada dalam database daftar paket .

Format penggunaan: `apt-cache search package_name`

`# apt-cache search mc`

10. `apt-cache show`

Digunakan untuk menampilkan informasi detail dari paket yang ada .

Format penggunaan: `apt-search show package_name`

`# apt-search show mc`

11. `apt-cache showpkg`

Digunakan menampilkan informasi secara umum paket serta keterkaitan dengan paket lain.

Format penggunaan: `apt-cache showpkg`

`# apt-cache showpkg mc`

12. `apt-cache depends`

Digunakan untuk mengetahui ketergantungan suatu paket dengan paket yang lain.

Format penggunaan: `apt-cache depends on package name`

`# apt-cache depends mc`

E. Latihan

- Install Paket Untuk Pemrograman Web (Apache, PHP dan Mysql)

Bagian 9

Ekspansi Shell

A. Tujuan

- Mahasiswa memahami penggunaan argumen pada ekspansi shell
- Mahasiswa memahami penggunaan komentar pada ekspansi shell

B. Teori

perintah dan argumen

Bab ini memperkenalkan Anda untuk **ekspansi shell** dengan melihat lebih dekat pada perintah dan argumen. Mengetahui ekspansi shell penting karena banyak perintah pada sistem Linux Anda diproses dan kemungkinan besar diubah oleh shell sebelum dieksekusi.

Antarmuka baris perintah atau shell yang digunakan pada kebanyakan sistem Linux disebut bash, yang berarti Bourne lagi shell. Shell bash menggabungkan fitur dari sh (shell Bourne asli), csh (c shell), dan ksh (shell Korn).

Bab ini sering menggunakan perintah echo untuk mendemonstrasikan fitur shell. Gema perintahnya sangat sederhana: itu mengulangi input yang diterimanya.

```
paul@laika:~$ echo Burtonville
Burtonville
paul@laika:~$ echo Smurfs are blue
Smurfs are blue
```

Argumen

Salah satu fitur utama dari shell adalah melakukan scan baris perintah. Ketika Anda memasukkan perintah pada command prompt shell dan tekan tombol enter, maka shell akan mulai memindai garis itu, memotongnya dalam argumen. Saat memindai garis, shell dapat membuat banyak perubahan pada argumen yang anda ketikkan.

Proses ini disebut ekspansi shell. Ketika shell selesai memindai dan memodifikasi garis itu, maka itu akan dieksekusi.

• penghapusan **white space**

Bagian yang dipisahkan oleh satu atau lebih ruang putih berurutan (atau tab) dianggap sebagai argumen terpisah, setiap white space dihapus. Argumen pertama adalah perintah yang akan dieksekusi, argumen lain diberikan kepada perintah. Shell secara efektif memotong perintah anda menjadi satu atau lebih argumen.

Perintah

- perintah eksternal atau builtin

Tidak semua perintah berada di luar shell, ada yang builtin. Perintah eksternal adalah program yang memiliki biner sendiri dan berada di suatu tempat di sistem file. Banyak perintah eksternal berada di / **bin** atau / **sbin**. Perintah builtin merupakan bagian integral dari program shell itu sendiri.

C. Alat dan Bahan

- Komputer
- Virtual Box dengan Instalasi Linux OS

D. Praktik (Minggu 13)

1. Argumen

Ini menjelaskan mengapa empat baris perintah berikut ini sama setelah ekspansi shell.

```
[paul@RHELv4u3 ~]$ echo Hello World
Hello World
[paul@RHELv4u3 ~]$ echo Hello    World
Hello World
[paul@RHELv4u3 ~]$ echo    Hello    World
Hello World
[paul@RHELv4u3 ~]$    echo        Hello        World
Hello World
```

Perintah echo akan menampilkan setiap argumen yang diterimanya dari shell. perintah echo juga akan menambahkan white space baru di antara argumen yang diterimanya.

- single quotes

Anda dapat mencegah penghapusan white spaces dengan quoting spasi. Isi string yang dikutip dianggap sebagai satu argumen. Pada screenshot di bawah echo hanya menerima satu argumen.

```
[paul@RHEL4b ~]$ echo 'A line with    single
quotes' A line with    single    quotes
[paul@RHEL4b ~]$
```

- double quotes

Anda juga dapat mencegah penghapusan white spaces dengan double quoting spasi. Sama seperti di atas, echo hanya menerima satu argumen.

```
[paul@RHEL4b ~]$ echo "A line with    double
quotes" A line with    double    quotes
[paul@RHEL4b
~]$
```

- echo and quotes

Baris yang dikutip dapat menyertakan karakter khusus yang dikenal oleh perintah echo (saat menggunakan echo -e). Tangkapan layar di bawah ini menunjukkan cara menggunakan \n untuk baris baru dan \t untuk tab (biasanya delapan white spaces).

```
[paul@RHEL4b ~]$ echo -e "A line with \na
newline" A line with
a newline
[paul@RHEL4b ~]$ echo -e 'A line with \na
newline' A line with
a newline
[paul@RHEL4b ~]$ echo -e "A line with \ta
tab" A line with      a tab
[paul@RHEL4b ~]$ echo -e 'A line with \ta
tab' A line with      a tab
[paul@RHEL4b ~]$
```

Perintah echo dapat menghasilkan lebih dari white spaces, tabs dan baris baru. Lihat di halaman manual untuk daftar opsi.

2. commands

- **type**

Untuk mengetahui apakah perintah yang diberikan kepada shell akan dieksekusi sebagai perintah eksternal atau sebagai perintah bawaan, gunakan perintah jenis.

```
paul@laika:~$ type
cd cd is a shell
builtin
paul@laika:~$ type
cat cat is /bin/cat
```

Seperti yang Anda lihat, perintah cd adalah builtin dan perintah cat adalah eksternal. Anda juga dapat menggunakan perintah ini untuk menunjukkan apakah perintahnya alias atau tidak.

```
paul@laika:~$ type
ls
ls is aliased to `ls --
color=auto'
```

- **running external commands**

Beberapa perintah memiliki versi builtin dan eksternal. Ketika salah satu dari perintah ini dijalankan, versi builtin akan menjadi prioritas. Untuk menjalankan versi eksternal, Anda harus memasukkan path lengkap ke perintah.

```
paul@laika:~$ type -a
echo echo is a shell
builtin echo is
/bin/echo

paul@laika:~$ /bin/echo Running the external echo
command... Running the external echo command...
```

- **which**

Perintah yang akan mencari binari dalam environment variabel \$PATH (variabel akan dijelaskan nanti). Pada screenshot di bawah ini, ditentukan bahwa cd adalah builtin, dan ls, cp, rm, mv, mkdir, pwd, dan yang merupakan perintah eksternal.

```
[root@RHEL4b ~]# which cp ls cd mkdir pwd
```

```
/bin/cp
/bin/ls
/usr/bin/which: no cd in (/usr/kerberos/sbin:/usr/kerberos/bin:...
/bin/mkdir
/bin/pwd
```

3. Membuat sebuah *alias*

Shell memungkinkan Anda membuat alias. Alias sering digunakan untuk membuat nama yang mudah diingat untuk suatu perintah yang ada atau dengan mudah menyediakan parameter.

```
[paul@RHELv4u3 ~]$ cat
count.txt one
two
thr
ee
[paul@RHELv4u3 ~]$ alias dog=tac [paul@RHELv4u3 ~]$ dog count.txt three
two one
```

- **abbreviate commands**

Sebuah alias juga bisa berguna untuk menyingkat perintah yang ada.

```
paul@laika:~$ alias ll='ls -lh --
color=auto' paul@laika:~$ alias c='clear'
paul@laika:~$
```

- **default options**

Alias dapat digunakan untuk menyediakan perintah dengan opsi default. Contoh di bawah ini menunjukkan cara mengatur opsi `-i` default ketika mengetik `rm`.

```
[paul@RHELv4u3 ~]$ rm -i winter.txt
rm: remove regular file `winter.txt'?
no [paul@RHELv4u3 ~]$ rm winter.txt
[paul@RHELv4u3 ~]$ ls winter.txt
ls: winter.txt: No such file or
directory [paul@RHELv4u3 ~]$ touch
winter.txt [paul@RHELv4u3 ~]$ alias
rm='rm -i' [paul@RHELv4u3 ~]$ rm
winter.txt
rm: remove regular empty file `winter.txt'? no
[paul@RHELv4u3 ~]$
```

Beberapa distribusi mengaktifkan alias default untuk melindungi pengguna dari penghapusan file secara tidak sengaja (`'rm-i '`, `'mv -i '`, `'cp -i '`)

- **viewing aliases**

Anda dapat memberikan satu alias atau lebih sebagai argumen ke perintah alias untuk mendapatkan definisinya. Tidak memberikan argumen memberikan daftar lengkap alias saat ini.

```
paul@laika:~$ alias c
ll alias c='clear'
alias ll='ls -lh --color=auto
```

- **unalias**

Anda dapat melakukan Undo sebuah alias dengan perintah `unalias`.

```
[paul@RHEL4b ~]$ which
rm
/bin/
rm
[paul@RHEL4b ~]$ alias rm='rm -
i' [paul@RHEL4b ~]$ which rm
alias rm='rm -
i'
/bin/rm
[paul@RHEL4b ~]$ unalias
rm
[paul@RHEL4b ~]$ which
rm
/bin/
rm
[paul@RHEL4b
~]$
```

4. Menampilkan shell expansion

Anda dapat menampilkan ekspansi shell dengan set -x, dan berhenti menampilkannya dengan set + x. Anda mungkin ingin menggunakan ini lebih lanjut dalam kursus ini, atau ketika ragu tentang apa yang dilakukan shell dengan perintah Anda.

```
[paul@RHELv4u3 ~]$ set -x
++ echo -ne
'\033]0;paul@RHELv4u3:~\007'
[paul@RHELv4u3 ~]$ echo $USER
+ echo
paul paul
++ echo -ne
'\033]0;paul@RHELv4u3:~\007'
[paul@RHELv4u3 ~]$ echo \ $USER
+ echo '$USER'
$USER
++ echo -ne
'\033]0;paul@RHELv4u3:~\007'
[paul@RHELv4u3 ~]$ set +x
+ set +x
[paul@RHELv4u3 ~]$ echo $USER
Paul
```

E. Latihan

1. Berapa banyak argumen dalam baris ini (tidak termasuk perintah itu sendiri).

```
touch '/etc/cron/cron.allow' 'file 42.txt' "file 33.txt"
```

2. Apakah perintah **tac** adalah shell builtin?
3. Apakah ada alias yang sudah ada untuk rm?
4. Baca halaman manual, pastikan Anda memahami opsi -i rm. Buat dan hapus file untuk menguji opsi -i.
5. Jalankan: **alias rm = 'rm -i'**. Uji alias Anda dengan file tes. Apakah ini berfungsi seperti yang diharapkan?
6. Buat daftar semua alias saat ini.
7. Buat alias yang disebut 'city' yang menampilkan kampung halaman Anda.

Gunakan alias Anda untuk menguji bahwa itu berhasil.

8. Jalankan set -x untuk menampilkan ekspansi shell untuk setiap perintah.
9. Uji fungsionalitas set -x dengan mengeksekusi alias city dan rm Anda.
10. Jalankan set + x untuk berhenti menampilkan ekspansi shell.
11. Hapus alias city Anda.
12. Apa lokasi cat dan perintah passwd?
13. Jelaskan perbedaan antara perintah berikut:

```
echo
```

```
/bin/echo
```

14. Jelaskan perbedaan antara perintah berikut:

```
echo Hello
```

```
echo -n Hello
```

15. Tampilan A B C dengan dua spasi antara B dan C.

Bagian 10

Operator Kontrol & Variabel Shell

A. Tujuan

- Mahasiswa memahami penggunaan pemrograman shell untuk memecahkan beragam masalah
- Mahasiswa dapat menggunakan operator logika dalam pemrograman shell linux
-

B. Teori

Anda dapat menempatkan dua atau lebih perintah pada baris yang sama yang dipisahkan oleh titik koma; `;`. Shell akan memindai garis hingga mencapai titik koma. Semua argumen sebelum titik koma ini akan dianggap sebagai perintah terpisah dari semua argumen setelah titik koma. Kedua seri akan dieksekusi secara berurutan dengan shell menunggu setiap perintah selesai sebelum memulai yang berikutnya.

- **& ampersand**

Ketika sebuah baris berakhir dengan ampersand `&`, shell tidak akan menunggu perintah selesai. Anda akan mendapatkan prompt shell Anda kembali, dan perintah dijalankan di latar belakang. Anda akan mendapat pesan ketika perintah ini selesai dieksekusi di latar belakang.

- **combining `&&` and `||`**

Anda dapat menggunakan logika AND dan logika OR ini untuk menulis struktur **if-then-else** pada baris perintah. Contoh ini menggunakan `echo` untuk menampilkan apakah perintah `rm` berhasil

- **end of line backslash**

Gari terakhir dalam **backslash** terbalik dilanjutkan pada baris berikutnya. Shell tidak menafsirkan karakter newline dan akan menunggu pada ekspansi shell dan eksekusi baris perintah sampai baris baru tanpa backslash ditemui.

- Tanda \$ dolar

Karakter penting lainnya yang ditafsirkan oleh shell adalah tanda dolar `$`. Shell akan mencari **environment variable** seperti string yang mengikuti tanda dolar dan menggantikannya dengan nilai variabel (atau dengan apa-apa jika variabel tidak ada).

- **env**

Perintah **env** tanpa opsi akan menampilkan daftar **exported variables**. Perbedaan dengan `set` dengan opsi adalah `set` yang mencantumkan semua variabel, termasuk yang tidak diekspor ke **child shells**. Tetapi **env** juga dapat digunakan untuk memulai cangkang yang bersih (cangkang tanpa lingkungan yang diwariskan). Perintah **env -i** membersihkan lingkungan untuk **child shells**.

- **delineate variables**

Sampai sekarang, kita telah melihat bahwa bash menafsirkan variabel mulai dari tanda dolar, berlanjut sampai munculnya karakter non-alfanumerik pertama yang bukan garis bawah. Dalam beberapa situasi, ini bisa menjadi masalah. Masalah ini dapat diatasi dengan tanda kurung kurawal seperti pada contoh ini.

C. Alat dan Bahan

- Komputer
- Virtual Box dengan Instalasi Linux OS

D. Praktik (Minggu 14)

1. Operator-Operator Kontrol

- **semicolon**

```
[paul@RHELv4u3 ~]$ echo Hello
Hello
[paul@RHELv4u3 ~]$ echo World
World
[paul@RHELv4u3 ~]$ echo Hello ; echo World
Hello World
[paul@RHELv4u3 ~]$
```

- **& ampersand**

```
[paul@RHELv4u3 ~]$ sleep 20
& [1] 7925
[paul@RHELv4u3 ~]$
...wait 20
seconds...
[paul@RHELv4u3 ~]$
[1]+  Done                  sleep 20
```

- **\$? dollar question mark**

Kode keluar dari perintah sebelumnya disimpan dalam variabel shell \$?. Sebenarnya \$? adalah parameter shell dan bukan variabel, karena Anda tidak dapat menetapkan nilai ke \$?.

```
paul@debian5:~/test$ touch
file1 paul@debian5:~/test$
echo $?
0
paul@debian5:~/test$ rm
file1 paul@debian5:~/test$
echo $?
0
paul@debian5:~/test$ rm file1
rm: cannot remove `file1': No such file or
directory paul@debian5:~/test$ echo $?
1
paul@debian5:~/tes
t$
```

- **&& double ampersand**

Shell akan menafsirkan && sebagai sebuah **logical AND**. Saat menggunakan && perintah kedua dijalankan hanya jika yang pertama berhasil (mengembalikan status nol keluar).

```
paul@barry:~$ echo first && echo
second first
second
paul@barry:~$ zecho first && echo second
-bash: zecho: command not found
```

Contoh lain dari prinsip logical AND. Contoh ini dimulai dengan **cd** diikuti oleh **ls**, lalu **cd** yang tidak bekerja yang tidak diikuti oleh **ls**.

```
[paul@RHELv4u3 ~]$ cd gen && ls
file1 file3 File55 fileab FileAB
fileabc file2 File4 FileA Fileab
fileab2 [paul@RHELv4u3 gen]$ cd gen && ls
-bash: cd: gen: No such file or directory
```

- **|| double vertical bar**

simbol **||** mewakili logika OR . Perintah kedua dijalankan hanya ketika perintah pertama gagal (mengembalikan status keluar non-nol).

```
paul@barry:~$ echo first || echo second ; echo
third first
third
paul@barry:~$ zecho first || echo second ; echo third
-bash: zecho: command not
found second
third
paul@barry:
~$
```

Contoh Lain dari prinsip logika OR

```
[paul@RHELv4u3 ~]$ cd gen || ls
[paul@RHELv4u3 gen]$ cd gen || ls
-bash: cd: gen: No such file or directory
file1 file3 File55 fileab FileAB
fileabc file2 File4 FileA Fileab
fileab2
```

- **combining && and ||**

```
paul@laika:~/test$ rm file1 && echo It worked! || echo It
failed! It worked!
paul@laika:~/test$ rm file1 && echo It worked! || echo It failed!
rm: cannot remove `file1': No such file or directory
It failed!
paul@laika:~/test$
```

- **# pound sign**

Semua yang ditulis setelah tanda pound (#) diabaikan oleh shell. Ini berguna untuk menulis sebuah **shell comment** tetapi tidak memiliki pengaruh pada eksekusi perintah atau ekspansi shell.

```
paul@debian4:~$ mkdir test      # we create a directory
paul@debian4:~$ cd test        ##### we enter the
directory paul@debian4:~/test$ ls      # is it empty ?
paul@debian4:~/test$
```

- **\ escaping special characters**

Karakter backslash \ memungkinkan penggunaan karakter kontrol, tetapi tanpa shell interpreting, ini disebut **escaping characters**.

```
[paul@RHELv4u3 ~]$ echo hello \;
world hello ; world
```

```
[paul@RHELv4u3 ~]$ echo hello\ \ \
world hello    world
[paul@RHELv4u3 ~]$ echo escaping \\\ \#\ \& \"\
\' escaping \ # & " '
[paul@RHELv4u3 ~]$ echo escaping
\\\?*\\"\' escaping \?*"'
```

- **end of line backslash**

```
[paul@RHEL4b ~]$ echo This command line \
> is split in three \
> parts
This command line is split in three parts
[paul@RHEL4b ~]$
```

2. shell variables

Dalam bab ini kita belajar untuk mengelola variabel lingkungan di shell. Variabel-variabel ini sering dibutuhkan oleh aplikasi.

- **Tanda \$ dolar**

Ini adalah beberapa contoh menggunakan \$ HOSTNAME, \$ USER, \$ UID, \$ SHELL, dan \$ HOME.

```
[paul@RHELv4u3 ~]$ echo This is the $SHELL shell
This is the /bin/bash shell
[paul@RHELv4u3 ~]$ echo This is $SHELL on computer
$HOSTNAME This is /bin/bash on computer
RHELv4u3.localdomain [paul@RHELv4u3 ~]$ echo The userid of
$USER is $UID
The userid of paul is 500
[paul@RHELv4u3 ~]$ echo My homedir is
$HOME My homedir is /home/paul
```

- **case sensitive**

Contoh berikut menunjukkan bahwa shell variable adalah case sensitive

```
[paul@RHELv4u3 ~]$ echo Hello
$USER Hello paul
[paul@RHELv4u3 ~]$ echo Hello $user
Hello
```

- **creating variables**

Contoh ini menciptakan variabel \$ MyVar dan menetapkan nilainya. Ia kemudian menggunakan echo untuk memverifikasi nilainya.

```
[paul@RHELv4u3 gen]$ MyVar=555
[paul@RHELv4u3 gen]$ echo
$MyVar
555
[paul@RHELv4u3 gen]$
```

- **quotes**

Perhatikan bahwa tanda kutip ganda masih memungkinkan parsing variabel, sedangkan tanda kutip tunggal mencegah hal ini.

```
[paul@RHELv4u3 ~]$ MyVar=555
[paul@RHELv4u3 ~]$ echo
$MyVar
```

```
555
[paul@RHELv4u3 ~]$ echo "$MyVar"
555
[paul@RHELv4u3 ~]$ echo '$MyVar'
$MyVar
```

Shell bash akan mengganti variabel dengan nilainya dalam garis kutip ganda, tetapi tidak dalam baris yang dikutip tunggal.

```
paul@laika:~$
city=Burtonville
paul@laika:~$ echo "We are in $city
today." We are in Burtonville today.
paul@laika:~$ echo 'We are in $city
today.' We are in $city today.
```

- **set**

Anda dapat menggunakan perintah **set** untuk menampilkan daftar variabel lingkungan. Pada sistem Ubuntu dan Debian, perintah yang ditetapkan juga akan menampilkan fungsi shell setelah variabel shell. Gunakan **set | more** lanjut untuk melihat variabel-variabel itu.

- **unset**

Gunakan perintah **unset** untuk menghapus variabel dari lingkungan shell anda.

```
[paul@RHEL4b ~]$
MyVar=8472 [paul@RHEL4b
~]$ echo $MyVar
8472
[paul@RHEL4b ~]$ unset MyVar
[paul@RHEL4b ~]$ echo $MyVar

[paul@RHEL4b ~]$
```

- **\$PS1**

Variabel **\$ PS1** menentukan prompt shell Anda. Anda dapat menggunakan karakter khusus backslash yang lolos seperti `\u` untuk nama pengguna atau `\w` untuk direktori kerja.

Dalam contoh ini, kami mengubah nilai **\$ PS1** beberapa kali.

```
paul@deb503:~$
PS1=prompt prompt
promptPS1='prompt
' prompt
prompt PS1='> '
>
> PS1='\u@\h$
'
paul@deb503$
paul@deb503$
PS1='\u@\h:\w$'
paul@deb503:~$
```

Untuk menghindari kesalahan yang tidak dapat diperbaiki, Anda dapat mengatur permintaan pengguna normal menjadi hijau dan prompt root menjadi merah. Tambahkan yang berikut ini ke **.bashrc** Anda untuk prompt pengguna hijau:

```
# color prompt by
paul
RED='\[\033[01;31m\'
WHITE='\[\033[01;00m\'
]\'
GREEN='\[\033[01;32m\'
]\'
BLUE='\[\033[01;34m\'
,
export PS1="${debian_chroot:+($debian_chroot)}$GREEN\u$WHITE@$BLUE\h$WHITE\w\$"
"
```

• \$PATH

Variabel \$ PATH menentukan di mana shell mencari perintah untuk dieksekusi (kecuali perintah dibangun atau di-alias). Variabel ini berisi daftar direktori, dipisahkan oleh titik dua.

```
[[paul@RHEL4b ~]$ echo
$PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bi
n:
```

Shell tidak akan mencari di direktori saat ini untuk mengeksekusi perintah! (Mencari executable di direktori saat ini menyediakan cara mudah untuk meretas PC-DOS komputer). Jika Anda ingin shell untuk mencari di direktori saat ini, tambahkan a. di akhir \$ PATH Anda.

```
[paul@RHEL4b ~]$
PATH=$PATH:. [paul@RHEL4b
~]$ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin
:.. [paul@RHEL4b ~]$
```

Jalan Anda mungkin berbeda ketika menggunakan su daripada su - karena yang terakhir akan mengambil lingkungan dari pengguna target. Pengguna root biasanya memiliki direktori / sbin yang ditambahkan ke \$ PATH variabel.

```
[paul@RHEL3 ~]$ su
Password:
[root@RHEL3 paul]# echo $PATH
/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin
[root@RHEL3 paul]#
exit [paul@RHEL3 ~]$
su - Password:
[root@RHEL3 ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bi
n: [root@RHEL3 ~]#
```

• env

Perhatikan di screenshot ini bahwa bash akan mengatur variabel \$ SHELL saat startup.

```
[paul@RHEL4b ~]$ bash -c 'echo $SHELL $HOME $USER'
/bin/bash /home/paul paul
[paul@RHEL4b ~]$ env -i bash -c 'echo $SHELL $HOME $USER'
/bin/bash
[paul@RHEL4b ~]$
```

Anda dapat menggunakan perintah `env` untuk mengatur `$ LANG`, atau variabel lainnya, hanya untuk satu kejadian bash dengan satu perintah. Contoh di bawah ini menggunakan ini untuk menunjukkan pengaruh `$ LANG` variabel pada file globbing (lihat bab tentang file globbing).

```
[paul@RHEL4b test]$ env LANG=C bash -c 'ls File[a-z]' Filea Fileb
[paul@RHEL4b test]$ env LANG=en_US.UTF-8 bash -c 'ls File[a-z]' Filea FileA Fileb FileB
[paul@RHEL4b test]$
```

- **export**

Anda dapat mengeksport variabel shell ke shell lain dengan perintah **export**. Ini akan mengeksport variabel ke child shells..

```
[paul@RHEL4b ~]$ var3=three
[paul@RHEL4b ~]$ var4=four
[paul@RHEL4b ~]$ export var4
[paul@RHEL4b ~]$ echo $var3
$var4 three four
[paul@RHEL4b ~]$ bash
[paul@RHEL4b ~]$ echo $var3
$var4 four
```

Tapi itu tidak akan mengeksport ke shell induk (screenshot sebelumnya dilanjutkan).

```
[paul@RHEL4b ~]$ export var5=five
[paul@RHEL4b ~]$ echo $var3 $var4
$var5 four five
[paul@RHEL4b ~]$
exit exit
[paul@RHEL4b ~]$ echo $var3 $var4
$var5 three four
[paul@RHEL4b ~]$
```

- **delineate variables**

```
[paul@RHEL4b ~]$
prefix=Super
[paul@RHEL4b ~]$ echo Hello $prefixman and
$prefixgirl
Hello
and
[paul@RHEL4b ~]$ echo Hello ${prefix}man and
${prefix}girl
Hello Superman and
Supergirl
[paul@RHEL4b ~]$
```

- **unbound variables**

Contoh di bawah ini mencoba untuk menampilkan nilai variabel `$ MyVar`, tetapi gagal karena variabel tidak ada. Secara default shell akan menampilkan apa-apa ketika sebuah variabel tidak terikat (tidak ada).

```
[paul@RHELv4u3 gen]$ echo $MyVar

[paul@RHELv4u3 gen]$
```


Namun demikian, opsi **nounset** shell yang dapat Anda gunakan untuk menghasilkan kesalahan ketika variabel tidak ada.

```
paul@laika:~$ set -u
paul@laika:~$ echo $Myvar
bash: Myvar: unbound
variable paul@laika:~$ set
+u paul@laika:~$ echo
$Myvar

paul@laika:~$
```

Dalam bash shell **set -u** identik dengan **set -o nounset** dan juga **set + u** identik dengan **set + o nounset**.

E. Latihan

operator kontrol

Setiap pertanyaan dapat dijawab oleh satu baris perintah!

1. Saat Anda mengetik **passwd**, file mana yang dieksekusi?
2. Jenis file apa itu?
3. Jalankan perintah **pwd** dua kali. (ingat 0.)
4. Jalankan **ls** setelah **cd / etc**, tetapi hanya jika **cd / etc** tidak salah.
5. Jalankan **cd / etc** setelah **cd dll**, tetapi hanya jika **cd dll** gagal.
6. Echo **it worked** ketika **menyentuh test42** bekerja, dan echo **it failed** ketika **touch** gagal. Semua pada satu baris perintah sebagai pengguna biasa (bukan root). Uji baris ini di direktori home Anda dan di **/bin/**.
7. Jalankan **sleep 6**, apa perintah ini lakukan?
8. Jalankan **sleep 200** di latar belakang (jangan tunggu sampai selesai).
9. Tulis baris perintah yang menjalankan **file rm55**. Baris perintah Anda harus mencetak **'sukses'** jika **file55** dihapus, dan cetak **'gagal'** jika ada masalah.

variabel shell

1. Gunakan echo untuk menampilkan Hello diikuti oleh nama pengguna Anda. (gunakan variabel bash!)
2. Buat jawaban variabel dengan nilai 42.
3. Salin nilai \$ LANG ke \$ MyLANG.
4. Daftar semua variabel shell saat ini.
5. Daftar semua variabel shell yang diekspor.
6. Apakah perintah env dan set menampilkan variabel Anda?
7. Hancurkan variabel jawaban Anda.
8. Buat dua variabel, dan ekspor salah satunya.
9. Tampilkan variabel yang diekspor dalam cangkang anak interaktif.
10. Buat variabel, beri nilai 'Dumb', buat variabel lain dengan nilai 'do'. Menggunakan echo dan dua variabel untuk menggemakan Dumbledore.
11. Temukan daftar karakter backslash escape dalam manual bash. Tambahkan waktu ke Prompt PS1 anda.