Praktik Sistem Operasi

Mei - Juli 2012

Program Pra-S2 Ilmu Komputer
FMIPA UGM, Yogyakarta

husni@mail.ugm.ac.id
husni@if.trunojoyo.ac.id
Komputasi.wordpress.com

Tentang Saya

- Husni
- Email: husni@mail.ugm.ac.id, husni@if.trunojoyo.ac.id
- Website: komputasi.wordpress.com
- Facebook: facebook.com/lunix96
- Kuliah: Computer Architecture (Organization), Operating System, Computer Network, Automata & Computation, Data (Text & Web) Mining, Information Retrieval, Distributed Computing, Web Engineering.
- Riset: ***Modern Search Engine***
  - Web Mining, Crawling, Retrieval
  - Web Semantics, Natural Language Processing
  - Social Network, Recommender System, Web Technology
  - Distributed & Paralellel Processing

Sistem Operasi?

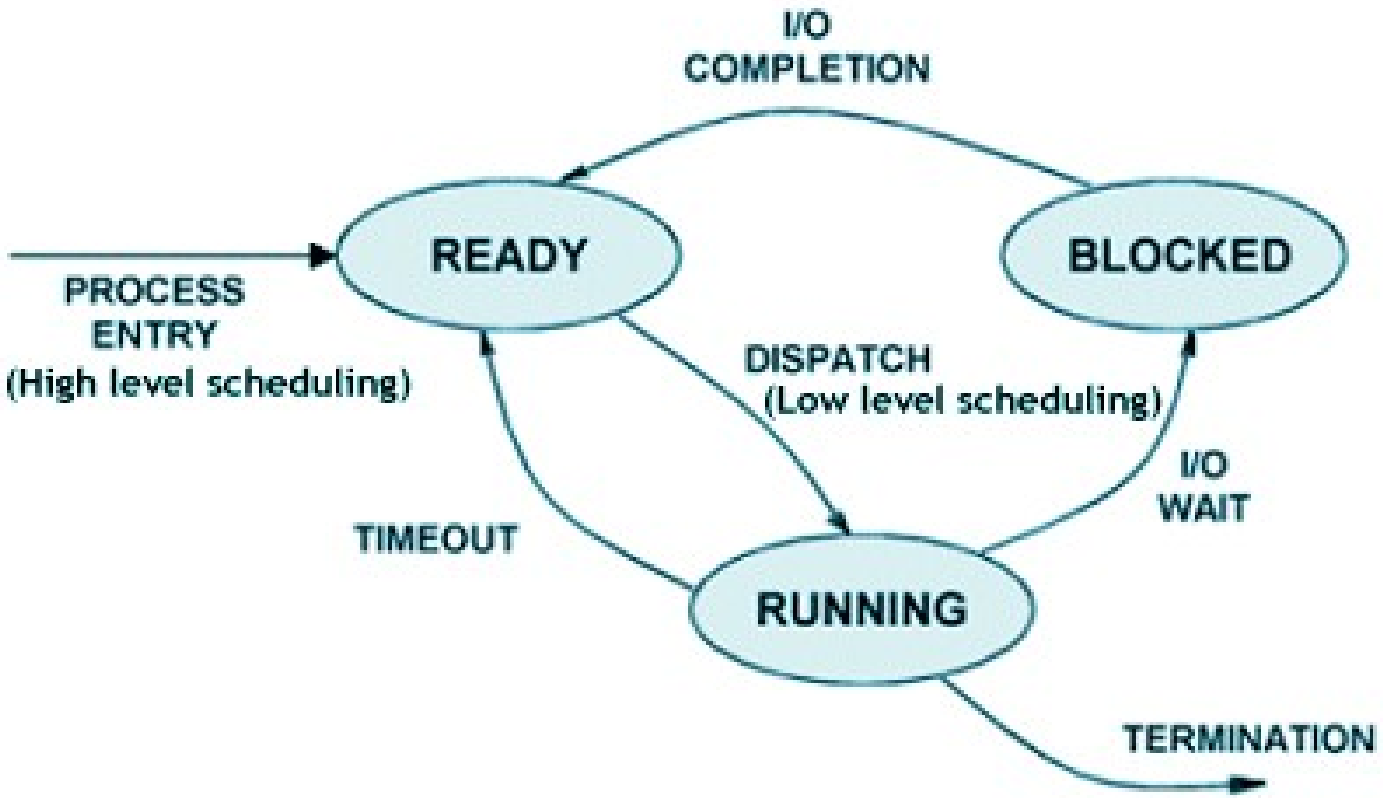
- Ilmu tentang mengatur atau mengelola agar **sistem** komputer ber**operasi** dengan baik
- Pengaturan ada lpada evel sistem, bukan aplikasi
- Penerjemahan cara pengaturan berkehidupan manusia ke dalam pengoperasian sistem komputer
- Contoh: Toilet umum, Antrian di Perempatan Jalan, dan Bangjo (*Traffic Light*).

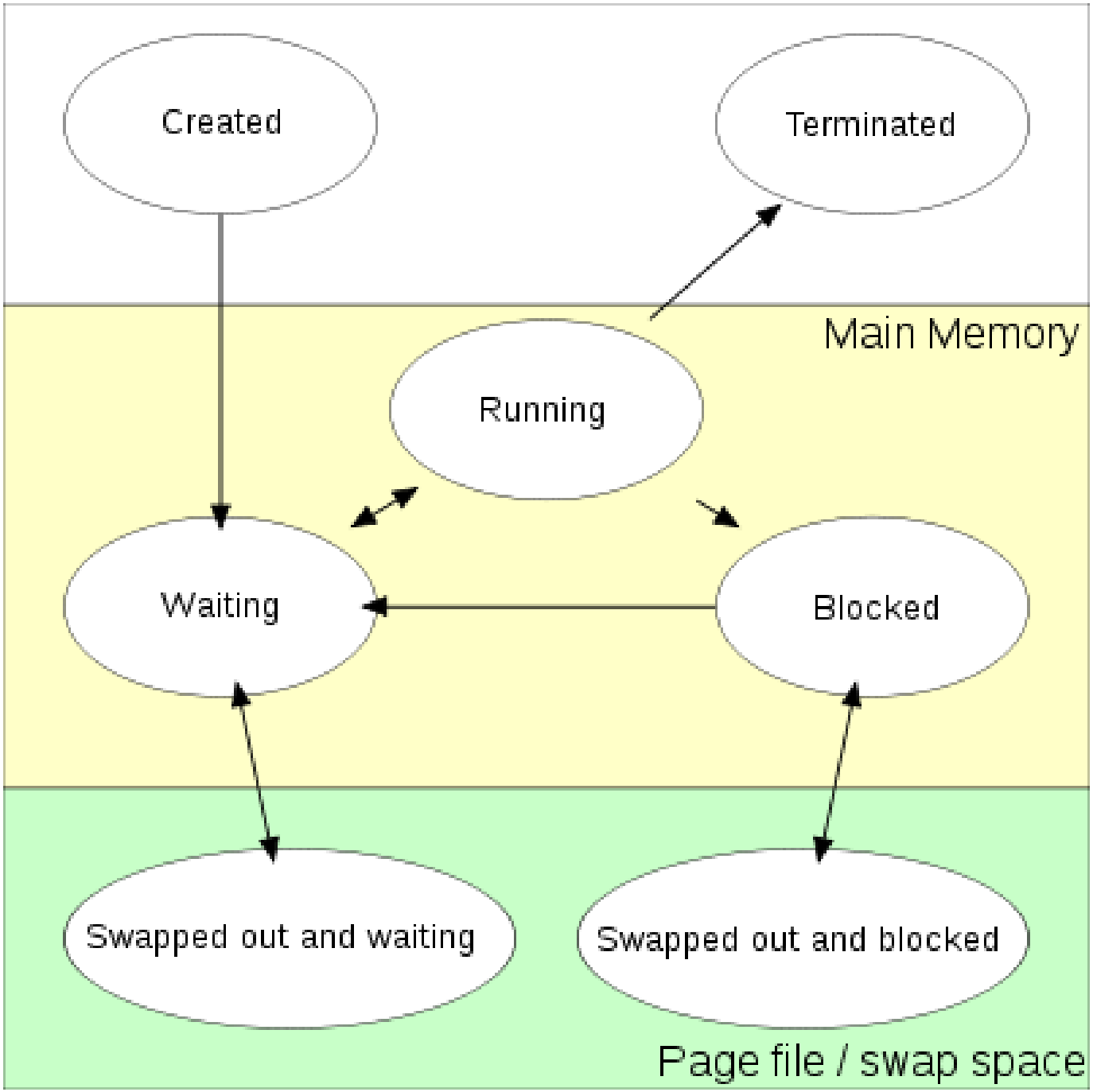
Apa yang dikelola?

- Proses-proses – Processor
- Ruang Penyimpanan – Memory
- Perangkat Keluar Masuk – I/O

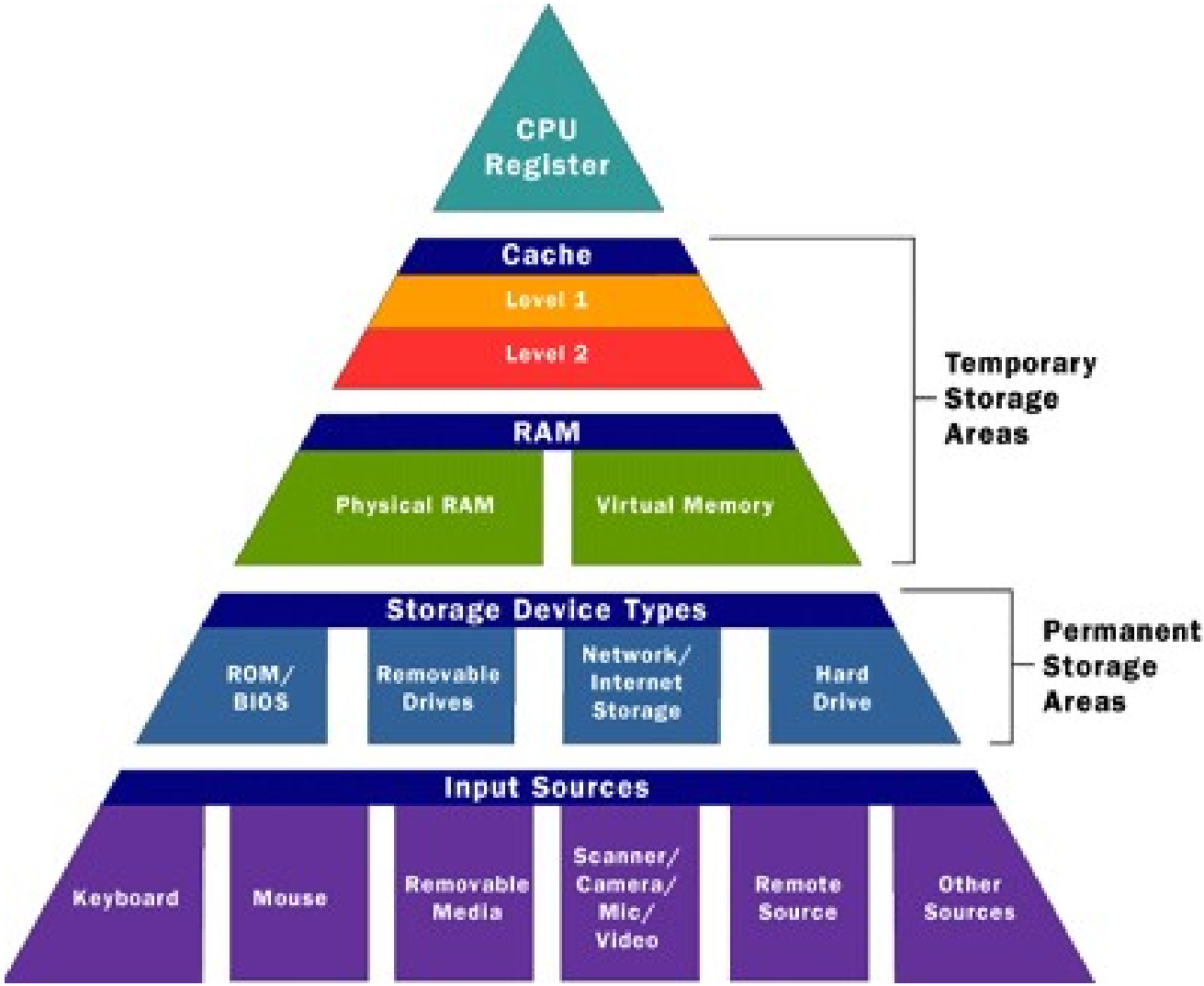
Proses(or)

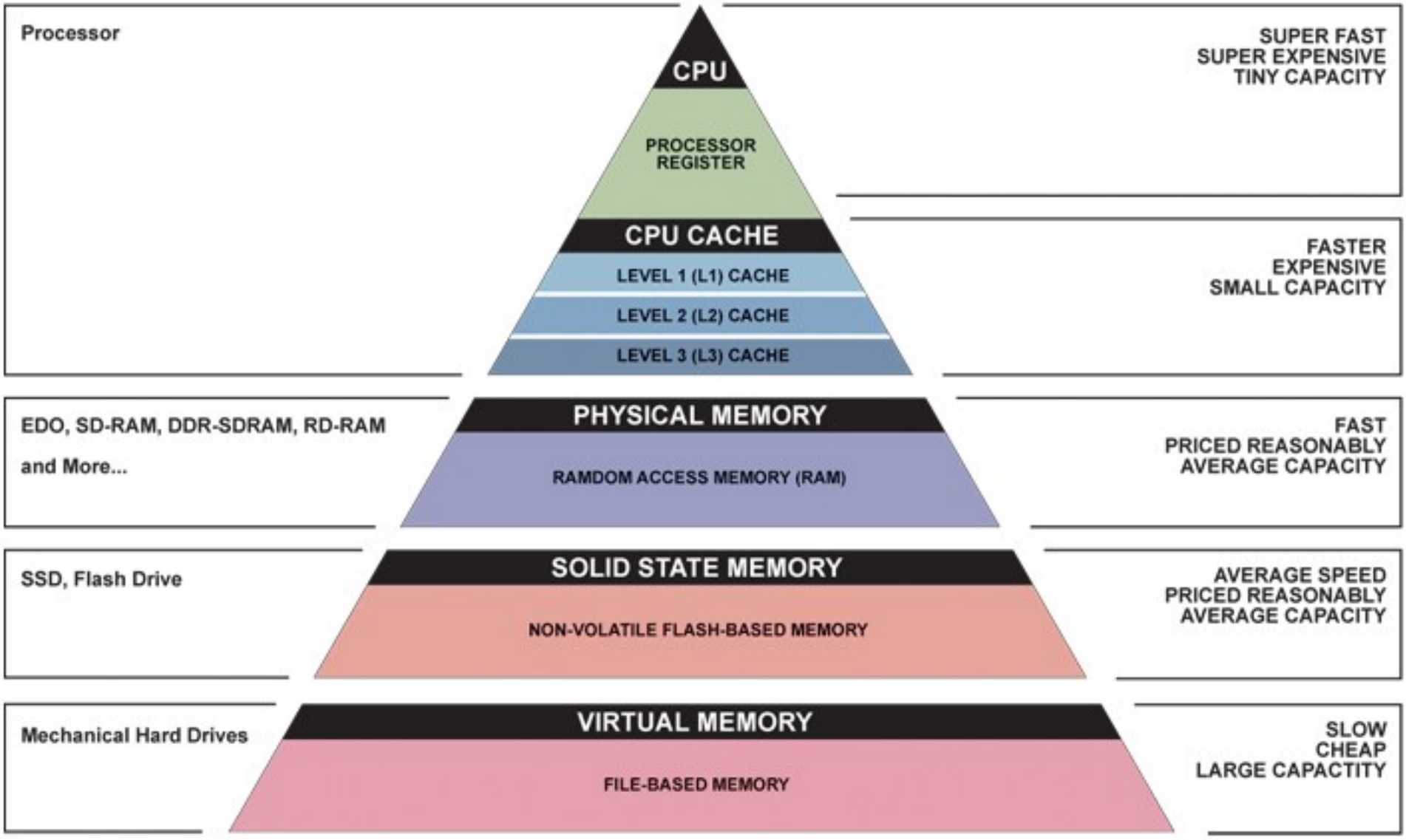

intel
Core™ i7


I/O
COMPLETION
READY
BLOCKED
PROCESS
ENTRY
(High level scheduling)
DISPATCH
(Low level scheduling)
I/O
WAIT
TIMEOUT
RUNNING
TERMINATION

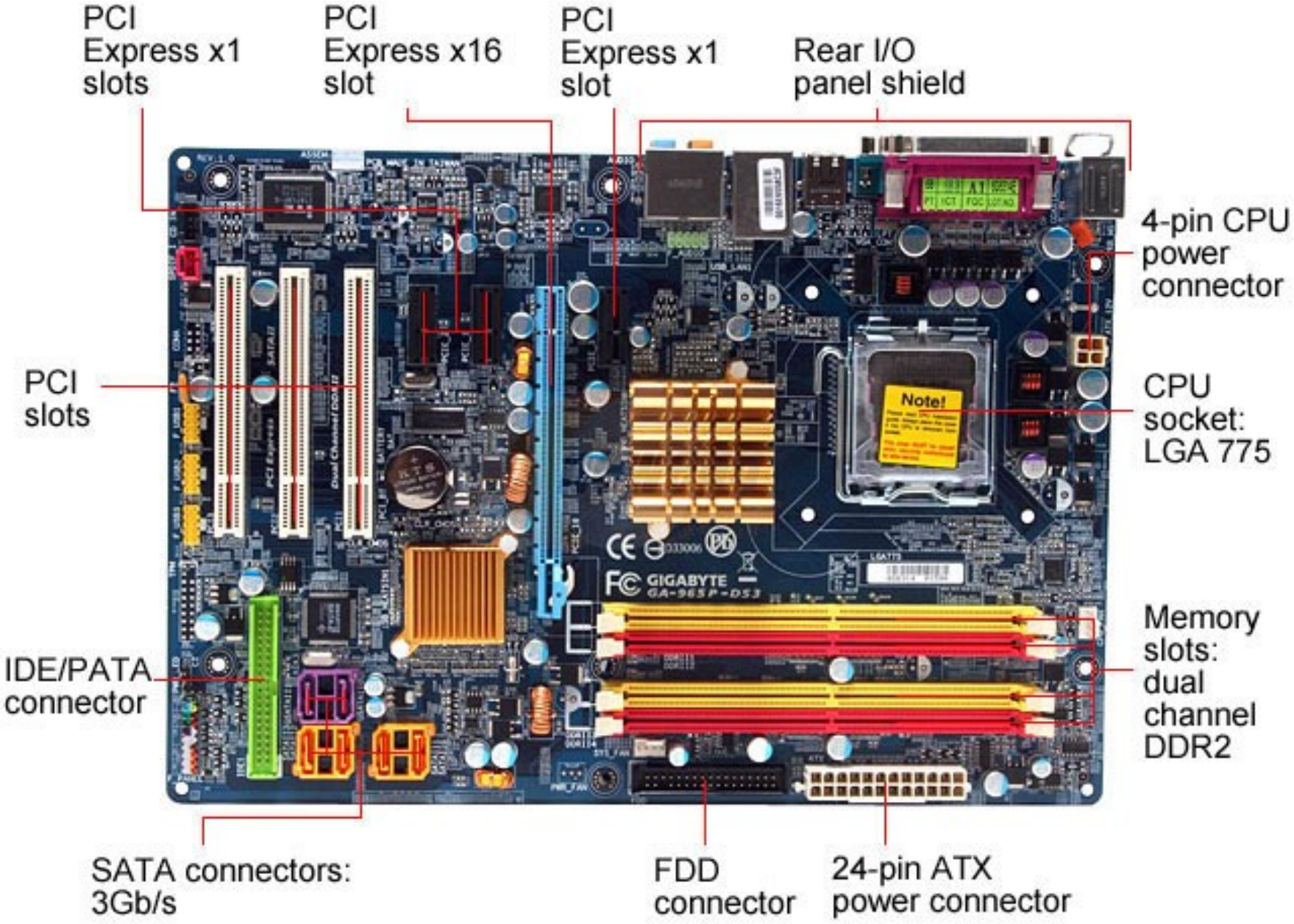

Created
Terminated
Main Memory
Running
Waiting
Blocked
Swapped out and waiting
Swapped out and blocked
Page file / swap space

Memory


CPU
Register
Cache
Level 1
Level 2
RAM
Physical RAM
Virtual Memory
Temporary
Storage
Areas
Storage Device Types
ROM/
BIOS
Removable
Drives
Network/
Internet
Storage
Hard
Drive
Permanent
Storage
Areas
Input Sources
Keyboard
Mouse
Removable
Media
Scanner/
Camera/
Mic/
Video
Remote
Source
Other
Sources



▲ Simplified Computer Memory Hierarchy
Illustration: Ryan J. Leng


PCI Express x1 slots
PCI Express x16 slot
PCI Express x1 slot
Rear I/O panel shield
4-pin CPU power connector
PCI slots
CPU socket: LGA 775
Memory slots: dual channel DDR2
IDE/PATA connector
SATA connectors: 3Gb/s
FDD connector
24-pin ATX power connector
Note!
GIGABYTE
GA-965P-DS3

Praktik SO?

- TIDAK (lagi) membahas konsep Sistem Operasi
- Fokus pada penguasaan Sistem Operasi yang telah ada.
- Di akhir kuliah, diharapkan mahasiswa:
  - Mampu menggunakan berbagai perintah penting pada Shell Linux
  - Mampu membuat program Shell Linux sederhana
  - Mampu menangani proses-proses yang dijalankan oleh Sistem Operasi atas permintaan pengguna.
  - Mampu memanfaatkan Shell Linux secara efektif untuk berbagai keperluan administrasi sistem.

Apa yang dipelajari?

- Pertemuan 1 – Mengenal Linux & Shellnya
  - Tentang Kuliah ini
  - Mengenal Linux & Instalasinya (Tugas Mandiri)
  - [01] Halaman Manual
  - [02] Struktur Direktori
- Pertemuan 2 – Direktori & File
  - [03] Bekerja dengan Direktori
  - [04] Bekerja dengan File
  - [05] Mengakses Isi File
- Pertemuan 3 – Ekspansi Shell I
  - [06] Perintah dan Argumen
  - [07] Operator Kendali
  - [08] Variabel
- Pertemuan 4 – Ekspansi Shell II
  - [09] Riwayat Shell
  - [10] File Globbing/Pembangkitan Nama File Otomatis

Apa yang dipelajari?

- Pertemuan 5 – Redireksi & Pipe
  - [11] Dasar Redireksi & Pipe
  - [12] Filter
  - [13] Latihan Soal 1
- Pertemuan 6 – Tool Dasar & Pemrograman Shell
  - [14] Tool Linux Dasar
  - [15] Dasar Script Shell
  - [16] Perulangan & Seleksi Kondisi
- Pertemuan 7 – Pemrograman Shell Lanjut
  - [17] Opsi dan Parameter
  - [18] Script Shell Lanjutan
  - [19] Latihan Soal 2
- Pertemuan 8 – Manajemen Proses
  - [20] Proses
  - [21] Prioritas
  - [22] Proses Background
  -

Tugas

- Buat kelompok @ 4 orang
- Jelaskan Arsitektur Processor **Intel Core i7**? Apa bedanya dengan i3 dan i5?
- Bagaimana membuat *bootable flashdisk* **Ubuntu 12.04** pada sistem operasi Linux (ada banyak tool, uraikan tahapan detail dari setiap tool yang digunakan)
- Lakukan **instalasi** Linux Ubuntu, Linux Mint, Arch Linux atau Fedora terbaru, *capture* gambarnya dan buat menjadi tutorial “Panduan Instalasi Linux”
- ***Paperless***, kirimkan ke email saya, terakhir Senin, 04 Juni 2012.

Web Site (Referensi)

- Komputasi.wordpress.com
- Repo.ugm.ac.id
- Howtoforge.org
- Linuxhomenetworking.com
- www.yo**linux**.com
- Ilmukomputer.org
- linux-tutorial.info

01 - Halaman Manual

Halaman Manual

- Setiap perintah (*command*) bawaan Linux disertai dengan halaman manual
- Perintah man digunakan untuk menampilkan manual dari suatu perintah
- Bentuk: man perintah
- Contoh: man ls
- Tekan q untuk keluar dari halaman manual

Format Perintah man

- man nama_program
- man file_konfigurasi
- man nama_daemon

- Gunakan man -k atau apropos untuk menampilkan halaman manual yang mengandung string tertentu
- Gunakan whatis untuk mengetahui kegunaan dari suatu perintah

Contoh

- man cat
- man sources.list
- man syslog.conf
- man syslogd
- man mysqld
- man -k mysql
- apropos mysql
- whatis cat

Apa hasilnya?

- man cat ls touch
- whatis cat ls mysql
- apropos mysql df du

Dimanakah Manualnya?

- Gunakan whereis -m perintah
- Contoh:

  whereis -m ifconfig

  ifconfig: /usr/share/man/man8/ifconfig.8.gz

  Halaman manual dapat dibukan langsung:

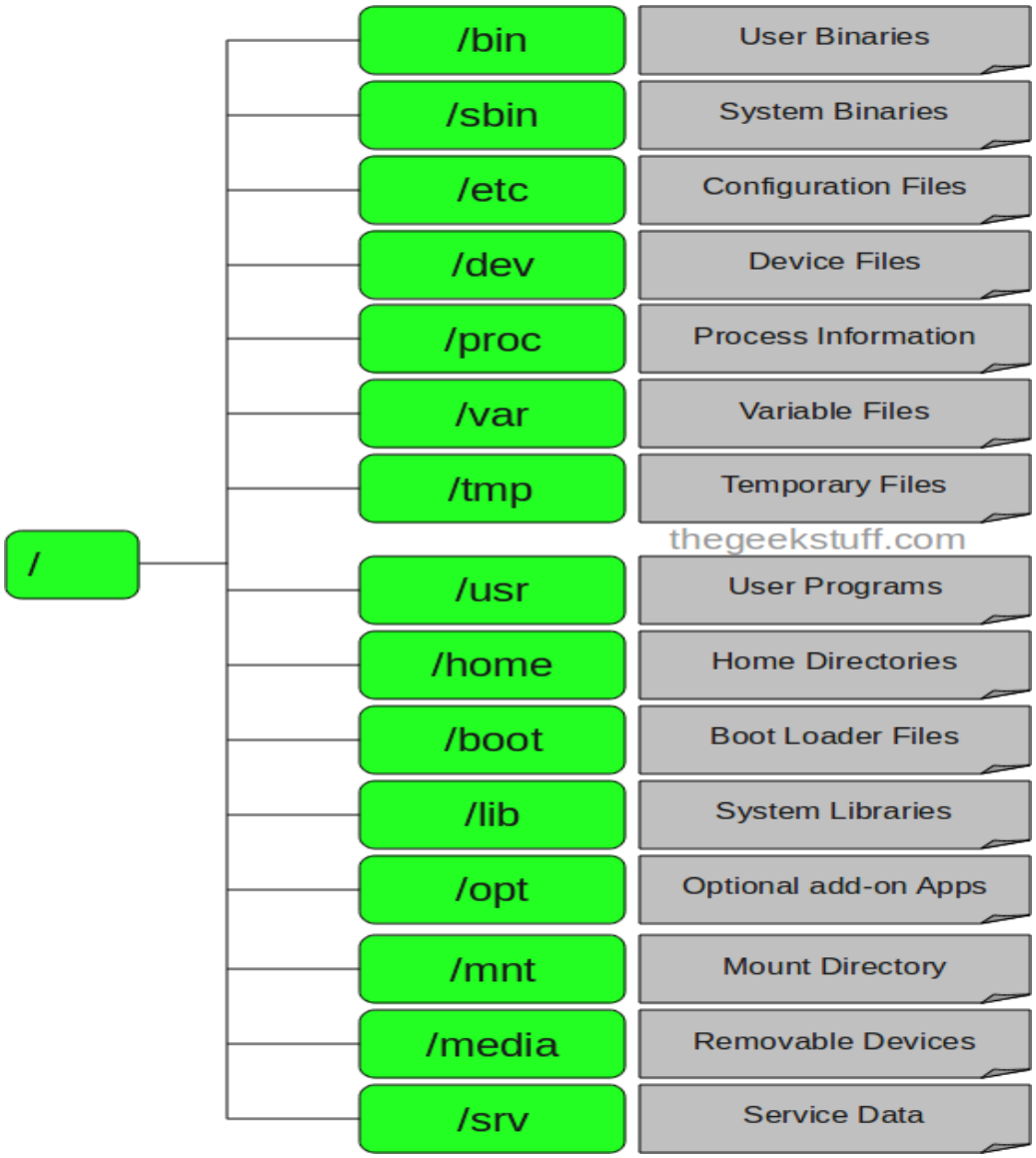
  man /usr/share/man/man8/ifconfig.8.gz

Pertanyaan

- Sebutkan bagian-bagian dari suatu halaman manual!
- Apa hasil dari eksekusi berikut?
  - man passwd
  - man 5 passwd
  - man man
  - man woman
  - mandb

02 - Struktur Direktori

Struktur Direktori



03 - Bekerja dengan Direktori

Bekerja dengan Direktori

- Direktori Aktif
- Pindah Direktori
- Alamat Absolut & Relatif
- Penulisan Alamat Otomatis
- Melihat Isi Direktori
- Membuat Direktori
- Menghapus Direktori

Dimana Anda Berada?

- Gunakan perintah pwd (*print working directory*)
- Contoh:

```
pwd
/home/d3tmj
```

Pindah Direktori

- Gunakan perintah cd (*change directory*)
- Format:

  cd direktori_tujuan
- Contoh:

  cd Documents

  pwd

  /home/d3tmj/Documents

Kembali ke *Home Directory*

- Gunakan perintah

  cd atau

  cd ~

cd .. dan cd -

- Coba pindah ke suatu direktori, misalnya Documents
- Kemudian pindahkan ke direktori /home/nama_pengguna/Downloads
- Jalankan perintah cd ..
- Gunakan perintah pwd
- Jalankan perintah cd -
- Gunakan perintah pwd

Alamat Absolut & Relatif

- Relatif
  - Direktori Downloads dan Documents relatif terhadap direktori /home/nama_pengguna
  - Dapat diakses langsung, misal cd Downloads
- Absolut
  - Alamat absolut dari direktori Downloads adalah /home/nama_pengguna/Downloads
  - Akses ke direktori tersebut harus menggunakan alamat absolut, jika direktori aktif anda tidak sama dengan lokasi direktori Downloads tersebut

Kegunakan Tombol TAB

- Otomatis melengkapkan alamat yang dituliskan oleh pengguna
- Contoh 1
  - Ketik cd /ho
  - Tekan Tab
  - Tekan sesuatu
  - Tekan Tab
- Contoh 2
  - Ketik man cal
  - Tekan Tab
  - Ketik calendar

Menampilkan Isi Direktori

- Gunakan perintah ls
- Variasi:
  - ls
  - ls -a
  - ls -l
  - ls -lh
- Silakan dicoba dan perhatikan apa yang terjadi!

Membuat Direktori

- Gunakan perintah mkdir
- Contoh:

  mkdir kuliah

- Variasi:

  mkdir -p ← sekaligus membuatkan direktori untuk induknya (di atasnya)

- Contoh:

  mkdir -p /kuliah/2012/06

Menghapus Direktori

- Gunakan perintah rmdir
- Contoh
  - rmdir kuliah/2012/06
- Variasi
  - rmdir -p
- Contoh
  - rmdir -p kuliah/2012

Latihan

- Tampilkan direktori aktif anda
- Pindahlah ke direktori /etc
- Pindahlah ke home directory hanya dengan 3 kali tekan keyboard.
- Pindahlah ke direktori /boot/grub (berapa kali tekan keyboard?)
- Pindahlah ke direktori parent (atasnya) direktori anda sekarang.
- Pindahlah ke direktori root (/)
- Tampikan isi direktori root tersebut
- Sekali lagi, tetapi dengan list panjang.
- Tetap ditempat, tampilkan isi direkttori /etc.
- Tampilkan isi direktori /bin dan /sbin sekaligus

Latihan

- Tampilkan isi direktori ~.
- Tampilkan semua file dalam home directory anda
- Tampilkan file dalam /boot dengan format yang lebih readable
- Buat direktori testdir dalam home directory
- Pindah ke direktori /etc. Buat direktori newdir dalam home direktori anda
- Buat 3 direktori bertingkat dalamn home direktory sekaligus, ~/dar/der/dor
- Hapus direktori testdir
- Hapus direktori ~/dar

Pushd dan Popd

- Apa manfaat dari perintah pushd dan popd? Gunakan perintah man bash untuk mengetahuinya!
  - pushd /var/cache/
  - pushd /temp
  - cd -
  - pwd
  - popd
  - cd -
  - pwd

Latihan

- Tampilkan direktori aktif anda
- Pindahlah ke direktori /etc
- Pindahlah ke home directory hanya dengan 3 kali tekan keyboard
- Pindahlah ke direktori /boot/grub (berapa kali tekan keyboard?)
- Pindahlah ke direktori parent (atasnya) direktori anda sekarang
- Pindahlah ke direktori root (/)
- Tampikan isi direktori root tersebut
- Sekali lagi, tetapi dengan list panjang
- Tetap di tempat, tampilkan isi direktori /etc
- Tampilkan isi direktori /bin dan /sbin sekaligus
- Tampilkan isi direktori ~
- Tampilkan semua file dalam home directory anda
- Tampilkan file dalam /boot dengan format yang lebih readable (human)
- Buat direktori testdir dalam home directory
- Pindah ke direktori /etc. Buat direktori newdir di dalam home direktory anda
- Buat 3 direktori bertingkat dalamn home direktory sekaligus, ~/dar/der/dor
- Hapus direktori testdir
- Hapus direktori ~/dar.

04 – Bekerja dengan File

Bekerja dengan File

- Case-sensitive
- Semua dianggap file
- Membuat file
- Menghapus File
- Menyalin File
- Memindahkan file
- Mengganti nama File

Huruf kecil dan Besar BEDA

- test.txt dan Test.txt adalah dua file yang berbeda
- Contoh

```
touch test.txt
touch Test.txt
ls *.txt
semuafile2.txt  semuafile.txt  test.txt  Test.txt
ls -l *.txt
-rw-rw-r-- 1 husni husni 20257638 Aug 17 05:56 semuafile2.txt
-rw-rw-r-- 1 husni husni 20257611 Aug 17 05:56 semuafile.txt
-rw-rw-r-- 1 husni husni        0 Aug 17 15:25 test.txt
-rw-rw-r-- 1 husni husni        0 Aug 17 15:25 Test.txt
```

Perintah file

- Digunakan untuk mengetahui jenis/type file
- Format: file nama_perintah
- Contoh

  file semuafile.txt

  semuafile.txt: ASCII text
- Variasi: file -s ← untuk file special, misalnya /dev/sda, /proc/cpuinfo
- Contoh:

  sudo file -s /dev/sda1

  /dev/sda1: x86 boot sector, code offset 0x52, OEM-ID "NTFS ", sectors/cluster 8, reserved sectors 0, Media descriptor 0xf8, heads 255, hidden sectors 2048, dos < 4.0 BootSector (0x80)

Perintah touch

- Digunakan untuk membuat file (tanpa isi)
- Contoh
  - touch file1.txt
  - touch file2
  - ls -l
- Apa manfaat touch -t. Cek man touch.
- Contoh:

  touch -t 201205050000 filesatu

  touch -t 130207111630 filedua

Menghapus File

- Gunakan perintah rm
- Contoh

  rm test.txt

  ls -l *est.txt

  -rw-rw-r-- 1 husni husni 0 Aug 17 15:25 Test.txt
- Variasi:
  - rm -i ← ada konfirmasi
  - rm -rf ← rekursif dan force (paksakan)
  - Contoh

    rm -i Test.txt

    rm: remove regular empty file `Test.txt'? n

Menyalin File

- Gunakan perintah cp
- Contoh

  cp Test.txt Test123.txt

- Variasi:
  - cp -r
  - Menyalin banyak file (rekursif) ke satu direktori
  - cp -i
  - cp -p

Perintah mv

- Digunakan untuk memindahkan file dari satu lokasi ke lokasi lain
- Juga dapat digunakan untuk mengganti nama file
- Ada juga perintah rename. Bagaimana cara menggunakannya?

Latihan

- Tampilkan semua file dalam direktori /bin
- Tampilkan jenis/tipe dari file /bin/cat, /etc/passwd dan /usr/bin/passwrd
- Download suatu file dari Google Image, namakan sebagai gambar.jpg. Tampilkan tipe file tersebut.
- Ganti nama file gambar.jpg menjadi gambar.pdf. Tampilkan informasi tipenya!
- Buat direktori ~/dirbaru dan masuklah ke direktori tersebut
- Buat file sekarang.txt dan kemarin.txt dalam direktori dirbaru tersebut
- Ubah tanggal pada kemarin.txt sesuai tanggal hari kemarin
- Salin kemarin.txt ke salinan.kemarin.txt
- Ganti nama salinan.kemarin.txt menjadi nama_anda
- Buat suatu direktori bernama ~/testbackup dan salin semua file dari ~/dirbaru ke dalamnya
- Gunakan satu perintah untuk menghapus direktori ~/testbackup dan semua file di dalamnya
- Buat suatu direktori ~/etc/backup dan salin semua file .conf dari /etc ke dalamnya.
- Gunakan perintah rename untuk mengganti nama semua file .conf menjadi .backup!

05 - Mengakses Isi File

Mengakses Isi File

- Head
- Tail
- Cat
- Tac
- More, Less
- Strings

Perintah head

- Defaultnya menampilkan 10 baris pertama dari suatu file
- Contoh

  head /etc/passwd
- Variasi:
  - head -n nama_file ← n baris pertama
  - head -cn nama_file ← n *byte* (karakter) pertama

Perintah tail

- Menampilkan bagian akhir dari file
- Contoh:

  tail /etc/passwd

Perintah cat (1)

- Menampilkan isi file ke suatu output
- Menggabungkan isi file
- Membuat file baru
- Contoh

  cat > filesatu.txt

  Ketik apa saja

  Tekan Ctrl D
- Menambahkan isi file

  cat >> filesatu.txt

Perintah cat (2)

- Membuat Tanda akhir File
  - cat > filesatu.txt <<stop
  - Ketik apa saja
  - stop
- Menyalin File
  - cat filesatu.txt > filedua.txt

Perintah tac?

- Apa kegunaan perintah tersebut?
- Coba !!!

  cat > fileku

  Satu

  Dua

  Tiga

  Empat

  Ctrl D

  tac fileku

Perintah more dan less

- Menampilkan isi suatu file per halaman
- Tekan spasi untuk menuju halaman berikutnya
- q untuk selesai

Latihan

- Tampilkan 13 baris pertama dari file /etc/services
- Tampilkan baris terakhir dari file /etc/passwd
- Gunakan perintah cat untuk membuat file bernama hitung.txt yang berisi sebagai berikut:

  Satu

  Dua

  Tiga

  Empat

  Lima
- Gunakan perintah cp untuk membuat backup (salinan) dari file ini ke cp_hitung.txt
- Gunakan perintah cat untuk membuat salinan dari file ini ke cat_hitung.txt
- Tampilkan isi cat_hitung.txt, tetapi dengan semua baris dalam urutan terbalik (baris terakhir menjadi baris pertama)

Latihan

- Gunakan perintah more untuk menampilkan /var/log/messages
- Tampilkan string karakter readable dari perintah /usr/bin/passwd
- Gunakan perintah ls untuk mendapatkan file paling besar dalam /etc
- Buka dua terminal (console). Pastikan anda berada pada direktori yang sama.Pada terminal pertama, ketik echo Ini adalah baris pertama > tailing.txt. Pada terminal kedua, jalankan tail -f tailing.txt . Pada terminal pertama, ketik echo Ini baris lainnya >> tailing.txt. Pada terminal kedua. Apa yang diperoleh pada terminal kedua? Hentikan tail -f dengan Ctrl C.
- Gunakan perintah cat untuk membuat file bernama tailing.txt yang mengandung isi dari tailing.txt diikuti dengan isi dari /etc/passwd.
- Gunakan perintah cat untuk membuat file bernama tailing.txt yang mengandung isi dari tailing.txt didahului oleh isi dari /ect/passwd.

06 - Perintah & Argumen

Perintah dan Argumen

- Perintah echo
- Argumen-argumen
- Perintah-perintah
- Alias

Perintah echo

- Menampilkan input yang diterimanya
- Contoh

  echo Pra-S2

  Pra-S2

  echo Master of Computer Science UGM Yogyakarta

  Master of Computer Science UGM Yogyakarta

Tahapan Eksekusi Perintah

- Scanning → Shell Linux melakukan scan terhadap perintah yang dimasukkan

  (semua input, baik perintah atau bukan, dianggap sebagai argumen)

- Modification → Shell melakukan penyesuaian atau perubahan terhadap argumen

  Proses ini dinamakan Shell Expansion

- Execution → Perintah dijalankan

Hilangnya *White Space*

- Bagian-bagian yang dipisahkan oleh satu atau lebih *white space* (atau tab) dianggap sebagai argumen
- Setiap *white space* dihapus, tidak termasuk argumen
- Contoh: perintah **echo** mencetak setiap argumen yang diterima dan dipisahkan dengan satu spasi

  echo Praktikum Sistem Operasi

  Praktikum Sistem Operasi

  echo Praktikum Sistem Operasi

  Praktikum Sistem Operasi

  echo Praktikum Sistem Operasi

  Praktikum Sistem Operasi

Tanda Petik Tunggal & Ganda

- Apa yang ada diantara dua tanda petik dianggap sebagai satu argumen. Tidak ada white space dalam tanda petik yang dihilangkan
- Contoh

  echo 'Praktikum Sistem Operasi'

  Praktikum Sistem Operasi

  echo "Praktikum Sistem Operasi"

  Praktikum Sistem Operasi
- Apa bedanya? :-)

Echo dan Tanda Petik

- Perintah echo -e dapat mengenali karakter khusus yang ada di antara 2 tanda petik
- Karakter khusus tersebut, misalnya \n (ganti baris) dan \t (tab, biasanya 8 spasi)
- Contoh
- echo -e 'Sebentar lagi akan ada \nbaris baru'

  Sebentar lagi akan ada

  baris baru
- echo -e "Sebentar lagi akan ada \nbaris baru"

  Sebentar lagi akan ada

  baris baru
- echo -e "Sebentar lagi ada \ttabulasi di sini"

  Sebentar lagi ada tabulasi di sini
- echo -e 'Sebentar lagi ada \ttabulasi di sini'

  Sebentar lagi ada tabulasi di sini

Perintah

- Ada 2 jenis perintah: built-in & Eksternal
- built-in: bawaan sistem operasi dan merupakan bagian dari shell.
- Eksternal: program yang mempunyai binernya sendiri, biasanya diletakkan dalam /bin atau /sbin
- Perintah type dapat digunakan untuk memeriksi jenis dari file (built in & eksternal)

  Contoh

  type cd

  cd is a shell builtin

  type more

  more is /bin/more

  type ls

  ls is aliased to `ls --color=auto'

-

Menjalankan Program Eksternal

- Beberapa program mempunyai versi built-n dan eksternalnya. Built-in mempunyai prioritas lebih tinggi
- Eksekusi program ekstenal harus menyebutkan lokasi file program tersimpan
- Contoh

  type -a echo

  echo is a shell builtin

  echo is /bin/echo

  /bin/echo "Praktikum SO telah dimulai"

  Praktikum SO telah dimulai

Perintah which

- Digunakan untuk mencari program (eksternal) dalam variabel lingkungan $PATH
- Contoh: (cd dan type perintah built-in)

  which cp mv cd ls cat echo man touch more type

  /bin/cp

  /bin/mv

  /bin/ls

  /bin/cat

  /bin/echo

  /usr/bin/man

  /usr/bin/touch

  /bin/more

Alias = Nama Lain

- Dibuat menggunakan perintah **alias**
- Contoh

  cat > hitung.txt

  satu

  dua

  tiga

  empat

  alias mundur=tac

  mundur hitung.txt

  empat

  tiga

  dua

  satu

Menyingkatkan Perintah

- Perintah yang panjang, biasanya banyak parameter dapat disingkatkan dengan alias
- Contoh

  alias ll='ls -lh --color=auto'

  ll

  alias c=clear

  alias x=exit

Mengatur Default Program

- Suatu program dapat dibuat berjalan dengan parameter default
- Misal: perintah rm dibuat agar defaultnya rm -i

```
touch file.txt
alias rm='rm -i'
rm file.txt
rm: remove regular empty file `file.txt'? y
ll file.txt
ls: cannot access file.txt: No such file or directory
```

Melihat & Menghapus Alias

- Perintah alias tanpa argumen menampilkan semua alias yang ada (aktif)
- Perintah alias diikuti argumen hanya menampilkan alias yang bernama argumen

  alias ls la rm

  alias ls='ls --color=auto'

  alias la='ls -A'

  alias rm='rm -i'

- Perintah unalias digunakan untuk menghapus alias

Latihan

- Berapa jumlah argumen pada baris ini (selain perintah)?

  touch '/etc/apt/source.list' 'linux ubuntu 12.04' "repo.ugm.ac.id"
- Apakah tac merupakan perintah shell built in?
- Apakah ada alias untuk perintah rm
- Buat dan hapus file menggunakan perintah rm dengan opsi -i
- Buat alias bernama rm untuk rm -i. Uji dengan suatu file. Berhasil?
- Tampilkan semua alias yang ada
- Buat alias bernama kota untuk menampilkan kota asal anda. Uji. Berhasil?
- Gunakan perintah set -x untuk menampilkan ekspansi shell untuk setiap perintah
- Uji fungsi set -x dengan menjalankan alias kota dan rm
- Eksekusi set +x untuk menghentikan tampilkan ekspansi shell

Latihan

- Hapus alias kota
- Dimana lokasi perintah cat dan passwd?
- Jelaskan perbedaan perintah berikut:

  echo

  /bin/echo
- Jelaskan perbedaan perintah berikut:

  echo Hello

  echo -n Hello
- Tampilkan A B C dengan dua spasi antara B dan C
- Bagaimana menampilkan output berikut tanpa menggunakan spasi?

  4+4 = 8

  10+14 = 24
- Gunakan echo untuk menampilkan:

  ??\\
- Gunakan satu perintah echo untuk menampilkan tiga kata pada 3 baris!

07 - OPerator Kendali

Operator Kendali

- ; → semicolon
- & → ampersand
- $? → dolar tanda tanya
- && → ampersand ganda
- || → bar vertikal ganda
- Kombinasi && dan ||
- # → tanda pound
- \ → karakter khusus escape

; Titik-koma

- Lebih satu perintah pada satu baris dapat dipisahkan dengan titik-koma (;)
- Eksekusi terhadap perintah-perintah dilakukan secara urut
- Contoh:

  echo Praktikum;echo Sistem Operasi; file hitung.txt

  Praktikum

  Sistem Operasi

  hitung.txt: ASCII text

& Ampersand

- Digunakan untuk segera mendapatkan shell meskipun eksekusi belum selesai. Perintah akan diseksekusi di Background
- Contoh: menyalin banyak (lama waktunya) file di background dan shell digunakan untuk pekerjaan lain yang singkat-singkat.

$? Dolar Tanda Tanya

- Parameter dari Shell
- Memegang status exit dari perintah yang dieksekusi sebelumnya
- Nilai 0 menunjukkan bahwa eksekusi berhasil dengan baik
- Contoh

  touch satu.txt; rm satu.txt

  rm: remove regular empty file `satu.txt'? y

  echo $?

  0

  rm satu.txt

  rm: cannot remove `satu.txt': No such file or directory

  echo $?

  1

&& Ampersand Ganda

- && dianggap sebagai logical AND. Digunakan diantara dua perintah.
- Perintah kedua dieksekusi hanya jika eksekusi perintah pertama sukses (status exit 0)
- Contoh

```
echo satu && echo dua

echor satu && echo dua   ← error

cd Downloads/ && ls

cd Downloads/ && ls      ← error
```

|| Bar Vertikal Ganda

- || dianggap sebagai logical OR.
- Perintah kedua dieksekusi hanya jika perintah pertama gagal (status exit tidak 0)
- Contoh

  echo satu || echo dua; echo tiga

  echor satu || echo dua; echo tiga ← error

  cd Downloads/ || ls

  cd Downloads/ || ls ← error

Kombinasi && dan ||

- Digunakan untuk membentuk struktur if-then-else
- Contoh: Jika eksekusi pertama (rm) berhasil maka jalankan perintah kedua (echo), jika gagal jalankan perintah ketiga (echo)

  touch filesatu.txt

  rm filesatu.txt && echo Delete Berhasil || echo Delete Gagal

  rm: remove regular empty file `filesatu.txt'? y

  Delete Berhasil

  rm filesatu.txt && echo Delete Berhasil || echo Delete Gagal

  rm: cannot remove `filesatu.txt': No such file or directory

  Delete Gagal

Tanda Pound

- Apapun yang ditulis setelah tanda # diabaikan oleh Shell, dianggap sebagai komentar
- Contoh

```
mkdir dirsatu #membuat direktori dirsatu
cd dirsatu    #masuk ke direktori dirsatu
```

\ Karakter Escape

- Digunakan untuk memasukkan karakter khusus yang tidak diterjemahkan oleh Shell,misalnya tanda ; & $ #
- Contoh

  echo titik \; koma

  echo \& \$ \# @

  echo escaping \\\?*\"\'
- Apa perbedaan dua perintah ini?

  echo *

  echo *

\ pada akhir baris

- Jika diletakkan di akhir baris, Shell akan memberikan baris baru tetapi \ tidak diproses saat eksekusi
- Contoh

```
echo This command line \
is split in three \
parts
```

This command line is split in three parts

Latihan

- Saat mengetikkan passwd, file apa yang dieksekusi?
- Jenis file apakah itu?
- Jalankan perintah pwd dua kali
- Jalankan ls setelah cd /etc, tetapi hanya jika cd /etc tidak error
- Jalankan cd /etc setelah cd etc, hanya jika cd etc gagal
- Tampilkan pesan “Berhasil” ketika touch file123 berhasil. Jika tidak tampilkan “Gagal”. Coba ini dalam home directory dan /bin
- Jalankan perintah sleep 6, apa yang dilakukan perintah ini?
- Jalankan sleep 200 pada background
- Tulis perintah untuk menampilkan “Berhasil” jika penghapusan file123 berhasil. Jika tidak tampilkan “Gagal”
- Gunakan echo untuk menampilkan "Ini teks dengan beberapa karakter aneh \ * [} ' ~ ` \\ ." (termasuk semua tanda petik)

08 - Variabel

Variabel

- Mengakses Variabel
- Variabel $PS1 dan $PATH
- Membuat Variabel
- Tanda Petik Ganda dan Tunggal
- Perintah set dan unset

Mengakses Variabel

- Nama variabel didahului oleh tanda $
- Bersifat *case-sensitive*
- Contoh (variabel lingkungan Linux)

  echo $HOSTNAME

  echo Komputer $HOSTNAME digunakan oleh $USER dengan home directory $HOME

Variabel $PS1

- Digunakan untuk menentukan tampilan prompt Shell
- Karakter escape \u mewakili username, \w mewakili working directory
- Contoh

  PS1=prompt

  PS1='Prompt >'

  PS1='\u@\h$ '

- **Homework** (harus login sebagai root)

  Bagaimana agar username untuk pengguna biasa berwarna **HIJAU**, sedangkan untuk root berwarna **MERAH**?

Variabel $PATH

- Menunjukkan lokasi dimana file-file program yang dapat dieksekusi berada
- Contoh

  echo $PATH

- Menambah PATH

  PATH=$PATH:. ← . mewakili directory aktif

Membuat Variabel

- Membuat variabel tidak melibatkan $
- Mengakses variabel harus didahului $
- Contoh

```
var1="Test 1 2 3"
echo $var1
```

Test 1 2 3

Tanda Petik

- Perhatikan perbedaan yang diberikan oleh petik tunggal (') dan ganda (“)!

```
saldo=1250000
echo $saldo
echo "$saldo"
echo '$saldo'
echo "Saldo bulan ini: Rp. $saldo"
echo 'Saldo bulan ini: Rp. $saldo'
```

Perintah set dan unset

- Perintah set menampilkan daftar variabel aktif
- Perintah unset digunakan untuk meniadakan suatu variabel
- Contoh

```
set | more
var1=123
echo $var1
unset var1
echo $var1
```

Latihan

- Apa beda dua baris berikut?

  echo `var1=5;echo $var1`

  echo 'var1=5;echo $var1'

- Apa hasil eksekusi baris berikut?

  echo `cd /etc; ls -d * | grep pass`

- Apa manfaat dari perintah set -u?

  set -u; echo $Myvar

  set +u; echo $Myvar

Latihan

- Apakah baris berikut mencetak Halo Superman dan Supergirl? Jika TIDAK, betulkan!

  prefix=Super; echo Halo $prefixman dan $prefixgirl
- Apa perbedaan antara perintah env dan set?
- Apa manfaat perintah export?
- Tampilkan username yang anda gunakan!
- Salin username tersebut ke variabel $pengguna
- Buat dua variabel bernilai 'Dumb' dan 'do'. Gunakan echo untuk mencetak teks 'Dumbledore'
- Tambahkan waktu (*time*) ke Prompt PS1

09 - Riwayat Shell

Riwayat Shell

- Mengulang Perintah
- Sejarah
- !n
- Ctrl r
- $HISTSIZE, $HISTFILE, $HISTFILESIZE
- Ekspresi Regular

Pengulang Perintah Terakhir

- Perintah Terakhir, gunakan !! (baca: bang-bang)
- Contoh

  tail -2 /etc/passwd

  husni:x:1000:1000:husni,,,:/home/husni:/bin/bash

  mysql:x:115:126:MySQL Server,,,:/nonexistent:/bin/false

  !!

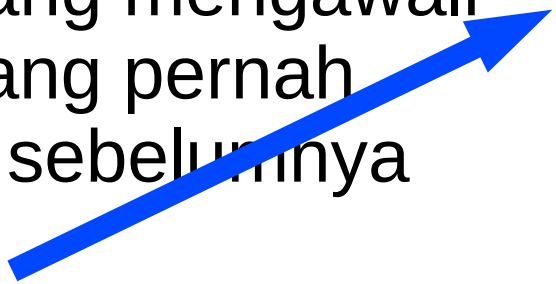
  tail -2 /etc/passwd

  husni:x:1000:1000:husni,,,:/home/husni:/bin/bash

  mysql:x:115:126:MySQL Server,,,:/nonexistent:/bin/false

Mengulang Perintah Lainnya

- Gunakan satu bang diikuti satu atau lebih karakter yang mengawali perintah yang pernah dijalankan sebelumnya
- Contoh



```
echo ini baris pertama > test.txt
cat test.txt
ini baris pertama
echo ini baris kedua >> test.txt
!c
cat test.txt
ini baris pertama
ini baris kedua
echo ini baris ketiga >> test.txt
!c
cat test.txt
ini baris pertama
ini baris kedua
ini baris ketiga
```

Melihat Isi Sejarah (*History*)

- Perintah history digunakan untuk melihat semua perintah yang pernah dijalankan
- Perintah history n digunakan untuk melihat n perintah terakhir yang dijalankan
- Contoh

history 10

```
217  echo ini baris kedua >> test.txt
218  cat test.txt
219  touch test.txt
220  cat test.txt
221  echo ini baris ketiga >> test.txt
222  cat test.txt
223  echo ini baris ketiga > test2.txt
224  cat test2.txt
225  history
```

!n

- !n digunakan untuk memerintahkan Shell menampilkan history nomor n dan kemudian mengeksekusinya
- Contoh

  !222

  cat test.txt

  ini baris pertama

  ini baris kedua

  ini baris ketiga

Ctrl r

- Kombinasi Tombol ini digunakan untuk mencari perintah tertentu yang telah ada di dalam history.
- Contoh: Mencari perintah berbunyi 'get' yang telah dijalankan sebelumnya

  Tekan Ctrl r dan masukkan kata kunci get

  (reverse-i-search)`get': sudo apt-get install python-orange

$HISTSIZE

- Variabel ini berisi jumlah perintah yang dapat disimpan oleh history. Beberapa distro Linux memberikan default 500 atau 1000.
- Variabel ini juga digunakan untuk mengubah jumlah tersebut
- Contoh

  echo $HISTSIZE

  1000

  HISTSIZE=12350

  !e

  echo $HISTSIZE

  12350

  HISTSIZE=1000

  echo $HISTSIZE

  1000

$HISTFILE

- Variabel ini menunjukkan file yang menyimpan history. Shell bash menyimpan history dalam file ~/.bash_history
- Contoh

  echo $HISTFILE

  /home/husni/.bash_history
- History akan disimpan ke file tersebut setelah keluar sesi dengan exit
- Jika keluar terminal gnome dengan mouse atau reboot, history tidak disimpan ke dalam file tersebut

$HISTFILESIZE

- Variabel ini menunjukkan jumlah perintah yang dapat disimpan dalam file history. Beberapa distro memberikan nilai default 2000
- Contoh

  echo $HISTFILESIZE

  2000

Ekspresi Reguler

- Ekspresi reguler (RegEx)dapat digunakan bersama dengan bang (!)
- Contoh: Mengganti 2 dengan 7 memanfaatkan Regex

```
echo Test 1 2 3 > test.txt
cat test.txt
Test 1 2 3
!e:s/2/7
echo Test 1 7 3 > test.txt
!c
cat test.txt
Test 1 7 3
```

```
touch file1.txt; touch file2.txt
echo Haloooo >> file1.txt
cat file1.txt
Haloooo
cat file2.txt

!e:s/1/2
echo Haloooo >> file2.txt
!c
cat file2.txt
Haloooo
```

Korn Shell (ksh)

- Perintah history digunakan untuk menampilkan history

  history 10 → menampilkan history mulai nomor 10

- Huruf r digunakan untuk mengerjakan history tertentu

  r e → mengeksekusi perintah terakhir dalam history yang berawalan e

  r → mengeksekusi perintah terakhir dalam history

Latihan

1. Jalankan perintah echo The answer to the meaning of life, the universe and everything is 42.
2. Ulangi perintah sebelumnya menggunakan hanya dua karakter (ada 2 cara!)
3. Tampilkan lima perintah terakhir yang ada dalam history.
4. Hasilkan seperti pertanyaan pertama tetapi memanfaatkan nomor baris yang dihasilkan pertanyaan ketiga
5. Berapa banyak perintah yang dapat berada di memory untuk sesi shell sekarang?
6. Dimana perintah-perintah ini disimpan ketika keluar Shell?
7. Berapa banyak perintah yang dapat dituliskan ke file history saat anda keluar dari sesi Shell sekarang?
8. Pastikan Shell bash anda nanti akan mampu mengingat 5000 perintah yang dijalankannya
9. Bukan *console* baru (Ctrl Shift t) dengan akun yang sama. Kapan history perintah dituliskan ke file history?

10 - Pembangkitan Nama File (File Globbing)

Pembangkitan Nama File

- * → Asterisk
- ? → Tanda Tanya
- [] → kurung siku
- Range a-z dan 0-9
- $LANG dan kurung siku
- Pencegahan
- Latihan

* Asterisk

- * dianggap cocok dengan rangkaian (satu atau lebih) karakter. Digunakan untuk membangkitkan nama file.
- Jika tidak diberikan *path*, maka dianggap direktori aktif
- Manual: man glob(7)
- Contoh

  ls te*

  test2.txt test.txt

  ls *.txt

  file1.txt file2.txt hitung.txt test2.txt test.txt

  ls *e*.txt

  file1.txt file2.txt test2.txt test.txt

? Tanda Tanya

- Digunakan untuk membangkitkan nama file. ? dianggap tepat satu karakter.
- Contoh

  ls test*

  test2.txt test3.txt test6.odp test.txt

  test3.docx test5.doc test_masuk.docx

  ls test?.*

  test2.txt test3.docx test3.txt test5.doc test6.odp

  ls test?.doc

  test5.doc

  ls test?.doc*

  test3.docx test5.doc

  ls tes??.txt

  test2.txt test3.txt

[] Kurung Siku

- Mencocokkan apapun yang ada dalam kurung siku tanpa melihat urutannya. Digunakan untuk membangkitkan nama file.
- Pemberian tanda ! sebelum karakter menunjukkan NEGASI.
- Contoh

  test[35]*

  test3.docx test3.txt test5.doc

  ls test[3]*

  test3.docx test3.txt

  ls test[56]*[d]*

  test5.doc test6.odp

  ls test*[dp]*

  test3.docx test5.doc test6.odp test_masuk.docx

  ls test[!5]*

  test2.txt test3.docx test3.txt test6.odp test_masuk.docx test.txt

Range a-z dan 0-9

- Range karakter dalam [] dapat digunakan
- Contoh

```
ls test[0-9]*
test2.txt  test3.docx  test3.txt  test5.doc  test6.odp
ls te[a-z]*
terserah.docx  test3.docx  test5.doc  test_masuk.docx
test2.txt      test3.txt   test6.odp  test.txt
ls te[a-z][a-z][!0-9]*
terserah.docx  test_masuk.docx  test.txt
ls test[0-9]?[d]*
test3.docx  test5.doc
ls test[0-9]?[!d]*
test2.txt  test3.txt  test6.odp
```

$LANG dan Kurung Siku

- Beberapa bahasa ($LANG) tidak *case-sensitive*, misal en_US.UTF-8

```
echo $LANG
en_US.UTF-8
ls [a-z]est[0-9]*
test2.txt   test3.txt  test6.odp  Test7.txt
test3.docx  test5.doc  Test7.odp  Test8.doc
ls [A-Z]est[0-9]*
test2.txt   test3.txt  test6.odp  Test7.txt
test3.docx  test5.doc  Test7.odp  Test8.doc
LANG=C
echo $LANG
C
ls [A-Z]est[0-9]*
Test7.odp  Test7.txt  Test8.doc
```

Pencegahan...

- Apa yang dihasilkan baris berikut?

  echo *

  echo '*'

  echo *

  echo “*”

  mkdir dir1; cd dir1; echo *

  touch file1.txt; touch file2.txt

  echo *

- Apa yang dilakukan untuk mencegah pembuatan nama file secara otomatis?

Latihan

- Buatlah direktori bernama TestDir dan masuk ke dalamnya.
- Buat file bernama berkas1, berkas10, berkas11, berkas2, Berkas3, Berkas3, berkasAB, berkasa, berkasAAA, berkas(, berkas 2
- Tampilkan semua file berawalan berkas
- Tampilkan semua file berawalan Berkas
- Tampilkan semua file yang dimulai dengan berkas dan diakhiri bilangan
- Tampilkan semua file yang dimulai dengan berkas dan diakhiri huruf
- Tampilkan semua file yang diawali dengan Berkas dan membuat satu digit sebagai karakter kelima
- Tampilkan semua file yang dimulai dengan Berkas dan mempunyai suatu digit sebagai karakter kelima dan tidak lain ada lainnya
- Tampilkan semua file yang dimulai huruf dan diakhir bilangan

Latihan

- Tampilkan semua file yang panjangnya tepat tujuh karakter.
- Tampilkan semua file yang berawalan B atau b fdan berakhiran 3 atau A
- Tampilkan semua file yang berawalan dengan b yang mempunyai e atau R sebagai karakter kedua dan diakhiri bilangan
- Tampilkan semua file yang tidak dimulai huruf F
- Salin nilai dari variabel $LANG ke $MyLANG.
- Perlihatkan perngaruh dari $LANG dalam menampilkan range A-Z atau a-z.
- Jika server anda dihack seseorang dan menghilangkan perintah ls. Apakah anda menggunakan perintah lain yang berfungsi serupa ls? Bagaimana?
- Adakah perintah selain cd yang dapat digunakan untuk berpindah direktori?

11 - Redireksi & Pipe

Redireksi dan Pipe

- stdin, stdout dan stderr
- Redireksi Output
- Redireksi Error
- Redireksi Input
- Redireksi yang Membingungkan
- Pembersihan File Cepat
- swapping stdout dan stderr
- Pipe
- Latihan

stdint, stdout dan stderr

- Shell mengambil input dari stdin (stream 0) dan mengirimkan output ke stdout (stream 1) dan pesan error ke stderr (stream 2)
- Keyboard berfungsi sebagai stdin. Display (monitor) berfungsi sebagai stdout dan stderr

Redireksi Output

- Stdout
- Standarnya adalah Display. Diredirect dengan tanda lebih besar > atau 1>
- Contoh

  echo Belajar Shell Linux

  Belajar Shell Linux

  echo Belajar Shell Linux 1> belajar.txt

  cat belajar.txt

  Belajar Shell Linux

Akibat dari > atau 1>

- Isi file yang dijadikan tujuan dibersihkan, walaupun redirect GAGAL

  echp Belajar Shell Linux 1> belajar.txt

  cat belajar.txt

- Dapat dicegah dengan opsi noclobber

  set -o noclobber

  echo Belajar Shell Linux 1> belajar.txt

  bash: belajar.txt: cannot overwrite existing file

  set +o noclobber

  echp Belajar Shell Linux 1> belajar.txt

  cat belajar.txt

Menembus noclobber

- Opsi set -o noclobber dapat ditembus dengan *redirect* berbentuk >|
- Contoh

  set -o noclobber

  echo Belajar Shell Linux - Pertemuan 04 1> belajar.txt

  bash: belajar.txt: cannot overwrite existing file

  echo Belajar Shell Linux - Pertemuan 04 1>| belajar.txt

  cat belajar.txt

  Belajar Shell Linux - Pertemuan 04

Beda antara > dan >>

- > digunakan untuk membuat file baru dengan isi baru
- >> digunakan untuk menambahkan isi baru ke dalam file yang telah ada
- Contoh

  echo ini membuat file > file101.txt

  cat file101.txt

  ini membuat file

  echo ini baris baru dalam file101 >> file101.txt

  cat file101.txt

  ini membuat file

  ini baris baru dalam file101

  echo ini baris terakhir dalam file101 >> file101.txt

  cat file101.txt

  ini membuat file

  ini baris baru dalam file101

  ini baris terakhir dalam file101

Redireksi Error

- 2> dapat digunakan untuk mengalihkan pesan error ke suatu file dan menghilangkan tampilan tersebut di layar
- Contoh

  find / > semuafile.txt 2> /dev/null

  cat semuafile.txt
- 2>&1 digunakan untuk me-redirect stdout ke stderr ke file yang sama
- Contoh

  find / > semuafile_plus_error.txt 2>&1

Redireksi Input

- Paling simpel adalah menggunakan < atau 0<
- Contoh
- cat < file101.txt
- ini membuat file
- ini baris baru dalam file101
- ini baris terakhir dalam file101
-

<< dan <<<

- << dapat digunakan untuk meredirect input sampai ditemukan deretan khusus, misalnya EOF

  cat <<EOF > file102.txt

  > satu

  > EOF

  cat file102.txt

  cat <<stop > file103.txt

  > satu nol

  > tiga

  > stop

- <<< digunakan untuk meredirect string langsung ke suatu perintah

  base64 <<< husni@mail.ugm.ac.id

  aHVzbmlAbWFpbC51Z20uYWMuaWQK

  base64 -d <<< aHVzbmlAbWFpbC51Z20uYWMuaWQK

  husni@mail.ugm.ac.id

Kejelasan Redireksi

- Apa maksud 3 baris berikut?

  cat winter.txt > snow.txt 2> errors.txt

  2> errors.txt cat winter.txt > snow.txt

  < winter.txt > snow.txt 2> errors.txt cat

Pipe

- Pipe (|) mengambil stdout dari perintah sebelumnya dan mengirimnya sebagai stdin untuk perintah berikutnya
- Contoh

  ls /etc > etcfiles.txt

  tail -4 etcfiles.txt
- Dapat disingkat menjadi

  ls /etc | tail -4
- Bar vertikal atau pipe diletakkan diantara dua perintah yang akan dieksekusi secara simultan
- ls /etc | tail -4 | tac

Latihan

- Gunakan ls untuk meng-output-kan isi dari direktori /etc/ ke file etc.txt.
- Aktifkan opsi Shell noclobber
- Pastikan nocclobber aktif dengan mengulang ls terhadap /etc/.
- Non-aktifkan opsi noclobber
- Buka dua console atau shell sekaligus. Buat file kosong bernama tailing.txt. Ketik tail -f tailing.txt. Gunakan Shell kedua untuk menambahkan sebaris teks ke file tersebut. Pada Shell pertama, periksa hasilnya.
- Buat file yang mengandung nama lima orang. Gunakan cat dan redireksi output untuk membuat file tersebut dan gunakan << untuk menandai akhir input

12 - Filter

Filter

- Perintah-perintah yang digunakan bersama dengan pipe sering dinamakan filter

cat

tee

grep

cut

tr

wc

sort

uniq

comm

od

sed

cat dan tee

- Tidak ada yang dilakukan oleh cat di antara dua pipe.

  tac count.txt | cat | cat
- Perintah tee digunakan untuk mendapatkan hasil antara pada banyak pipe. Hasilnya sama dengan perintah cat
- Contoh

  cat file201.txt | tee temp201.txt | tac

  Satu

  Nol

  Dua

  cat temp201.txt

  Dua

  Nol

  Satu

grep

- grep digunakan untuk menyaring baris yang cocok dengan string tertentu

  cat > file205.txt

  2. Agus Mustofa, Medan, Sumatera Utara

  3. Husni Ilyas, Slemen, Yogyakarta

  4. Raden Mas Azzam Altaf, Trenggalek, Jawa Timur

  5. Siti Malahayati, Palembang, Sumatera Selatan

  cat file205.txt | grep Sumatera

  2. Agus Mustofa, Medan, Sumatera Utara

  5. Siti Malahayati, Palembang, Sumatera Selatan

  echo '6. Agus Bashori, Kudus, Jawa Tengah' >> file205.txt

  grep Agus file205.txt

  2. Agus Mustofa, Medan, Sumatera Utara

  6. Agus Bashori, Kudus, Jawa Tengah

grep -i dan -v

- grep -i (case insensitive)

  grep Ra file205.txt

  4. Raden Mas Azzam Altaf, Trenggalek, Jawa Timur

  grep -i Ra file205.txt

  2. Agus Mustofa, Medan, Sumatera Utara

  4. Raden Mas Azzam Altaf, Trenggalek, Jawa Timur

  5. Siti Malahayati, Palembang, Sumatera Selatan

- grep -v digunakan untuk mendapatkan output yang tidak sesuai dengan string tertentu

  grep -vi jawa file205.txt

  2. Agus Mustofa, Medan, Sumatera Utara

  3. Husni Ilyas, Slemen, Yogyakarta

  5. Siti Malahayati, Palembang, Sumatera Selatan

grep -An -Bn dan -Cn

- Menampilkan baris yang dihasilkan bersama dengan n baris setelah (A), sebelum (B) atau setelah dan sebelum (C)
- Contoh

  grep -A1 Raden file205.txt

  4. Raden Mas Azzam Altaf, Trenggalek, Jawa Timur

  5. Siti Malahayati, Palembang, Sumatera Selatan

  husni@husni-Z475:~$ grep -B1 Raden file205.txt

  3. Husni Ilyas, Slemen, Yogyakarta

  4. Raden Mas Azzam Altaf, Trenggalek, Jawa Timur

  husni@husni-Z475:~$ grep -C1 Raden file205.txt

  3. Husni Ilyas, Slemen, Yogyakarta

  4. Raden Mas Azzam Altaf, Trenggalek, Jawa Timur

  5. Siti Malahayati, Palembang, Sumatera Selatan

cut

- Digunakan untuk mengambil kolom tertentu dari suatu file.
- Contoh: mengambil field pertama dan ketiga dari file /etc/passwd. Antar field dipisahkan oleh titik-dua (:)

  cut -d: -f1,3 /etc/passwd | tail -4

  saned:114

  husni:1000

  mysql:115

  jetty:116
- Pemisah spasi harus diapit tanda petik

  cut -d" " -f2 file205.txt | tail -2

  Siti

  Agus

cut -cawal-akhir

- Mengambil baris file dari mulai karakter posisi awal s.d akhir
- Perintah berikut digunakan untuk mendapatkan karakter ke-5 s.d 15 dari file /etc/passwd

  cut -c5-15 /etc/passwd | tail -4

  d:x:114:123

  i:x:1000:10

  l:x:115:126

  y:x:116:127

tr

- Digunakan untuk men-TRanslasi karakter tertentu
- Contoh: mengganti huruf e dengan A

  cat file205.txt | tr 'e' 'A' | tail -2

  5. Siti Malahayati, PalAmbang, SumatAra SAlatan

  6. Agus Bashori, Kudus, Jawa Tangah
- Contoh: mengubah semuanya menjadi huruf BESAR

  at file205.txt | tr 'a-z' 'A-Z' | tail -2

  5. SITI MALAHAYATI, PALEMBANG, SUMATERA SELATAN

  6. AGUS BASHORI, KUDUS, JAWA TENGAH
- Contoh: mengganti ganti baris dengan spasi

  cat file201.txt | tr '\n' ' '

  satu dua tiga empat lima

tr

- tr -s digunakan untuk menghilangkan banyak karakter berulang, hanya menjadi satu

  cat > file2010.txt

  sssatuuuu

  duuuua

  cat file2010.txt | tr -s 'u'

  sssatu

  dua

- tr dapat digunakan untuk mengenrkip dengan 'rot13'

  cat file205.txt | tr 'a-z' 'nopqrstuvwxyzabcdefghijklm' | tail -2

  5. Svgv Mnynunlngv, Pnyrzonat, Shzngren Sryngna

  6. Athf Bnfubev, Khqhf, Jnjn Tratnu

- Dapat pula ditulis: cat file205.txt | tr 'a-z' 'n-za-m' | tail -2

tr -d

- Digunakan untuk menghapus karakter tertentu
- Contoh: hapus setiap huruf e yang ditemukan!

  cat file205.txt | tr -d e

  2. Agus Mustofa, Mdan, Sumatra Utara
  3. Husni Ilyas, Slmn, Yogyakarta
  4. Radn Mas Azzam Altaf, Trnggalk, Jawa Timur
  5. Siti Malahayati, Palmbang, Sumatra Slatan
  6. Agus Bashori, Kudus, Jawa Tngah

WC

- Perintah untuk menghitung jumlah karakter (-c), kata (-w) dan baris (-l).
- Contoh

```
wc -c file205.txt
207 file205.txt
wc -w file205.txt
31 file205.txt
wc -l file205.txt
5 file205.txt
wc file205.txt
  5  31 207 file205.txt
```

sort

- Perintah untuk pengurutan secara alfabet

  sort file201.txt

  dua

  empat

  lima

  satu

  tiga

- Pengurutan berdasarkan nomor kolom dapat dilakukan dengan opsi -kn.

  sort -k2 file205.txt

  sort -k4 file205.txt

- Pengurutan berdasarkan nilai numerik: sort -n -k1 file205.txt

uniq

- Digunakan untuk menghilangkan duplikasi
- Opsi -c digunakan untuk menghitung jumlah kemunculan

```
cat file210.txt
Joko
Ani
Joko
Susi
sort file210.txt
Ani
Joko
Joko
Susi
```

```
sort file210.txt | uniq
Ani
Joko
Susi
sort file210.txt | uniq -c
      1 Ani
      2 Joko
      1 Susi
```

comm

- Digunakan untuk membandingkan file (*stream*).
- Output default dari comm terdiri dari 3 kolom. Jika ingin menghilangkan kolom m dan n maka tambahkan opsi -mn
- Isi kedua file yang dibandingkan harus urut abjad

cat > file210.txt

Adi

Budi

Joko

Madun

cat > file211.txt

Ani

Budi

Maman

Zaenab

```
husni@husni-Z475:~$ comm file210.txt file211.txt
Adi
        Ani
                Budi
Joko
Madun
        Maman
        Zaenab
```

comm -12 file210.txt file211.txt

Budi

comm -13 file210.txt file211.txt

Ani

Maman

Zaenab

od

- Digunakan untuk menampilkan isi file dalam notasi tertentu, misal ascii, octal atau hexadecimal

  cat > file220.txt

  abc

  123

  A

  od -t x1 file220.txt

  0000000 61 62 63 0a 31 32 33 0a 41 0a

  0000012

  od -b file220.txt

  0000000 141 142 143 012 061 062 063 012 101 012

  0000012

  od -c file220.txt

  0000000 a b c \n 1 2 3 \n A \n

  0000012

sed

- Digunakan untuk mengedit stream memanfaatkan ekspresi reguler

  echo Pra-S2 | sed 's/a/o/'

  Pro-S2

  husni@husni-Z475:~$ echo 20 Juni | sed 's/20/22/'

  22 Juni

  husni@husni-Z475:~$ echo 20 Mei 30 Mei | sed 's/Mei/Juni/'

  20 Juni 30 Mei
- Opsi /g digunakan untuk memberikan efek perubahan global

  husni@husni-Z475:~$ echo 20 Mei 30 Mei | sed 's/Mei/Juni/g'

  20 Juni 30 Juni
- Opsi /d digunakan untuk tidak menampilkan baris yang mengandung string tertentu, misalnya 'di'

  cat file210.txt | sed '/di/d'

  Joko

  Madun

Contoh pipe

- who
- who | wc -l
- who | cut -d' ' -f1 | sort
- who | cut -d' ' -f1 | sort | uniq
- grep bash /etc/passwd
- grep bash /etc/passwd | cut -d: -f1

Latihan

- Letakkan semua pengguna bash (terurut) ke dalam file bashusers.txt.
- Letakkan semua pengguna yang login (terurut) ke dalam file onlineusers.txt.
- Buatlah daftar berisi semua file dalam /etc yang mengandung string 'samba'.
- Buat daftar terurut dari semua file dalam /etc yang mengandung string 'samba' tetapi case insensitive .
- Lihat output dari /sbin/ifconfig. Tuliskan sebaris perintah untuk menampilkan hanya IP address dan subnetmasknya
- Tulis sebaris perintah untuk menghapus semua yang bukan huruf dari suatu stream
- Tulis sebaris perintah yang menerima suatu file teks dan meng-outputkan semua kata pada baris terpisah
- Tulis suatu spell checker pada command line. (Ada kamus di dalam direktori /usr/share/dict/)

13 – Latihan Soal 1

Soal 01

- Membuat file bernama “hari demi hari”, menampung 7 baris nama hari. File abc akan menyimpan 3 hari pertama dalam urutan terbalik.

Jawaban 01

cat > “hari demi hari”

Rabu

Kamis

Jumat

Sabtu

Minggu

Senin

Selasa

Ctrl-D

head -3 hari\ demi\ hari | tac > abc ; more abc

Jumat

Kamis

Rabu

Soal dan Jawaban 02

- File cde akan menyimpan kebalikan dari isi file abc (baris demi baris)
- Jawaban:

  tac abc > cde; cat cde

  Rabu

  Kamis

  Jumat

Soal 03

- Perintah untuk memasukkan teks (string) ke dalam file, tetapi string terbaru selalu tersisipkan pada baris paling atas.

Jawaban 03: Cara Pertama

Memanfaatkan perintah cat untuk menggabung isi dua file

echo **Budi Santoso** > teman; cat teman
Budi Santoso

cat teman > temp1 && echo **Ani Susilawati** > temp2 && cat temp2 temp1 > teman; more teman
Ani Susilawati
Budi Santoso

cat teman > temp1 && echo **Manis Manja** > temp2 && cat temp2 temp1 > teman; more teman
Manis Manja
Ani Susilawati
Budi Santoso

cat teman > temp1 && echo **Joko Budianto** > temp2 && cat temp2 temp1 > teman; more teman
Joko Budianto
Manis Manja
Ani Susilawati
Budi Santoso

Jawaban 03: Cara Kedua

Menggunakan satu file temporer yang berisi baris urut sesuai kronologi dimasukkan. Kemudian hasil tac terhadap file temporer dimasukkan ke file teman.

echo **Andi Surapati** > temp; more temp

Andi Surapati

echo **Diana Panggabean** >> temp && tac temp > teman; more teman

Diana Panggabean

Andi Surapati

echo **M Shalih** >> temp && tac temp > teman; more teman

M Shalih

Diana Panggabean

Andi Surapati

Soal 04

- Menggabungkan nilai suatu variabel langsung ke dalam string tertentu tanpa spasi.
- Silakan dicoba contoh-contoh di slide berikut!

Jawaban 04

```
Super=’Super’
echo $Super

echo “$Superman dan $Superwoman sedang belajar SO”
echo ‘$Superman dan $Superwoman sedang belajar SO’
echo $Super’man’ dan $Super’woman’ sedang belajar SO
echo $Super”man” dan $Super”woman” sedang belajar SO
echo “$Super”man dan “$Super”woman sedang belajar SO
echo $Super\man dan $Super\woman sedang belajar SO
echo ${Super}man dan ${Super}woman sedang belajar SO
```

Perhatikan apa yang dihasilkan?

Contoh Lain

```
d1=Dumb
d2=do

echo ${d1}le${d2}re
```

Dumbledore

```
echo $d1′le’$d2′re’
```

Dumbledore

```
echo “$d1″le”$d2″re
```

Dumbledore

Soal dan Jawaban 05

- Menampilkan history dari shell Linux dan simpan ke dalam file sejarah
- Jawaban:

  history 45 > sejarah

14 – Tool Linux Dasar

Tool Linux Dasar

- find
- locate
- date
- cal
- sleep
- time
- gzip - gunzip
- zcat - zmore
- bzip2 - bunzip2
- bzcat - bzmore
- Latihan

find

- Digunakan untuk mencari file
- find /etc
- find /etc -name "*.conf"
- find . -name "*.conf" ← hanya dalam direktori aktif, termasuk sub direktorinya
- find . -type f -name "*.conf" ← tipe file
- find /data -type d -name "*.bak" ← tipe direktori
- find . -newer file42.txt ← lebih baru daripada file42.txt
- find . -name "*.doc" -exec cp {} /backup/ \;
- find . -name "*.txt" -ok rm {} \;

locate

- Bertugas mencari file. Menggunakan index yang dibangun sebelumnya. Index diupdate dengan perintah updatedb
- Contoh

  locate Pra-S2

  /home/husni/.~lock.Praktikum SO Pra-S2 - 03.odp#

  /home/husni/Praktikum SO Pra-S2 - 03.odp

  /home/husni/Praktikum SO Pra-S2 - 03.pdf

  /home/husni/.config/libreoffice/3/user/backup/Praktikum SO Pra-S2 - 03.odp_0.odp

date

- Menampilkan jam, tanggal, timezone, dan lain-lain
- Contoh

  date

  Tue Jun 19 21:27:54 WIT 2012

  date +'%A %d-%m-%Y'

  Tuesday 19-06-2012

  date +%s

  1340116172 ← jumlah detik sejak 01 01 1970

  date -d '1970-01-01 + 2000000000 seconds'

  Wed May 18 03:33:20 WIT 2033

calendar

- Menampilkan kalendar bulan ini dan hari ini ditandai.

  cal

- Juga dapat digunakan untuk menampilkan kalendar bulan dan tahun tertentu

  cal 06 1978

sleep dan time

- Perintah sleep menyebabkan delay selama n detik
- Contoh:

  sleep -10
- Perintah time digunakan untuk mengetahui berapa lama suatu perintah dieksekusi
- Contoh:

  time date

  time sleep 5

  time bzip2 file220.txt

gzip dan gunzip

- Digunakan untuk mengompress dan mendekompres file
- Contoh: memadatkan dan menguraikan file file220.txt

  ls -lh file220.txt

  gzip file220.txt

  ls -lh file220.txt.gz

  gunzip file220.txt.gz

  ls -lh file220.txt
- File yang dipadatkan dengan gzip dapat dilihat dengan zcat dan zmore
- Contoh: zcat file220.txt.gz

bzip2 dan bunzip2

- Hasil kompresi bzip2 lebih kecil daripada gzip, tetapi butuh waktu lebih besar
- Contoh

  bzip2 file220.txt

  bunzip2 file220.txt.bz2
- Perintah bzcat dan bzmore dapat digunakan untuk melihat isi dari file yang dipadatkan menggunakan bzip2
- Contoh

  bzmore file220.txt.bz2

Latihan

- Jelaskan perbedaan dua perintah berikut!

  find /data -name "*.txt"

  find /data -name *.txt
- Jelaskan perbedaan dua pernyataan berikut! Apakah keduanya berjalan saat ada 200 file .odf dalam /data? Bagaimana jika ada 2 juta file .odf?

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

  find /data/*.odf > data_odf.txt
- Tulis suatu perintah find yang mencari semua file yang dibuat setelah 30 Januari 2011
- Tuliskan perintah find yang mencari semua file .odf yang dibuat pada Oktober 2011
- Hitung jumlah dari file *.conf di dalam /etc dan semua sub-direktorinya

Latihan

- Dua perintah berikut melakukan hal sama: menyalin file *.odf ke /backup/ . Mengapa menggunakan perintah kedua?

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

  find /data -name "*.odf" -exec cp {} /backup/ \;
- Buat suatu file bernama loctest.txt. Dapatkah file ini ditemukan (dengan locate)?Mengapa? Solusinya?
- Gunakan find dan -exec untuk mengganti semua file .htm menjadi.html
- Jalankan perintah date. Tampilkan tanggal dalam format YYYY/MM/DD.
- Jalankan perintah cal. Tampilkan kalendar 1582 dan 1752. Perhatikan hasilnya? Apa komentar anda?

15 – Dasar Shell Scripting

Dasar Shell Scripting

- Script Pertama: haloo
- Komentar
- Variabel
- Agar Variabel dikenal di Luar Script
- Troubleshooting a script
- Latihan

Script Pertama: haloo

- **Catatan**: pastikan anda telah menguasai dengan baik materi tentang Maksimalisasi Shell dan Pipe & Filtering
- Contoh: file script bernama haloo

  echo echo Saya sedang belajar Linux > haloo

  cat haloo

  echo Saya sedang belajar Linux

  chmod +x haloo ← menjadikan executable

  ./haloo ← mengeksekusi script

  Saya sedang belajar Linux

She-Bang

- String **#!/bin/bash** sering diletakkan pada baris pertama setiap script Shell
- **#!** - dibaca she-bang adalah 2 karakter yang mengawali script shell
- Maksudnya, baris-baris script di bawah #!/bin/bash akan dieksekusi menggunakan program shell bash, bukan shell lain
- cat > script02
- #!/bin/bash
  echo Dua baris terakhir dari /etc/passwd:
  echo `tail -2 /etc/passwd`
- ./script02
  Dua baris terakhir dari /etc/passwd:
  mysql:x:115:126:MySQL

Komentar

- Setiap karakter setelah tanda pound # dianggap komentar
- Selain baris pertama, jika setelah she # diikuti bang !
- Contoh

```
#!/bin/bash
# script: script03
# Hanya menampilkan nama user dan hostname
#
echo $USER login pada $HOSTNAME
```

- chmod +x script03
- ./script03

  husni login pada husni-Z475

Variabel

- Pembuatan variabel pada script shell sama seperti pembuatan variabel langsung pada shell
- Contoh

```
#!/bin/bash
# script04
# menggunakan variabel
var1=”Joko Susanto”
echo Nilai var1 = $var1
```

- Variabel di dalam script tidak dikenali di luar (di Shell Linux)

```
echo $var1
```

Agar Variabel dikenal di Luar Script

- Dikenal dengan istilah *Sourcing The Script*
- Variabel di dalam script, dapat dikenali langsung dari shell
- Contoh

./script04

Nilai var1 = Joko Susanto

echo $var1

source ./script04

Nilai var1 = Joko Susanto

echo $var1

Joko Susanto

Cara Lain:

. ./script04

Nilai var1 = Joko Susanto

echo $var1

Joko Susanto

Troubleshoot Terhadap Script

- File script juga dapat dieksekusi dengan memanggil program bash

  bash script03

  husni login pada husni-Z475

- Tahapan eksekusi, *step by step*, dapat diketahui memanfaatkan opsi -x. Ini bermanfaat men-debug script atau mencari posisi kesalahan dalam script

  bash -x script03

  + echo husni login pada husni-Z475

  husni login pada husni-Z475

Latihan

- Buat sebuah script yang mendefinisikan dua variabel dan menampilkan outputnya
- Buatlah agar script sebelumnya berpengaruh terhadap shell yang sedang digunakan
- Adakah cara lebih singkat untuk meng-*source the script*?
- Berikan komentar di dalam script tersebut sehingga lebih mudah dipahami oleh pemrogram pemula sekalipun
- Buat sebuah script yang mampu menampilkan hostname, IP address, subnet mask dan gateway dari komputer yang anda gunakan!

16 – Seleksi & Perulangan

Seleksi & Perulangan

- Test[ ]
- If then else
- If then elif
- Perulangan for
- Perulangan while
- Perulangan until
- Latihan

Test[ ]

- Perintah test digunakan untuk menguji apakah sesuatu bernilai true atau false.
- Apakah 17 lebih besar daripada 45?

  test 17 -gt 45 ; echo $?

  1

  Hasilnya adalah 1, berarti false (salah)
- Apakah 17 lebih kecil daripada 45?

  test 17 -lt 45 ; echo $?

  0

  Nilai 0 menunjukkan true (benar)

Test[ ]

- Jika nilai numerik (0 dan 1) tidak nyaman, kita dapat memodifikasi bentuk perintah sehingga output berupa string “benar” atau “salah”

  test 17 -lt 45 && echo Benar || echo Salah

  Benar

  test 17 -gt 45 && echo Benar || echo Salah

  Salah

- Pahami kembali peran operator dalam operasi shell Linux

Contoh test[] - Lihat man test

- [-d foo] Adakah direktori bernama foo?
- [-e bar] Adalah file bernama bar?
- ['/etc' = $PWD] Apakah string /etc sama dengan variabel $PWD?
- [$1 != 'rahasia'] Apakah parameter pertama berbeda dengan rahasia?
- [55 -lt $bar] Apakah 55 kurang dari nilai $bar?
- [$foo -ge 1000] Apakah nilai $foo lebih atau sama dengan 1000
- ["abc" < $bar] Apakah urutan abc sebelum nilai dari $bar
- [-f foo] Apakah foo suatu file biasa?
- [-r bar] Apakah bar suatu file readable
- [foo -nt bar] Apakah file foo lebih baru daripada file bar?
- [-o nounset] Apakah opsi Shell nounset diset?

Pemanfaatan Kombinasi AND & OR

- [-d Public] && echo Ada || echo Tidak ada

  Ada
- ['/etc' = $PWD] && echo Iya || echo Bukan

  Bukan
- [66 -gt 55 -a 66 -lt 500] && echo true || echo false

  true
- [66 -gt 55 -a 660 -lt 500] && echo true || echo false

  false
- [66 -gt 55 -o 660 -lt 500] && echo true || echo false

  true

if then else

- Bagaimana menentukan pilihan
- Jika kondisi tertentu terpenuhi maka lakukan sesuatu, jika tidak maka lakukan yang lain
- Contoh: pilihan01

  Tampilkan pesan ada tidaknya file test5.doc di direktori aktif?

  ```
  #!/bin/bash
  if [ -f test5.doc ]
  then echo File test5.doc ADA!
  else echo File test5.doc TIDAK ADA!
  fi
  ```

- chmod +x pilihan01
- ./pilihan01

  File test5.doc ADA!

if then elif

- If bersarang di dalam else
- Contoh: pilihan02

```
#!/bin/bash
# file script: pilihan02
nilai=99
if [ $nilai -eq 23 ]
then
echo "23 TEPAT dan PAS."
elif [ $nilai -gt 23 ]
then
echo "TERLALU BANYAK."
else
echo "TIDAK CUKUP."
fi
```

- chmod +x pilihan02
- ./pilihan02

  TERLALU BANYAK.

Perulangan for

- Lakukan perulangan SEBANYAK yang ditentukan

```
#!/bin/bash
# file script: for01
for i in 1 2 4
do
  echo $i
done
```

- chmod +x for01
- ./for01

```
1
2
4
```

Perulangan for

- Contoh: Bentuk lain dari for (*range*)

```
#!/bin/bash
# file script: for02
for counter in {1..20}
do
  echo Menghitung 1 sampai 20, sekarang: $counter
sleep 1
done
```

- Eksekusi: (apa hasilnya?)

```
chmod +x for02
./for02
```

Perulangan while

- Lakukan perulangan SELAMA kondisi terpenuhi

```
#!/bin/bash
# file script: while01
i=100;
while [ $i -ge 0 ] ;
do
  echo Menghitung mundur, dari 100 s.d 0, Sekarang $i;
let i--;
done
```

- Bagaimana cara mengeksekusi script ini?

Perulangan until

- Lakukan perulangan SAMPAI kondisi terpenuhi

```
#!/bin/bash
# file script: until01
let i=100;
until [ $i -le 0 ] ;
do
  echo Menghitung mundur, dari 100 s.d 1, Sekarang $i;
let i--;
done
```

- Coba eksekusi script until01 di atas!

Latihan

- Tuliskan script yang menggunakan for untuk menghitung dari 1 s.d 17500
- Tuliskan script yang mengguakan while untuk menghitung 3 s.d 9
- Tuliskan script yang menggunakan until untuk menghitung mundur dari 8 s.d 3
- Tuliskan script yang menghitung jumlah dari file yang berakhiran .txt dalam direktori aktif anda
- Bagaimana jika tidak ditemukan file berakhiran .txt? BETULKAN!

17 – Parameter & Opsi

Parameter dan Opsi

- Parameter Script
- Shift
- Input runtime: read
- Men-source-kan file config
- getopts
- Latihan

Parameter Script

- Script shell bash dapat mempunyai parameter
- Contoh

```
#!/bin/bash
# file script: par01
echo 'Parameter pertama:' $1
echo 'Parameter kedua   :' $2
echo 'Parameter ketiga   :' $3
echo \$ $$ - PID dari script ini
echo \# $# - Jumlah parameter
echo \? $? - Kode kembalian terakhir
echo \* $* - Semua parameter
```

- chmod +x par01
- ./par01 Satu Dua Tiga

```
Parameter pertama: Satu
Parameter kedua   : Dua
Parameter ketiga   : Tiga
$ 15530 - PID dari script ini
# 3 - Jumlah parameter
? 0 - Kode kembalian terakhir
* Satu Dua Tiga - Semua parameter
```

Parameter Script

- Parameter atau argumen pertama? $1
- Nama file script? $0
- Contoh

  cat > par02

  echo File script ini bernama $0

  ./par02

  bash: ./par02: Permission denied

  chmod +x par02

  ./par02

  File script ini bernama ./par02

Pernyataan shift

- Digunakan untuk memparse semua parameter satu demi satu
- Contoh

```
#!/bin/bash
# file script: par03
if [ "$#" == "0" ]
then
  echo Sertakan minimal satu parameter.
  exit 1
fi
while (( $# ))
do
  echo Anda telah memberikan $1
  shift
done
```

- chmod +x par03
- ./par03
  Sertakan minimal satu parameter.
- ./par03 Satu Demi Kamu
  Anda telah memberikan Satu
  Anda telah memberikan Demi
  Anda telah memberikan Kamu
- ./par03 "S2 Ilmu Komputer UGM"
  Anda telah memberikan S2 Ilmu Komputer UGM

Input Runtime

- Saat script berjalan, input dari pengguna dapat diminta dengan pernyataan read
- Contoh

```
#!/bin/bash
# file script: run01
echo -n Masukkan suatu angka:
read angka
echo Angka yang dimasukan adalah $angka
```

- chmod +x run01
- ./run01

  Masukkan suatu angka:24

  Angka yang dimasukan adalah 24

Input Runtime

- Contoh

```
#!/bin/bash
# file script: run02
echo 'Masukkan nama awal'
echo -n 'diikuti nama akhir:'
read namaawal namaakhir
echo "Halooo $namaawal $namaakhir"
```

- chmod +x run02
- ./run02

```
Masukkan nama awal
diikuti nama akhir:Susi Susanti
Halooo Susi Susanti
```

Menjadikan file config sebagai source

- Sering digunakan untuk men-*source*-kan file konfigurasi
- Contoh: Menggunakan isi file myApp.conf di dalam file script shell myApp.sh

```
# File konfigurasi dari script myApp
# nama file: myApp.conf
# Tentukan path di sini
myAppPath=/var/myApp
# Tentukan jumlah pengguna
users=5
```

Menjadikan file config sebagai source

- Contoh pemanggilan

```
#!/bin/bash
# file script: myApp.sh
# Welcome to the myApp application
. ./myApp.conf
echo Ada sebanyak $users pengguna
```

- chmod +x myApp.sh
- ./myApp.sh

  Ada sebanyak 5 pengguna

getopts

- Digunakan untuk mendapatkan opsi-opsi yang disertakan saat memanggil shell script
- Contoh

```
#!/bin/bash
# file script: get01
while getopts ":afz" option;
do
  case $option in
   a)
     echo received -a
   ;;
   f)
     echo received -f
   ;;
   z)
     echo received -z
   ;;
   *)
     echo "invalid option -$OPTARG
   ;;
  esac
done
```

- chmod +x get01
- ./get01 -afZX
  received -a
  received -f
  invalid option -Z

getopts

- Suatu opsi juga dapat diharusnya menyertakan parameter
- Contoh Eksekusi
- chmod +x get02
- ./get02 -a -f Husni

  diterima -a

  diterima -f dengan Husni
- ./get02 -a -f Husni -z

  diterima -a

  diterima -f dengan Husni

  diterima -z
- ./get02 -a -f

  diterima -a

  Opsi -f memerlukan parameter.

getopts

```
#!/bin/bash
# file script: get02
while getopts ":af:z" opsi;
do
  case $opsi in
  a)
    echo diterima -a
  ;;
  f)
    echo diterima -f dengan $OPTARG
  ;;
  z)
    echo diterima -z
  ;;
  :)
    echo "Opsi -$OPTARG memerlukan parameter."
  ;;
  *)
    echo "Opsi tidak valid -$OPTARG"
  ;;
  esac
done
```

Latihan

- Tulis suatu script yang menerima dua parameter (berupa nama file) dan output-kan ada atau tidaknya kedua file tersebut
- Tulis suatu script yang meminta masukan nama file. Pastikan ada tidaknya file tersebut. Apakah file tersebut *writable*? Jika tidak buat agar *writeable*.
- Buat suatu script myApp2.sh yang melibatkan file konfigurasi myApp2.conf. Pastikan script meminta input dari pengguna. Simpan setiap interaksi pengguna ke dalam file myApp2.log

18 – Scripting Lanjut

Scripting Lanjut

- Apa kegunaan empat pernyataan berikut?

  eval

  (( ))

  let

  case

- Bagaimana membuat fungsi shell?
- Kerjakan soal-soal pada halaman Latihan

Latihan

- Tulis suatu *script* yang meminta input dua bilangan dan output-kan hasil penjumlahan dan perkaliannya
- Perbaiki script no.1 untuk memastikan bahwa bilangan yang dimasukkan hanya 1 s.d 100, jika tidak maka keluar dengan pesan error.
- Perbaiki script no.1. Beritahukan pengguna jika hasil penjumlahan sama dengan hasil perkalian
- Tulis suatu script dengan melibatkan pernyataan case yang insensitive, gunakan opsi shopt nocasematch. Opsi nocasematch berguna me-reset ke nilai yang dipegang sebelum script dijalankan.
- Pelajari script sistem Linux dalam /etc/init.d dan /etc/rc.d dan coba pahami.

19 – Latihan Soal 2

Soal-Soal

- Buat sebuah script shell yang dapat digunakan untuk mengatur IP address, netmask dan gateway dari Linux Ubuntu yang anda gunakan
- Buat sebuah script yang mengkonfigurasi file wvdial.conf secara lengkap sehingga dapat terkoneksi ke Internet. Pastikan isian nomor telp, username dan password dapat diterima dari pengguna. Silakan belajar tentang wvdial dan konfigurasi modem di Linux Ubuntu.
- Tampilkan semua file berawalan s, berakhiran .conf dan dibuat sebelum tanggal 01 Juli 2012
- Menggunakan shell script, buat program untuk menyimpan (ke dalam file teman) dan menampilkan data teman anda (Nama, email dan Blog). Program ini dipanggil dengan menyertakan opsi dan parameter, misal:

  namaprogram tambah -n Husni -e husni@mail.ugm.ac.id -b http://komputasi.wordpress.com

Soal-Soal

- Buatlah suatu script yang akan mendownload halaman web tertentu yang dimasukkan pengguna saat program berjalan. Setelah download selesai, tampilkan informasi mengenai file tersebut
- Buat script yang menerima argumen berupa akhiran file saat dipanggil. Script kemudian akan menampilkan semua file dengan akhiran tersebut.

Soal-Soal

- Pelajari kembali cara menggunakan perintah ls dan find (misal: man ls dan man find). Gunakan perintah find atau ls untuk menampilkan/mencari file dengan kriteria berikut:

  a. Dibuat pada tanggal 11 Juli 2012

  b. Dibuat sebelum 11 Juli 2012

  c. Dibuat 18 hari yang lalu

  d. Dibuat antara 1 Januari s.d 30 Juni 2012

  e. Berukuran (size) lebih dari 10 KB

  f. Berukuran tidak lebih dari 150 KB

Soal-Soal

- Buat sebuah script shell (melibatkan opsi & parameter) untuk mengkonfigurasi IP address, subnet mask, gateway dan DNS dari mesin Linux yang digunakan (gunakan ifconfig atau tulis ke file interfaces). Uji apakah konfigurasi tersebut (gunakan perintah ping ke suatu host)!

  Contoh eksekusi:

  mynetconfig if eth0 -ip 10.1.1.1 -mask 255.255.255.0 -gw 10.1.1254 -dns 212.121.212.2

Soal-Soal

- Kita dapat menggunakan kombinasi perintah ls dan grep untuk menampilkan file yang dibuat pada tanggal tertentu.

  Pertanyaan:

  Bagaimana cara mengganti tanggal pembuatan dari file-file yang dibuat pada tanggal tertentu (dahulu) dengan tanggal dan jam saat ini (dalam satu baris perintah)?

20 – Proses dan Sinyal

Proses & Sinyal

- Istilah Penting
- Dasar Proses
- Sinyal
- Latihan

Istilah Penting

- **process**

  Source code ter-compile yang sedang berjalan (running) pada sistem.

- **PID**

  Setiap proses mempunyai suatu process id atau PID.

- **PPID**

  Setiap proses mempunyai suatu proses induk (parent, dengan suatu PPID). Proses anak (child) sering dimulai oleh proses induk.

- **init**

  Proses init selalu mempunyai process ID bernilai 1. Proses init dimulai oleh kernel sendiri sehingga secara teknis tidak mempunyai suatu proses induk. init bertindak sebagai suatu induk foster bagi proses-proses orphaned.

Istilah Penting

- **kill**

  Saat suatu proses berhenti berjalan, proses tersebut mati (die). Perintah kill digunakan untuk mematikan proses tertentu.

- **daemon**

  Proses yang berjalan pada saat system startup dan tetap berjalan (selamanya) disebut proses daemon atau daemon. Proses demikian tidak pernah mati.

- **zombie**

  Ketika suatu proses dimatikan (killed), tetapi masih ada di dalam sistem, maka proses demikian dinamakan zombie. Zombie ini tidak dapat dikill karena “sudah mati”.

$$, $PPID dan pidof

- Beberapa variabel lingkungan shell mengandung informasi mengenai proses. Variabel **$$** akan memegang ID proses yang berjalan.Variabel **$PPID** menyimpan PID dari proses induk.
- Sebenarnya, $$ adalah suatu parameter shell, bukan variabel, sehingga tidak dapat dilewatkankan suatu nilai.
- echo $$ $PPID

  2371 2063
- Kita dapat menemukan semua ID proses dengan nama menggunakan perintah **pidof**.
- pidof firefox

  2205
- pidof apache2

  1198 1197 1196 1194 1193 1181

Induk dan Anak

- Proses-proses mempunyai hubungan *parent-child* (induk-anak). Setiap proses mempunyai suatu proses induk.
- Saat suatu bash baru dibuat, echo dapat digunakan untuk memastikan bahwa pid dari proses sebelum merupakan ppid (induk) dari shell baru tersebut (sebelumnya itu adalah proses anak).

  echo $$ $PPID

  3628 2065

  bash

  echo $$ $PPID

  4088 3628
- Perintah exit akan mengakhiri proses aktif dan kembali ke nilai awal dari $$ dan $PPID.

  exit

  echo $$ $PPID

  3628 2065

Fork dan Exec

- Suatu proses memulai proses lain dalam dua fase. Pertama, membuat fork dari dirinya, salinan identik. Kedua, proses yang di-fork tadi mengeksekusi suatu exec untuk menggantikan proses yang di-fork dengan proses anak target.

  echo $$

  3441

  bash

  echo $$ $PPID

  3500 3441

- Dengan perintah **exec**, proses dieksekusi tanpa men-fork proses baru.

  echo $$

  4144

  ksh

  echo $$ $PPID

  4198 4144

  exit
  echo $$
  4144
  exec ksh
  echo $$ $PPID
  4144 2113

ps

- Perintah ini digunakan untuk melihat proses. Perhatikan hubungan induk-anak pada 3 proses bash berikut.

```
echo $$ $PPID
4230 2113
bash
echo $$ $PPID
4284 4230
bash
echo $$ $PPID
4338 4284
```

```
ps fx
2113  /usr/bin/xfce4-terminal
4230   \_ bash
4284      \_ bash
4338         \_ bash
4392            \_ ps fx
exit
ps fx
exit
ps fx
```

ps fax

- Perintah ps fax digunakan untuk menampilkan semua proses secara detail.

```
ps fax
2113  /usr/bin/xfce4-terminal
2114   \_ [xfce4-terminal] <defunct>
2115   \_ bash
2173   |   \_ sudo wvdial
2176   |       \_ wvdial
2178   |           \_ /usr/sbin/pppd 9600 modemcrtscts def
4230   \_ bash
4407       \_ ps fax
```

- Format lain ps yang sering digunakan adalah ps -ef

pgrep dan ps -C

- Seperti **ps -C**, **pgrep** digunakan untuk mencari proses berdasarkan nama perintahnya.

  sleep 1000 &

  [1] 4472

  pgrep sleep

  4472

  pgrep wvdial

  2176

  ps -C wvdial

  PID TTY TIME CMD

  2176 pts/0 00:00:10 wvdial

- Opsi **-l** pada pgrep dapat digunakan untuk menampilkan PID dan nama perintah dari proses sekaligus

  pgrep -l wvdial

  2176 wvdial

  pgrep -l sleep

  4472 sleep

top

- Perintah top dapat mengurutkan proses-proses sesuai dengan pemanfaatan CPU atau properti lain. Proses di dalam top juga dapat dimatikan (kill).
- Ketik h untuk bantuan tutorial
- Tool top sering juga menyediakan informasi mengenai memory (RAM) dan swap (virtual memory)

Mengirimkan Sinyal ke Proses

- **kill**
- Perintah kill digunakan untuk mematikan atau menghentikan suatu proses.
- sleep 1000 &
- [1] 4518
- pgrep sleep
- 4518
- kill 4518
- pgrep sleep
- [1]+ Terminated sleep 1000
- pgrep sleep
- [kosong]
- Menggunakan kill, Linux telah mengirimkan sinyal ke suatu proses.

Daftar Sinyal

- Proses-proses berjalan dapat saling menerima sinyal, begitu pula dari penggna.
- Perintah kill -l menampilkan 64 sinyal yang disediakan oleh SO.

Kill -l

1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL 5) SIGTRAP
6) SIGABRT 7) SIGBUS 8) SIGFPE 9) SIGKILL 10) SIGUSR1
11) SIGSEGV 12) SIGUSR2 13) SIGPIPE 14) SIGALRM 15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP
21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO 30) SIGPWR
31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX

kill -1 (SIGHUP)

- Sinyal pertama SIGHUP, HUP atau 1. Digunakan untuk memberitahukan suatu proses bahwa ia harus membaca ulang (re-read) file konfigurasi.
- Perintah kill -1 1 memaksa proses init (init selalu berjalan dengan PID 1) untuk membaca ulang file konfigurasinya.

kill -1 1

- Tergantung kepada pengembang apakah proses dapat melakukan ini running, atau apakah memerlukan stop dan start. Terserah kepada pengguna untuk membaca dokumentasi program.

kill -15 (SIGTERM)

- Sinyal SIGTERM juga disebut *standard kill*.
- Kapanpun kill dieksekusi tanpa menentukan sinyalnya, dianggap kill -15.
- Kedua bentuk perintah di bawah ini memberikan hasil identik.

kill 1942 atau

kill -15 1942

kill -9 (SIGKILL) dan killall

- Sinyal SIGKILL berbeda dari kebanyakan sinyal lain dimana tidak dikirim ke proses tetapi ke kernel Linux. kill -9 dikenal pula sebagai *sure kill*.
- Kernel akan menghentikan (*shoot down*) proses tersebut. Pengembang tidak punya cara untuk meng-*intercept* sinyal kill -9.

  kill -9 3342
- Perintah killall secara default juga mengirimkan sinyal 15 ke proses.
- Perintah ini dan killall5 pada SysV digunakan untuk men-*shut down* sistem.

  grep killall /etc/init.d/halt

  action $"Sending all processes the TERM signal..." /sbin/killall5 -15

  action $"Sending all processes the KILL signal..." /sbin/killall5 -9

pkill

- Perintah pkill digunakan untuk mematikan suatu proses berdasarkan nama perintahnya.

  Contoh:

  sleep 1000 &

  [1] 5541

  pkill sleep

  [1]+ Terminated sleep 1000

- Di dalam jendela top, menekan tombol keyboard k akan memunculkan prompt, kemudian kita dapat memasukkan PID dari daftar proses yang ada untuk dikill.

  PID to kill:

SIGSTOP dan SIGCONT

- Suatu proses yang sedang berjalan dapat di-suspend ketika ia meneriman sinyal SIGSTOP. Sama dengan sinyal kill -19 di Linux, tetapi berbeda pada sejumlah sistem Unix lain.
- Proses yang di-suspend tidak menggunakan siklus CPU tetapi tetapi berada di Memory dan dapat digerakkan ulang dengan sinyal SIGCONT (kill -18 pada Linux).
- Penjelasan dan contoh penggunaan kedua sinyal ini ada pada bagian proses-proses background.

Latihan

- Gunakan perintah **ps** untuk mencari proses init berdasarkan namanya.
- Berapakah **process id** dari proses init tersebut?
- Gunakan perintah **who am i** untuk mengetahui nama terminal yang digunakan
- Memanfaatkan nama terminal di atas, gunakan perintah **ps** untuk menemukan semua proses yang berkaitan dengan terminal tersebut.
- Berapakah **process id** dari Shell yang sedang digunakan?
- Berapakah **parent process id** dari Shell tersebut?
- Buatlah dua instance dari perintah **sleep 3342** di belakang layar
- Berapakah **process id** dari semua perintah sleep?
- Tampilkan hanya **dua proses sleep** di dalam top. Kemudian keluar top
- Gunakan perintah **kill standard** untuk mematikan salah satu dari proses sleep tersebut
- Gunakan **satu perintah** untuk mematikan semua proses sleep

21 – Prioritas Proses

Prioritas Proses

- Prioritas
- Perintah nice dan renice
- Latihan

Nilai Priority & Nice

- Semua proses mempunyai suatu nilai prioritas (*priority*) dan bagus (*nice*).
- Proses dengan prioritas lebih tinggi (*higher*) akan mendapatkan lebih banyak cpu time dari pada proses dengan prioritas rendah (*lower*).
- Kita dapat mempengaruhi ini dengan perintah nice dan renice.

Pipes (mkfifo)

- Proses-proses dapat berkomunikasi satu dengan lainnya melalui pipes. Pipe-pipe ini dapat dibuat dengan perintah mkfifo.
- Contoh: Empat pipe berbeda dalam satu direktori.

```
mkdir proses
cd proses/
mkfifo pipe1 pipe2 pipe3 pipe4
ls -l
total 0
prw-rw-r-- 1 husni husni 0 Aug 13 09:43 pipe1
prw-rw-r-- 1 husni husni 0 Aug 13 09:43 pipe2
prw-rw-r-- 1 husni husni 0 Aug 13 09:43 pipe3
prw-rw-r-- 1 husni husni 0 Aug 13 09:43 pipe4
```

Perintah cat

- Untuk memperlihatkan manfaat perintah top dan nice, kita buat cat menggunakan pipe-pipe sebelumnya untuk membangkitkan suatu beban penuh pada CPU.
- Perintah cat disalin ke nama berbeda.

  cp /bin/cat kucing1

  cp /bin/cat kucing2

  echo -n x | ./kucing1 - pipe1 > pipe2 &

  [1] 2210

  ./kucing1 < pipe2 > pipe1 &

  [2] 2215

  echo -n z | ./kucing2 - pipe3 > pipe4 &

  [3] 2217

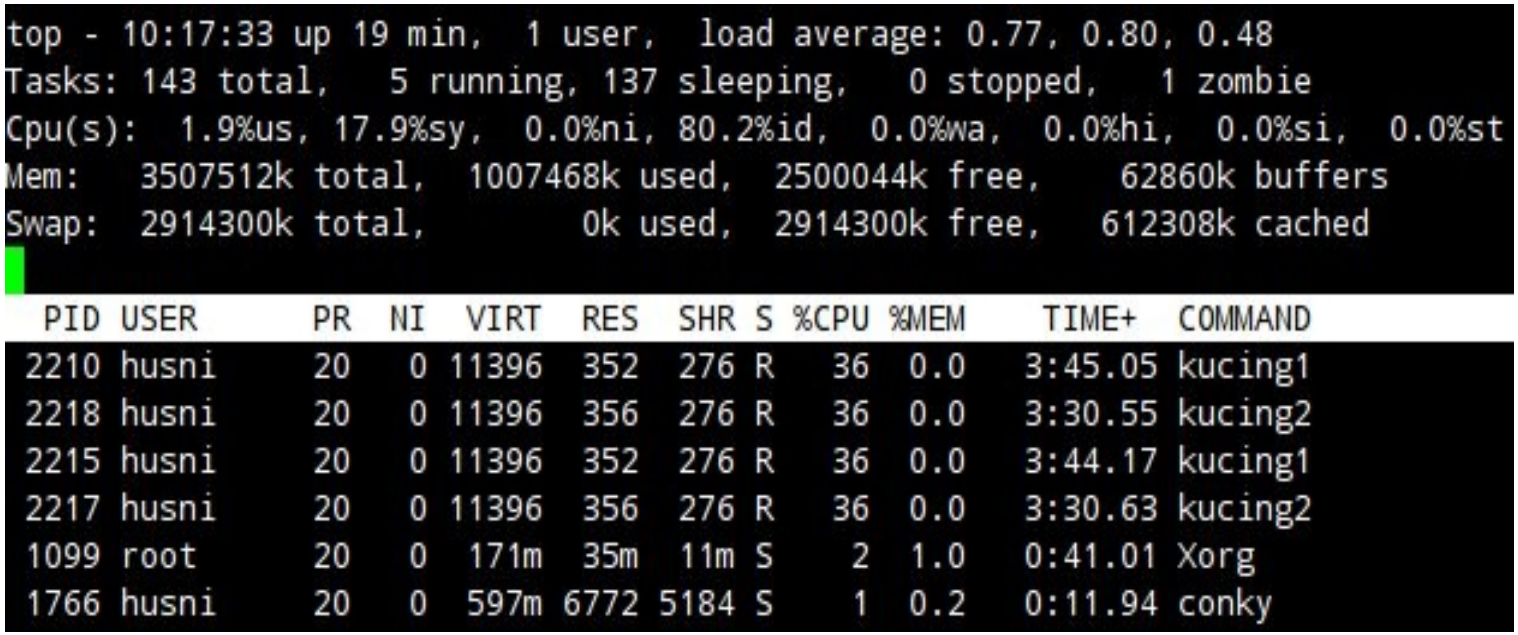
  husni@Z475:~/proses$ ./kucing2 < pipe4 > pipe3 &

  [4] 2218

- Perintah-perintah di atas akan membuat dua proses kucing1 yang menggunakan cat untk mengikat karakter x antara pipe1 dan pipe2. Begitu pula, untuk karkter z dan kucing2.

Gunakan top

- Jalankan top tanpa opsi untuk menampilkan semua proses. Pada bagian atas, terlihat seperti ini

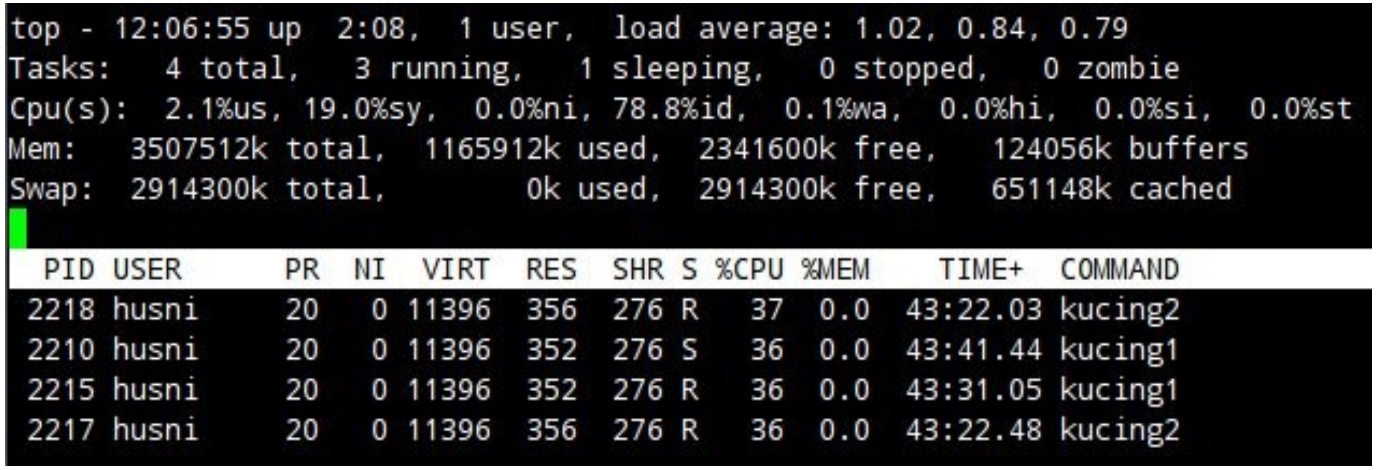


- Pada processor dengan core tunggal, cpu idle time (0.0%id) bernilai nol. Ini karena proses cat menghabiskan keseluruhan cpu. Hasil di atas diambil pada processor dengan 4 core.

top -p

- Perintah top dengan opsi -p dapat digunakan untuk memantau hanya proses-proses tertentu

  top -p 2210, 2215, 2217, 2218



- Rata-rata waktu processor untuk setiap proses adalah sekitar 1/3 (36%).
- Pada kebanyakan sistem Linux, setiap proses yang diakibatkan eksekusi aplikasi oleh pengguna, mempunyai nilai prioritas sama, biasanya 20.

renice

- Nilai nice (NI) awal dari setiap proses juga sama, yaitu 0.
- Perintah renice dapat digunakan untuk mengubah nilai nice dari suatu proses yang sedang berjalan (berdasarkan PID-nya).
- Contoh: cara renice dua proses kucing1.

  renice +8 2210

  2210 (process ID) old priority 0, new priority 8

  renice +8 2215

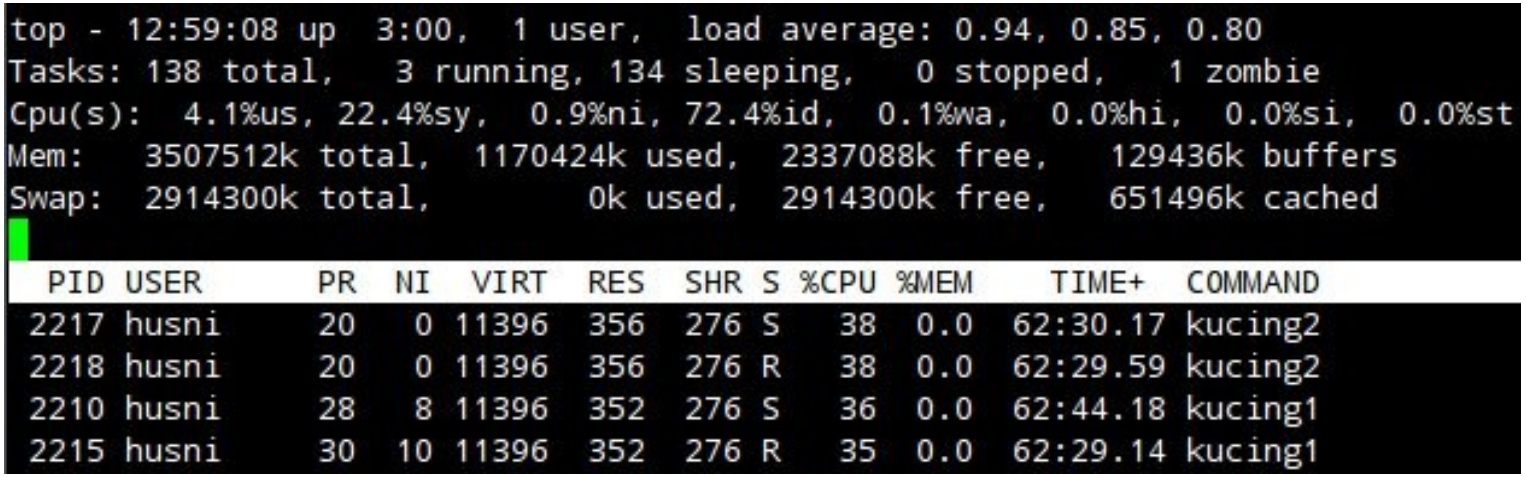
  2215 (process ID) old priority 0, new priority 8

  renice +10 2215

  2215 (process ID) old priority 8, new priority 10
- Pengguna normal dapat memberikan nilai nice antara 0 dan 20. Hanya root yang dapat memberikan nilai nice negatif. **HATI-HATI**. Nilai negatif dapat menyebabkan tidak berfungsinya keyboard, atau kesulitan SSH ke suatu sistem.

Pengaruh Nilai nice

- Dampak dari suatu nilai nice terhadap proses yang sedang berjalan dapat beragam.
- Proses nilai nilai nice lebih tinggi akan mempunyai prioritas lebih rendah, sehingga proses kucing1 selalu berada di belakang kucing2.
- Biasanya, persentase CPU time (%CPU) proses dengan nilai besar menjadi lebih kecil, artinya tidak mendapat jatah CPU lebih atau sama dengan proses yang nilai nicenya lebih kecil.



- Proses yang lebih penting, diberikan nilai nice lebih kecil. Nilai nice negatif dapat memberikan pengatuh serere terhadap usability sistem.

nice

- Perintah nice bekerja sama seperti renice tetapi digunakan saat memulai suatu perintah.
- Contoh: menjalankan program leafpad dengan nilai nice awal 5 (bukan 0)

nice -5 sleep 1000 &

nice -10 leafpad

```
PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+
COMMAND       2940 husni    25   5 11376  608  512 S    0  0.0
0:00.00 sleep        2946 husni    30  10  168m  11m 8972 S    0
0.3   0:00.15 leafpad
```

Latihan

- Buat suatu direktori baru dan enam pipe di dalam direktori tersebut.
- Ikat suatu karakter di antara dua pipe.
- Gunakan perintah top dan ps untuk menampilkan informasi (pid, ppid, priority, nice value, ...) mengenai dua proses cat ini.
- Ikat karakter lain antara dua pipe lain, tetapi kali ini tetapkan nilai dari perintahnya. Pastikan bahwa semua proses cat berebutan CPU (usahakan pada processor tunggal).

  (Coba buat dua cat lagi dengan pipe yang tersisa).
- Gunakan perintah ps untuk memastikan bahwa dua proses cat baru mempunyai suatu nilai nice. Gunakan opsi -o dan -C.
- Gunakan perintah renice untuk menaikkan nilai nice dari 10 menjadi 15.

22 – Proses Background

Proses Background

- Proses Belakang Layar
- Latihan

Perintah jobs

- Aktifitas yang berjalan di belakang layar (*background*) dari shell aktif kita dapat ditampilkan dengan perintah jobs. Secara default, tidak ada proses pengguna yang berjalan di *background*.
- Contoh:

  jobs

  sleep 1000 &

  [1] 9363

  jobs

  [1]+ Running sleep 1000 &
- Bagian ini akan sering menggunakan perintah jobs.

Ctrl Z

- Suatu proses dapat di-*suspend* dengan kombinasi tombol keyboard Ctrl-Z. Sinyal SIGSTOP akan dikirimkan ke kernel Linux, secara efektif membekukan operasi dari proses tersebut.
- Contoh

  Jika Ctrl-Z diberlakukan terhadap nano maka nano akan menjadi proses *background*. nano yang background ini dapat dilihat dengan perintah jobs.

  nano test1.txt

  jobs

  [1]- Running sleep 1000 &

  [2]+ Stopped nano test1.txt

Karakter &

- Proses-proses yang dapat dimulai sebagai proses *background* (menyertakan karakter & pada akhir *command line*-nya) juga dapat dilihat dengan perintah jobs.
- Contoh

```
find / > semuafile.txt 2>/dev/null &
[3] 9401
[1]   Done                    sleep 1000
jobs
[2]+  Stopped                 nano test1.txt
[3]-  Running                 find / > semuafile.txt 2> /dev/null &
```

jobs -p

- Opsi -p pada perintah jobs dapat digunakan untuk melihat process id dari proses-proses *background*.

```
sleep 500 &
find / > semuafile.txt 2>/dev/null &
jobs -p
9379
9404
9423
9430
ps `jobs -p`
  PID    TTY      STAT   TIME     COMMAND
 9379   pts/3    T         0:00        nano test1.txt
 9404   pts/3    S      0:00     sleep 1000
 9423   pts/3    S      0:00     sleep 500
 9430   pts/3    R      0:00     find /
```

Perintah fg

- Perintah fg digunakan untuk mengambil suatu proses background dan membawanya ke foreground sehingga terlihat sebagaimana proses umumnya. Nomor dari proses (bukan PID) background dijadikan sebagai parameter dari fg.
- Contoh:

  Membawa ke depan proses “nano test1.txt” yang bernomor 2 dari background

  jobs

  [2]+ Stopped nano test1.txt

  [4] Running sleep 1000 &

  [5] Running sleep 500 &

  fg 2

Perintah bg

- Proses-proses yang di-suspend di background dapat dimulai (dijalankan) tetap di background dengan perintah bg.
- Perintah bg akan mengirimkan sinyal SIGCONT.
- Contoh: Perintah sleep di-suspend dan diaktifkan kembali di latar-belakang.
- sleep 500
- ^Z
- [3]+ Stopped sleep 500
- husni@Z475:~$ jobs
- [1]   Running sleep 1000 &
- [2]- Running sleep 1500 &
- [3]+ Stopped sleep 500
- bg 3
- [3]+ sleep 500 &
- jobs
- [1]   Running sleep 1000 &
- [2]- Running sleep 1500 &
- [3]+ Running sleep 500 &

Latihan

- Gunakan perintah jobs untuk memastikan apakah anda mempunyai proses yang berjalan di background.
- Gunakan perintah vi untuk membuat suatu file teks. Suspend vi tersebut ke background.
- Pastikan dengan perintah jobs bahwa vi benar-benar tersusped di background.
- Mulailah **find / > semuafile.txt 2>/dev/null** di foreground. Kemudian suspend ke background sebelum selesai.
- Mulailah dua proses sleep yang panjang dalam background.
- Tampilkan semua proses yang ada di background.
- Gunakan perintah kill untuk menghentikan proses sleep terakhir.
- Lanjutkan proses find di background (pastikan berjalan lagi).
- Letakkan salah stau perintah sleep kembali ke foreground.

Latihan

- Jelaskan darimana datangnya angka-angka di bawah ini! Kapan variabel digantikan oleh nilainya? Dengan shell apa?

  echo $$ $PPID

  4224 4223

  bash -c "echo $$ $PPID"

  4224 4223

  bash -c 'echo $$ $PPID'

  5059 4224

  bash -c `echo $$ $PPID`

  4223: 4224: command not found