

Shared Library di Linux

Aplikasi umumnya tidak berdiri sendiri, namun membutuhkan pustaka lain agar dapat bekerja. Hal ini sangat umum di berbagai sistem operasi. Di Windows, kita mengenal DLL. Di Linux, kita mengenal berbagai file .SO. Artikel ini akan membahas hal-hal seputar pengaturan pustaka di Linux agar sistem senantiasa berjalan dengan baik.

Tanpa aplikasi, komputer tentu saja akan menjadi tidak menarik. Aplikasi secara langsung menjembatani pengguna dan sistem operasi. Di Linux, kita mengenal belasan ribu atau bahkan lebih aplikasi baik besar maupun kecil, baik yang kompleks ataupun sederhana. Aplikasi-aplikasi tersebut dibangun dengan puluhan (atau lebih) bahasa pemrograman yang memiliki karakteristik berbeda-beda. Seperti disebutkan sebelumnya, umumnya aplikasi tidak berdiri sendiri, namun akan membutuhkan pustaka lain untuk dapat bekerja. Bisa dibayangkan betapa kompleksnya sistem apabila terdapat ratusan atau lebih aplikasi terinstal di sistem yang masing-masing membutuhkan beberapa pustaka (yang mungkin membutuhkan pustaka lain lagi). Pengaturan pustaka dan sistem yang baik secara langsung akan menyebabkan sistem kita sehat wal'afiat tidak kurang apapun.

Berikut ini, kita akan membahas hal-hal seputar pengaturan pustaka di Linux, mulai dari dasar-dasar pengenalan pustaka sampai berbagai masalah umum dan penyelesaiannya. Kita akan mulai dengan mengenal pustaka di Linux.

Apa itu pustaka?

Pustaka atau *library* secara sederhana bisa diartikan sebagai kumpulan fungsi. Sebuah file pustaka misalnya, merupakan kumpulan beberapa fungsi yang memiliki tugas spesifik.

Sebuah aplikasi umumnya akan memiliki ketergantungan terhadap beberapa kategori pustaka berikut ini:

- Pustaka sistem. Pustaka sistem ini berarti pustaka yang datang bersama sistem Linux. Bisa dikatakan, pustaka-pustaka

pada kategori ini merupakan pustaka yang dibutuhkan oleh semua aplikasi yang berjalan pada sistem Linux yang terinstal.

- Pustaka yang datang bersama bahasa pemrograman. Pustaka-pustaka pada kategori ini mungkin diperlukan, mungkin tidak. Beberapa bahasa pemrograman yang besar dan kompleks, misal Borland Kylix, datang bersama cukup banyak pustaka bawaan yang akan digunakan oleh aplikasi yang dihasilkan. Tapi, model bahasa pemrograman tersebut tidaklah terlalu umum. Banyak bahasa yang tidak mengharuskan aplikasi yang dihasilkan tetap tergantung pada pustaka yang datang bersama bahasa pemrograman.
- Pustaka-pustaka lain. Ini adalah yang paling banyak. Pustaka-pustaka di kategori ini mungkin diperlukan, mungkin tidak. Tapi, umumnya, aplikasi besar tidak akan lepas dari ketergantungan terhadap pustaka-pustaka di kategori ini.

Di Windows, pustaka secara fisik adalah file-file DLL yang sangat banyak ditemukan pada direktori Windows dan subdirektori di dalamnya, serta pada subdirektori di program files. Di Linux, secara umum, pustaka secara fisik adalah file-file SO (berekstensi .so) yang tersebar di /lib, /usr/lib, /usr/local/lib, di tempat-tempat lain seperti di /opt dan mungkin home directory user serta lokasi lainnya.

Berikut ini adalah contoh beberapa pustaka berawalan libc di /lib:

```
$ ll /lib/libc*
lrwxrwxrwx 1 root root 11
2005-06-28 10:12 /lib/libcap.so
-> libcap.so.1
```

```
lrwxrwxrwx 1 root root 14
2005-06-28 10:12 /lib/libcap.
so.1 -> libcap.so.1.92
-rwxr-xr-x 1 root root 33063
2005-03-19 12:07 /lib/libcap.
so.1.92
-rwxr-xr-x 1 root root 182420
2005-03-19 10:45 /lib/libcidsn.
so.1
lrwxrwxrwx 1 root root 17
2005-06-28 10:14 /lib/libcom_
err.so.2 -> libcom_err.so.2.1
-rwxr-xr-x 1 root root 6392
2005-03-19 12:10 /lib/libcom_
err.so.2.1
-rwxr-xr-x 1 root root 39464
2005-03-19 10:45 /lib/libcrypt.
so.1
-rwxr-xr-x 1 root root 1205944
2005-03-19 10:45 /lib/libc.so.6
```

Sebagai catatan, satu hal yang khas dengan penamaan pustaka di Linux adalah nama file yang (selalu) diawali oleh lib.

Pustaka-pustaka sistem yang tersedia memiliki fungsi-fungsi spesifik yang siap digunakan oleh aplikasi-aplikasi yang membutuhkannya.

Kenapa tidak sendiri saja?

Pustaka memang menyediakan banyak fasilitas bagi aplikasi. Dengan tersedianya ribuan pustaka di Linux, *developer* dapat memanfaatkan pustaka-pustaka tersebut untuk membangun berbagai solusi.

Namun, seperti yang kita bahas sebelumnya, dengan bergantungnya aplikasi pada pustaka, maka kita akan berada pada sistem yang cukup rumit untuk diatur.

Beberapa dari Anda mungkin akan ber-

tanya. Kenapa aplikasi tidak berdiri sendiri saja? Bebas dari ketergantungan terhadap pustaka? Atau, datang bersama pustaka sendiri dan tidak menggunakan yang berada di sistem? Berikut ini mungkin beberapa alasan yang bisa diterima:

- Apabila aplikasi memaksakan untuk mengimplementasikan semua fungsi sendiri, lepas dari ketergantungan terhadap pustaka yang telah ada, maka untuk membangun satu aplikasi saja, kita akan membutuhkan waktu yang sangat lama. Ini jelas bertentangan dengan dunia *free software* yang memiliki keinginan untuk saling berbagi. Dengan menggunakan *resource* (misal: pustaka), diharapkan masing-masing bisa membangun aplikasi yang semakin baik. Beberapa aplikasi memang harus tergantung pada pustaka tertentu karena mengimplementasikan sendiri sangat membutuhkan waktu. Contoh yang baik adalah ketika Anda ingin membangun aplikasi GUI. Ingat bahwa di Linux, GUI hanyalah aplikasi user space yang dimungkinkan oleh proyek Xfree86/X.org. Tidak dimasukkan ke dalam kernel. Jadi, apabila Anda ingin membuat aplikasi GUI dan tidak ingin memiliki ketergantungan pada berbagai pustaka pembuat GUI seperti GTK+, QT dan lain-lain, maka Anda harus berurusan dengan Xlib (yang sangat

susah). Sebagai kesimpulan, ada kondisi di mana kita harus bergantung.

- Apabila aplikasi memaksakan untuk datang bersama pustaka sendiri (yang mungkin dibuat sendiri atau dicomot dari komunitas *free software*), maka akan ada kondisi di mana terjadi redundansi pustaka di sistem (yang mungkin memiliki versi yang berbeda-beda) yang berpotensi untuk membuat sistem tidak stabil ataupun lebih susah untuk diadministrasi. Model datang dengan pustaka sendiri ini sangat disenangi oleh aplikasi-aplikasi di Windows. Hampir semua aplikasi besar akan datang dengan pustaka sendiri (yang mungkin sangat proprietary) yang kemudian diletakkan di dalam direktori aplikasi. Redundansi pustaka sangat berpotensi terjadi, ditambah, kita banyak melakukan pekerjaan sia-sia dengan menulis ulang kembali sesuatu yang sudah ada. Di Windows, hal ini mungkin disebabkan juga karena masalah lisensi yang tidak mengijinkan suatu pustaka untuk digunakan oleh pihak lain. Namun, di Linux, hal ini tentunya lebih berbeda. Karena kita berada di dunia *free software*, maka kita sebaiknya menggunakan sesuatu yang telah ada. Bahasa *kerennya*, *use the battery!* Apabila kita tetap menggunakan pustaka yang ada namun membundelnya sendiri, maka kita ber-

potensi untuk menimbulkan redundansi pustaka dan menambah masalah administrasi sistem.

- Apabila aplikasi mencoba untuk melakukan kompilasi secara statik, maka ukuran aplikasi akan membesar, dan, di masa depan, ada kemungkinan beberapa aplikasi yang digunakan tidak lagi bisa dikompilasi secara statik. Kompilasi secara statik adalah usaha untuk memasukkan semua kode pustaka ke dalam kode *executable*. Dengan demikian, aplikasi tidak lagi membutuhkan pustaka agar dapat dijalankan. Pendekatan kompilasi statik hanya disarankan untuk aplikasi yang dijalankan pada kondisi ekstrim (seperti untuk rescue). Untuk lebih jelasnya, di bagian akhir tulisan, kita akan membahas lebih lanjut tentang isu ini.

Sebagai kesimpulan, aplikasi sangat sulit untuk berdiri sendiri. Apa yang bisa kita lakukan adalah memahami baik-baik sistem pustaka di Linux dan memanfaatkannya sebaik mungkin pula.

Melihat kebutuhan pustaka

Di Linux, kita dengan mudah bisa melihat apa saja kebutuhan pustaka suatu aplikasi ataupun pustaka. Dengan fasilitas yang disediakan tersebut, hendaknya kita bisa menjaga sistem kita lebih baik dan bersih.

Untuk melihat kebutuhan pustaka aplikasi atau pustaka, gunakanlah program `ldd`, yang datang bersama paket `glibc`, yang pasti akan selalu tersedia di sistem Linux.

Berikut ini adalah contoh penggunaan program `ldd` untuk melihat kebutuhan pustaka program `ls`:

```
$ ldd /bin/ls
        linux-gate.so.1 =>
(0xffffe000)
        librt.so.1 => /lib/tls/
librt.so.1 (0x4002f000)
        libc.so.1 => /lib/
libc.so.1 (0x40038000)
        libselinux.so.1 => /lib/
libselinux.so.1 (0x4003e000)
        libc.so.6 => /lib/tls/
libc.so.6 (0x4004d000)
        libpthread.so.0 => /lib/
tls/libpthread.so.0 (0x40166000)
        /lib/ld-linux.so.2
(0x40000000)
```



Situs web GNU Libc.

```
libattr.so.1 => /lib/
libattr.so.1 (0x40178000)
```

Untuk program yang dikompilasi statik (semua kebutuhan pustaka telah dimasukkan ke dalam aplikasi), ldd akan menampilkan pesan yang jelas seperti contoh berikut:

```
$ ldd /bin/sash
not a dynamic executable
```

Namun, apabila Anda ingin menguji ketergantungan aplikasi dengan saksama, pesan not a dynamic executable tersebut tidak selalu menjamin bahwa Anda cukup mengopikan program yang diinginkan ke tempat tujuan. Benar apabila program yang Anda inginkan adalah program yang dihasilkan dengan cara dikompilasi. Namun, apabila program merupakan script (PHP, Python, Perl, shell, dan lain-lain), maka ldd tidak dapat digunakan sama sekali. Ini disebabkan karena script-script tersebut merupakan file teks biasa. Jadi, berhati-hatilah. Yakinkan bahwa program yang diinginkan benar-benar program yang dikompilasi. Gunakan program file seperti contoh berikut:

```
$ ldd /bin/sash
not a dynamic executable
```

```
$ file /bin/sash
/bin/sash: ELF 32-bit LSB
executable, Intel 80386, version
1 (SYSV), for GNU/Linux 2.2.5,
statically linked, stripped
```

Contoh tersebut membuktikan bahwa /bin/sash merupakan aplikasi statik yang benar-benar merupakan program yang dikompilasi. Program /bin/sash aman di-copy-kan ke sistem lain.

Perhatikan contoh yang sedikit menipu berikut ini:

```
$ ldd /usr/bin/ldd
not a dynamic executable
```

```
$ file /usr/bin/ldd
/usr/bin/ldd: Bourne shell
script text
```

Terlihat pada contoh tersebut bahwa walaupun ldd terlihat sebagai aplikasi yang tidak membutuhkan pustaka lain, tapi sesungguhnya, ldd merupakan sebuah shell script.

Kita akan melanjutkan ke pembahasan mengenai cara melihat kebutuhan pustaka pada pustaka. Program yang digunakan adalah program yang sama:

```
$ ldd /usr/lib/libapr.so.0
linux-gate.so.1 =>
(0xffffe000)
librt.so.1 => /lib/tls/
librt.so.1 (0x40038000)
libm.so.6 => /lib/tls/
libm.so.6 (0x40041000)
libcrypt.so.1 => /lib/
libcrypt.so.1 (0x40064000)
libnsl.so.1 => /lib/
libnsl.so.1 (0x40096000)
libpthread.so.0 => /lib/
tls/libpthread.so.0 (0x400ab000)
libdl.so.2 => /lib/
libdl.so.2 (0x400bd000)
libc.so.6 => /lib/tls/
libc.so.6 (0x400c2000)
/lib/ld-linux.so.2
(0x80000000)
```

Terakhir, masih dalam konteks kebutuhan pustaka, terkadang kita bisa saja berhadapan dengan kasus yang cukup menjengkelkan. Lihatlah contoh berikut ini, di mana kita akan melihat kebutuhan satu aplikasi berikut:

```
$ ldd Student
linux-gate.so.1 =>
(0xffffe000)
libgtk-x11-2.0.so.0
=> /opt/gnome/lib/libgtk-x11-
2.0.so.0 (0x4002f000)
libgdk-x11-2.0.so.0
=> /opt/gnome/lib/libgdk-x11-
2.0.so.0 (0x402f7000)
libatk-1.0.so.0 =>
/opt/gnome/lib/libatk-1.0.so.0
(0x40372000)
libgdk_pixbuf-2.0.so.0
=> /opt/gnome/lib/libgdk_pixbuf-
2.0.so.0 (0x4038b000)
libpangoft2-1.0.so.0
=> /opt/gnome/lib/libpangoft2-
1.0.so.0 (0x4039f000)
libpangox-1.0.so.0 => /
opt/gnome/lib/libpangox-1.0.so.0
(0x403a6000)
libpango-1.0.so.0 =>
/opt/gnome/lib/libpango-1.0.so.0
(0x403b2000)
```

[DIPOTONG]

```
libjpeg.so.62 => /usr/
lib/libjpeg.so.62 (0x404ea000)
libtiff.so.3 => /usr/
lib/libtiff.so.3 (0x40509000)
libm.so.6 => /lib/tls/
libm.so.6 (0x40559000)
libpq.so.3 => not found
libstdc++.so.5 =>
/tmp/TMP/stdc/libstdc++.so.5
(0x4057d000)
libgcc_s.so.1 => /lib/
libgcc_s.so.1 (0x40639000)
[DIPOTONG]
libXcursor.so.1 =>
/usr/X11R6/lib/libXcursor.so.1
(0x4092e000)
libXrender.so.1 =>
/usr/X11R6/lib/libXrender.so.1
(0x40937000)
libXext.so.6 => /
usr/X11R6/lib/libXext.so.6
(0x4093f000)
```

```
libpangoft2-1.0.so.0
=> /opt/gnome/lib/libpangoft2-
1.0.so.0 (0x4094d000)
/lib/ld-linux.so.2
(0x40000000)
libexpat.so.0 => /usr/
lib/libexpat.so.0 (0x40974000)
```

```
$ ldd Student | wc -l
37
```

Bisa dilihat bahwa aplikasi Student tersebut membutuhkan 37 pustaka, dengan pustaka berikut ini tidak ditemukan pada sistem:

```
$ ldd Student | grep -i found
libpq.so.3 => not found
```

Bisa kita lihat bahwa hanya terdapat satu pustaka yang lupa terinstal. Ini bisa sedikit menipu apabila Anda lantas terburu-buru mengopikan libpq.so.3 ke sistem dan berharap aplikasi dapat berjalan sesuai harapan. Pada kenyataannya, ketika libpq.so.3 ditambahkan ke sistem, total kebutuhan pustaka akan menjadi 46, bukan tetap 37.

```
$ ldd Student | wc -l
46
```

Hal ini disebabkan karena libpq.so.3 sendiri membutuhkan beberapa pustaka

lain sebagai berikut:

```
$ ldd /usr/lib/libpq.so.3
        linux-gate.so.1 =>
        (0xffffe000)
        libssl.so.0.9.7 => /usr/
        lib/libssl.so.0.9.7 (0x40033000)
        libcrypto.so.0.9.7 =>
        /usr/lib/libcrypto.so.0.9.7
        (0x40063000)
        libkrb5.so.3 => /usr/
        lib/libkrb5.so.3 (0x40156000)
        libcrypt.so.1 => /lib/
        libcrypt.so.1 (0x401c9000)
        libresolv.so.2 => /lib/
        libresolv.so.2 (0x401fb000)
        libnsl.so.1 => /lib/
        libnsl.so.1 (0x4020e000)
        libpthread.so.0 => /lib/
        tls/libpthread.so.0 (0x40224000)
        libc.so.6 => /lib/tls/
        libc.so.6 (0x40236000)
        libdl.so.2 => /lib/
        libdl.so.2 (0x4034f000)
        libk5crypto.so.3 =>
        /usr/lib/libk5crypto.so.3
        (0x40353000)
        libcom_err.so.2 => /lib/
        libcom_err.so.2 (0x40376000)
        /lib/ld-linux.so.2
        (0x80000000)
        libkrb5support.so.0 =>
        /usr/lib/libkrb5support.so.0
        (0x4037a000)
```

Oleh karena itu, sebelum Anda mendapatkan semua pustaka yang dibutuhkan oleh suatu aplikasi, jangan terburu-buru mengopikan pustaka ke dalam USB disk Anda untuk dibawa ke sistem lain.

Lebih jauh tentang lokasi pustaka

Kita telah melihat ketergantungan aplikasi terhadap pustaka-pustaka di sistem. Berikut ini, kita akan melihat lebih jauh tentang beberapa isu seputar lokasi pustaka di sistem.

Pertama-tama, kita akan membahas bahwa sistem telah menyediakan beberapa direktori berikut ini untuk menyimpan pustaka:

- /lib, merupakan direktori tempat menyimpan pustaka yang paling penting bagi sistem. GLibc (akan dibahas kemudian) pun disimpan disini. Kehilangan pustaka yang tersimpan di direktori ini

sangat berpotensi untuk membuat sistem tidak dapat berjalan sama sekali.

- /usr/lib, merupakan direktori tempat menyimpan pustaka-pustaka sistem yang paling umum. Hampir semua pustaka akan disimpan di direktori ini. Kehilangan beberapa pustaka di direktori ini mungkin menyebabkan sistem terganggu, mungkin pula tidak.
- /usr/local/lib, merupakan direktori tempat menyimpan pustaka yang dikompilasi sendiri oleh user, atau direktori tempat menyimpan pustaka yang tidak terlalu penting bagi jalannya sistem. Sangat disarankan untuk menyimpan semua pustaka kompilasi sendiri di direktori ini.

Namun, beberapa aplikasi besar (oleh berbagai distro), menyimpan pustaka di dalam direktori aplikasinya sendiri (seperti konsep program files di Windows). Contohnya adalah /opt/kde3 di distro SUSE. Direktori /opt/kde3 dimaksudkan sebagai rumah bagi aplikasi kde3. Di dalam direktori ini, terdapat tree direktori yang cu-

kup komplut, termasuk lib (/opt/kde3/lib) yang berguna untuk menyimpan pustaka-pustaka yang dibutuhkan oleh kde3. Kondisi demikian bahwa kde3 menyimpan pustaka di /opt/kde3/lib bukanlah aturan KDE, namun lebih kepada pembuat distro-nya. Distro Redhat AS 4 misalnya, menyimpannya bersama pustaka-pustaka lain di sistem.

Perhatikanlah baik-baik keluaran ldd berikut ini untuk melihat lokasi pustaka:

```
$ ldd /bin/lis
        linux-gate.so.1 =>
        (0xffffe000)
        librt.so.1 => /lib/tls/
        librt.so.1 (0x4002f000)
        libacl.so.1 => /lib/
        libacl.so.1 (0x40038000)
        libselinux.so.1 => /lib/
        libselinux.so.1 (0x4003e000)
        libc.so.6 => /lib/tls/
        libc.so.6 (0x4004d000)
        libpthread.so.0 => /lib/
        tls/libpthread.so.0 (0x40166000)
        /lib/ld-linux.so.2
        (0x40000000)
```

MORE SPACE RELIABILITY & TIME LESS... TIME & MONEY

LINUX and FreeBSD

Features :

- Unlimited data transfer
- Complete control panels
- POP3 email, FTP access
- SSH, CGI, SQL.
- and much more...
- Start from Rp. 19.500,-/ month
- Free Setup *)
- 2 Months Free **)

Server Hosting

Features :

- Location NOC Jakarta - Indonesia (IIX)
- Size server : 1 U Rackmount
- Bandwidth : 128 kbps
- IP Address : 8 (max)
- Colocation : Rp. 1.000.000,-/ month

ALSO

- Colocation & Dedicated Server in USA
- Domain Name Register
- Benefit Reseller Program

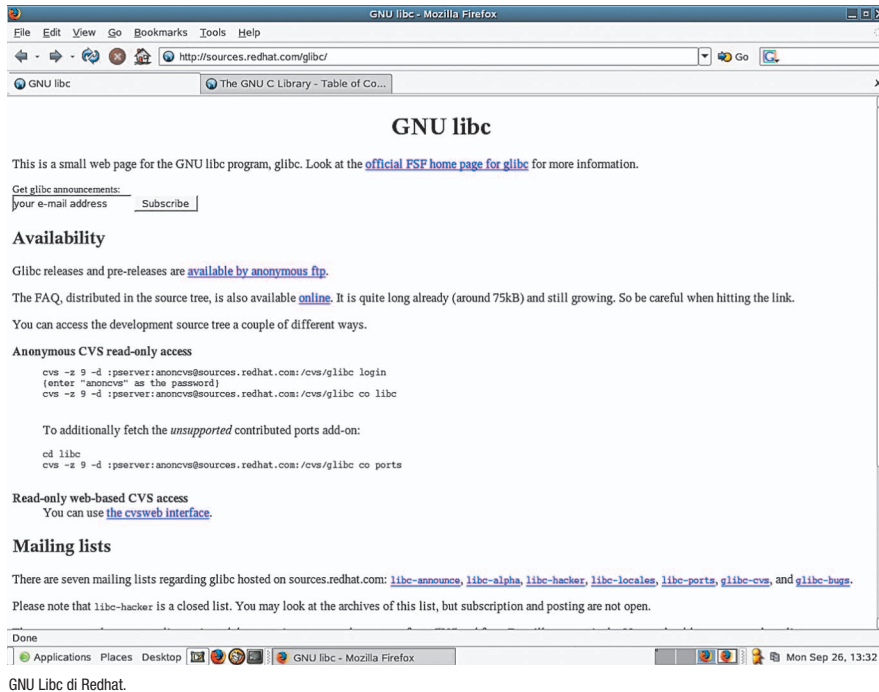
Limited Offer :
Dedicated Server
Rp. 1.250.000,-/ mo

*"IT'S NEVER BEEN EASIER
TO TAKE YOUR BUSINESS ONLINE"*

Note : *) Transfer (restriction apply)
**) 1 year payment

CAKRAWEB
Supporting You to a Web Success

Cyber Building (d/h Elektrindo) 10 th Floor
Jl. Kuningan Barat No. 8 Jakarta Selatan 12710
Phone. (021) 526 8000 Fax. (021) 52 66 444
<http://www.cakraweb.com> - info@cakraweb.com



```
libattr.so.1 => /lib/
libattr.so.1 (0x40178000)
```

Sebelah kiri adalah pustaka yang dibutuhkan, sementara sebelah kanan adalah lokasi fisik file (dengan pengecualian linux-gate.so.1, yang akan dibahas lebih lanjut).

ld.so.conf, registry pustaka

Sistem Linux kita memiliki sebuah file bernama ld.so.conf yang terletak di /etc. Di file inilah kita bisa menambah lokasi pustaka baru yang dituliskan baris demi baris. Untuk menulis ke file ini, akses root akan diperlukan dan daftar pustaka yang telah ditambahkan akan berlaku untuk semua user.

Berikut ini adalah contoh file /etc/ld.so.conf milik penulis:

```
/usr/X11R6/lib/Xaw95
/usr/X11R6/lib/Xaw3d
[DIPOTONG]
/opt/gnome/lib
/opt/gnome2/lib
include /etc/ld.so.conf.d/*.conf
```

Di file ini, selain menambahkan lokasi direktori secara langsung, kita juga bisa menggunakan perintah *include* untuk memasukkan file konfigurasi yang tersimpan di lokasi lain. Di contoh ld.so.conf milik penulis, semua file *.conf yang tersimpan di

/etc/ld.so.conf.d/ akan dimasukkan juga.

Setelah ld.so.conf diubah, kita perlu memanggil program ldconfig yang akan membuat cache dari semua pustaka, yang selanjutnya akan disimpan di /etc/ld.so.cache secara default. Umumnya, ldconfig akan dipanggil dengan argumen -v untuk menampilkan proses secara verbose. Contoh:

```
# ldconfig -v
```

File ld.so.cache sendiri merupakan file binary yang berisikan cache yang nantinya bisa digunakan oleh sistem. Untuk melihat isi cache, kita bisa memberikan perintah ldconfig dengan argumen -p, sebagai contoh:

```
# /sbin/ldconfig -p | less
1088 libs found in cache `/etc/ld.so.cache'
libzvt.so.2 (libc6) =>
/opt/gnome/lib/libzvt.so.2
libzvbi.so.0 (libc6) =>
/usr/lib/libzvbi.so.0
libzvbi-chains.so.0
(libc6) => /usr/lib/libzvbi-chains.so.0
libzio.so.0 (libc6) =>
/usr/lib/libzio.so.0
libzio.so (libc6) =>
/usr/lib/libzio.so
...
...
...
```

Menambahkan pustaka user

Terkadang, kita men-download aplikasi tertentu yang datang dengan pustakanya sendiri dan kita berada pada sistem besar di mana kita tidak memiliki hak untuk mengubah ld.so.conf. Dalam kondisi yang umum, kita juga tidak bisa meng-copy-kan begitu saja file .SO ke dalam direktori aktif aplikasi tersebut.

Apa yang bisa kita lakukan agar aplikasi dapat menemukan pustaka yang dibutuhkannya? Caranya, kita bisa mengubah variabel shell LD_LIBRARY_PATH. Berikut ini adalah caranya:

```
$ export LD_LIBRARY_PATH=~:/temp/
program/USE/gtkextra/lib:~/temp/
program/USE/wx25/lib:/tmp/TMP/
wx261:/tmp/TMP/stdc
```

Contoh tersebut akan mengekspor variabel shell LD_LIBRARY_PATH yang berisikan beberapa lokasi sekaligus (dipisahkan oleh titik dua).

Satu hal yang penting adalah, sebaiknya kita tidak menumpuk pustaka di dalam home direktori kita. Sistem akan berpotensi untuk memiliki pustaka yang redundan. Selain itu, cukup rumit untuk melakukan pengaturan pustaka-pustaka tersebut, misalnya ketika *upgrade* pustaka ingin dilakukan.

Pustaka yang wajib dimiliki

Kita tentunya tidak perlu memiliki semua pustaka yang datang bersama distro agar dapat menjalankan semua aplikasi yang ada. Selain bisa menyebabkan sistem menjadi terlalu kotor, beberapa pustaka di masa depan mungkin saja akan konflik satu dengan lainnya.

Berikut ini adalah beberapa pustaka (di luar pustaka sistem) yang mungkin ingin Anda miliki:

- Apabila Anda termasuk penggemar aplikasi *desktop* dan sangat berpotensi untuk menginstal aplikasi desktop, maka pastikan Anda telah memiliki pustaka GTK+ (versi 1.x dan 2.x), Qt dan beberapa pustaka lain seperti fltk dan tk.
- Apabila Anda seorang developer, maka Anda mungkin akan membutuhkan berbagai pustaka seperti libxml2, berbagai pustaka gambar seperti libjpeg, giflib, dan libpng, serta pustaka-pustaka lain. Dengan memanfaatkan pustaka yang

ada, Anda tidak perlu lagi mengimplementasikan sendiri berbagai kebutuhan Anda.

Apabila Anda adalah pengguna sistem biasa, maka pastikan Anda mengikuti instalasi default pada sistem. Secara umum, distro yang Anda gunakan seharusnya telah memaketkan pustaka-pustaka standar sehingga apabila tidak diperlukan, instalasi pustaka tambahan tidak akan dilakukan.

Glibc: pustaka super penting

Glibc adalah GNU C Library, pustaka mendasar sistem yang akan digunakan oleh semua aplikasi di Linux. Pustaka ini sangat penting dan tidak boleh diutak-atik. Libc disimpan di /lib.

Apa yang terjadi apabila program yang Anda download membutuhkan libc yang berbeda dengan apa yang Anda miliki? Apakah yang akan Anda lakukan, satu hal yang jelas, jangan pernah mengganti libc milik Anda.

Cara lain yang lebih bijak:

- Carilah aplikasi ke dalam CD/DVD Anda.
- Apabila tidak ditemukan, carilah paket untuk distro Anda di Internet.
- Apabila tidak ditemukan juga (kasus yang cukup jarang terjadi), cara yang lebih disarankan adalah dengan melaku-

kan kompilasi sendiri.

- Cara lain yang bisa digunakan adalah dengan memiliki dua atau lebih libc di sistem. Cara ini digunakan pada berbagai distro. Butuh pengaturan (dan kehati-hatian) ekstra untuk melakukannya dan secara umum tidak disarankan.

Kompilasi secara statis

Berikut ini kita akan membuat program sederhana menggunakan bahasa C, dan melihat perbedaan antara kompilasi statis dan kompilasi dengan memanfaatkan pustaka sistem.

Berikut ini adalah source code programnya, a.c:

```
#include <stdio.h>

int main()
{
    printf("Hello World\n");
    return 0;
}
```

Berikut ini adalah kompilasi dengan memanfaatkan pustaka sistem:

```
$ gcc a.c -o a
$ ls -al a
-rwxr-xr-x 1 nop users 7045
2005-08-17 02:55 a
```

Berikut ini kita akan melihat pustaka yang dibutuhkan oleh a:

```
$ ldd a
        linux-gate.so.1 =>
(0xffffe000)
        libc.so.6 => /lib/tls/
libc.so.6 (0x4002f000)
        /lib/ld-linux.so.2
(0x40000000)
```

Sekarang, kita akan melihat kompilasi secara statis:

```
$ gcc a.c -o a -static
$ ls -al a
-rwxr-xr-x 1 nop users 461686
2005-08-17 02:57 a
```

Program a tidak membutuhkan pustaka sistem sebagaimana terlihat berikut ini:

```
$ ldd a
        not a dynamic executable
```

Sebagai catatan, perhatikanlah perbedaan ukuran antara program yang dikompilasi statis dan tidak. Pada contoh kita kali ini, perbedaan ukurannya mencapai 65 kali. Luar biasa, bukan?

Linux-gate

Apabila diperhatikan sebelumnya, setiap kali mendaftar pustaka yang dibutuhkan oleh sebuah aplikasi, kita akan melihat bahwa aplikasi membutuhkan linux-gate.so.1, yang tidak merujuk kepada file apapun di sistem.

Linux-gate ini tentu bukan semacam skandal yang umumnya diakhiri dengan akhiran gate. Linux-gate yang hadir di kernel 2.6 merupakan *gateway* antara user dan kernel. Semacam fasilitas baru pada kernel 2.6 untuk mengoptimalkan penggunaan pustaka sistem.

Sebagai kesimpulan, Anda tidak perlu mencari ketersediaan file ini di filesystem Anda. Sistem yang terjaga dengan baik tidak akan bermasalah dengan linux-gate.

Sampai di sini pembahasan kita mengenai pustaka sistem di Linux. Satu hal yang benar-benar harus diperhatikan adalah bagaimana kita menjaga sistem agar tetap bersih, sambil tidak menjadikan kita paranoid dan tidak mau menginstal apapun. Dengan memahami sistem pustaka di Linux, kita memiliki satu modal untuk membantu menjaga kebersihan sistem. Selamat mencoba!

Noprianto (noprianto@infolinux.co.id)

