Lightweight virtualization with GoboLinux' Runner

Lucas C. Villa Real
lucasvr@gobolinux.org

About GoboLinux

- Alternative distribution born in 2002
- Explores novel ideas in the Linux distribution ecosystem
- Introduces a rather different directory hierarchy

How different?

```
lucasvr@fedora ~] ls /
bin   dev   home   lib64        media   opt   root   sbin   sys   usr
boot  etc   lib    lost+found   mnt     proc  run    srv    tmp   var

lucasvr@fedora ~] ls /usr
bin  games  include  lib  lib64  libexec  local  sbin  share  src  tmp

lucasvr@fedora ~] ls /usr/local
bin  etc  games  include  lib  lib64  libexec  sbin  share  src
```

```
lucasvr@gobolinux ~] ls /
Data   Mount   Programs   System   Users
```

GoboLinux File System Hierarchy

/Programs

Self-contained programs: no need for a package manager

```
~] ls /Programs
```

AbsTk	DiffUtils	GnuTLS	Kerberos	LibXML2
ACL	Dit	GoboHide	Kmod	LibXSLT
Acpid	DosFSTools	GParted	Lame	Linux
AGNClient	E2FSProgs	Gperf	LCMS	Linux-Firmware
ALSA-Lib	EFIBootMgr	GPM	Less	Linux-PAM
ALSA-Utils	ELFUtils	Grep	LibDRM	Lsof
APR	EncFS	Groff	LibEvdev	Lua
APR-Util	ExFAT	GRUB	LibExif	LuaRocks
...				

/Programs

Multiple versions of a given program can coexist

```
~] ls /Programs/GTK+
2.24.22   2.24.30   3.10.6   3.21.4   Current   Settings

~] ls /Programs/GTK+/2.24.22
bin   doc   include   lib   Resources   share
```

```
~] ls /Programs/GTK+/2.24.22/bin
gtk-builder-convert   gtk-demo   gtk-query-immodules2.0   gtk-update-icon-cache

~] ls /Programs/GTK+/2.24.30/bin
gtk-builder-convert   gtk-demo   gtk-query-immodules2.0   gtk-update-icon-cache
```

/Programs

Easy to tell which files belongs to which packages

```
lucasvr@fedora ~] ls -l /bin/bash
-rwxr-xr-x.   1  root  root  1072008  Sep 30 05:25  /bin/bash
```

```
lucasvr@gobolinux ~] ls -l /bin/bash
lrwxrwxrwx   1  root  root  35  Dec 1 23:42  /bin/bash -> /Programs/Bash/4.4/bin/bash
```

/Programs

Dependency resolution made easy

```
lucasvr@gobolinux ~] cat /Programs/Bluez-ALSA/1.2.0/Resources/Dependencies
Bluez >= 5.47, < 6.0
Glib >= 2.49.5, < 3.0.0
LibBSD >= 0.8.6
ORTP >= 0.20.0
SBC >= 1.3

lucasvr@gobolinux ~] ls /Programs/Bluez
5.47       Current       Settings
```

/Programs

- Listing available packages is trivial
- Removing a program is easy
- Binary and compiled packages are both 1st class citizens
- The whole system becomes intuitive

/System

```
~] ls /System
Aliens    Environment    Index    Kernel    Settings    Tasks
```

Settings: system settings

```
~] ls /System/Settings
acpi          cron.d      fam.conf      gtk-3.0          limits
apt           cups        fonts         hosts            localtime
asciidoc      cupsinit    fstab         ImageMagick-7    login.access
avahi         dbus-1      gconf         inittab          luarocks
...
```

/System

```
~] ls /System
Aliens    Environment    Index    Kernel    Settings    Tasks
```

Index: indexes to entries under /Programs

```
~] ls /System/Index
bin    include    lib    lib64    libexec    sbin    share
```

File System Virtualization with Runner

Runner

- Dynamically changes a process' view of /System/Index
  - Per-process name space (Resources/Dependencies)
  - True conflict resolution
- Container-free virtualization
- Transparent multi-arch support (32/64-bit)

Runner

```
~] cat /Programs/Bash/Current/Resources/Dependencies
Glibc 2.24
Ncurses 6.0
Readline 7.0

~] Runner -v bash
creating new namespace
adding dependency at /Programs/Glibc/2.24
adding dependency at /Programs/Ncurses/6.0
adding dependency at /Programs/Readline/7.0
bash~]
```

Runner

```
bash~] grep overlay /proc/self/mounts | awk {‘print $1 “ “ $2’}
overlay /System/Index/bin
overlay /System/Index/include
overlay /System/Index/lib
overlay /System/Index/libexec
overlay /System/Index/share
```

Runner

Building blocks:

- Per-process (private) namespaces
- OverlayFS

```
unshare(CLONE_NEWNS)
mount(“/System/Index”, MS_PRIVATE)
dependencies = create_dependencies_string(program)
mount(“overlay”, “/System/Index/bin”, dependencies->bin)
mount(“overlay”, “/System/Index/lib”, dependencies->lib)
...
```

Runner

Multi-arch support

```
/Programs/AdobeReader/Current] ./bin/acroread
zsh: no such file or directory: ./bin/acroread

/Programs/AdobeReader/Current] uname -m; file ./bin/acroread
x86_64
./bin/acroread: ELF 32-bit LSB executable, interpreter /lib/ld-linux.so.2

/Programs/AdobeReader/Current] ls -l /lib/ld-linux.so.2
ls: cannot access ‘/lib/ld-linux.so.2’: No such file or directory

/Programs/AdobeReader/Current] Runner ./bin/acroread
[voilà!]
```

Runner

Multi-arch support

```
/Programs/AdobeReader/Current] cat Resources/Architecture
i686

/Programs/AdobeReader/Current] cat Resources/Dependencies
ATK 2.10.0
Cairo 1.12.16
Expat 2.1.0
...

Programs/AdobeReader/Current] cat /Programs/ATK/2.10.0/Resources/Architecture
i686
```

What’s next?

AlienVFS

Pseudo-filesystem for programming language modules

- PIP / PIP3 (Python 2.x, 3.x)
- LuaRocks (Lua)
- CPAN (Perl)
- Gems (Ruby)

AlienVFS

```
~] ls /Mount/AlienVFS
CPAN:Authen::SASL       LuaRocks:luafilesystem   PIP:google-api-python-client
CPAN:Digest::HMAC       LuaRocks:luaposix        PIP:htmlmin
CPAN:Encode::Locale     PIP:agate                PIP:httplib2
CPAN:Error              PIP:agate-dbf            PIP:pyldap

~]  ls /Mount/AlienVFS/PIP:pyldap
2.4.28

~] ls /Mount/AlienVFS/PIP:pyldap/2.4.28
dsml.py   dsml.pyo  _ldap.so       ldapurl.pyc  ldif.py    ldif.pyo
dsml.pyc  ldap      ldapurl.py     ldapurl.pyo  ldif.pyc   pyldap-2.4.28-py2.7.egg-info
```

AlienVFS + /Programs

```
~] ls /Programs
```

ACL	CPAN:LWP	GoboHide	Kmod	LibXSLT
Acpid	CPAN:XML::Parser	Gparted	Lame	Linux
ALSA-Utils	CPAN:XML::Writer	Grep	LibDRM	Lua
APR	ELFUtils	Groff	LibEvdev	LuaRocks
APR-Util	ExFAT	GRUB	LibExif	LuaRocks:luaposix
...				

Related work

Related work

Plan9 From Bell Labs

- 9P remote filesystem protocol
  - File-based operations (read, write, stat, ...)
  - All services communicate through 9P: /proc, /net, /dev/draw, ...
- Per-process namespaces
- Union-mounts everywhere! (“import” command)
- /mips/bin, /i686/bin, ... → /bin

Related work

Docker

- Takes a whole distro as base
- Containers = base + read-write overlay stack
- Pros: fine-grained control over permissions and quotas

Related work

Snap

- Your program and dependencies on a SquashFS image
- Provides isolation and a writeable area

```
~] ls /snap
postgresql

~] ls /snap/postgresql
9.6.1    9.6.5    9.6.6    current

~] ls /snap/postgresql/current
bin      meta     wrappers
```

Related work

Mac OS Bundles

- Directory holding program + resources
  - [Static] executable code
  - Launch images, application icons
  - Info.plist (display name, version number)
  - Settings.bundle (application-specific preferences)
- Does not involve virtualization techniques

Wrapping it up

- Modular system layout eases it all!
- A large collection of uncompressed packages is all you need
- Trivial management
- Not aimed at becoming a full-fledged virtualization tool

Visit us now at
gobolinux.org