

The openSUSE Base System

Richard Günther

Feb. 23th, 2008

Contents

- 1 Motivation
- 2 Base:build
- 3 Base:install
- 4 Base-Bootstrap
- 5 Packaging Advice

Motivation

Care for the core part of openSUSE.

Not colorful, not moving, no sounds.

Hard to “see” any benefit.

Motivation

Care for the core part of openSUSE.

Not colorful, not moving, no sounds.

Hard to “see” any benefit.

Motivation

Care for the core part of openSUSE.

Not colorful, not moving, no sounds.

Hard to “see” any benefit.

Motivation

Still needs to be done.

Base:build

Identify and shrink the set of openSUSE packages that can build itself.

This is one core part of openSUSE.

We are interested in reducing the build time here.

Base:build

Identify and shrink the set of openSUSE packages that can build itself.

This is one core part of openSUSE.

We are interested in reducing the build time here.

Base:build

Identify and shrink the set of openSUSE packages that can build itself.

This is one core part of openSUSE.

We are interested in reducing the build time here.

Overview

Base:build is what you need to (cross-)bootstrap if you want to port openSUSE to a new architecture.

LFS – Linux From Scratch

make that: openSUSE from scratch

Overview

Base:build is what you need to (cross-)bootstrap if you want to port openSUSE to a new architecture.

LFS – Linux From Scratch

make that: openSUSE from scratch

Overview

Base:build is what you need to (cross-)bootstrap if you want to port openSUSE to a new architecture.

LFS – Linux From Scratch

make that: openSUSE from scratch

Overview

Base:build consists of

- 80 source packages
- 164 generated binary packages (i586)
- for some reasons for example tcl, gpm and less.

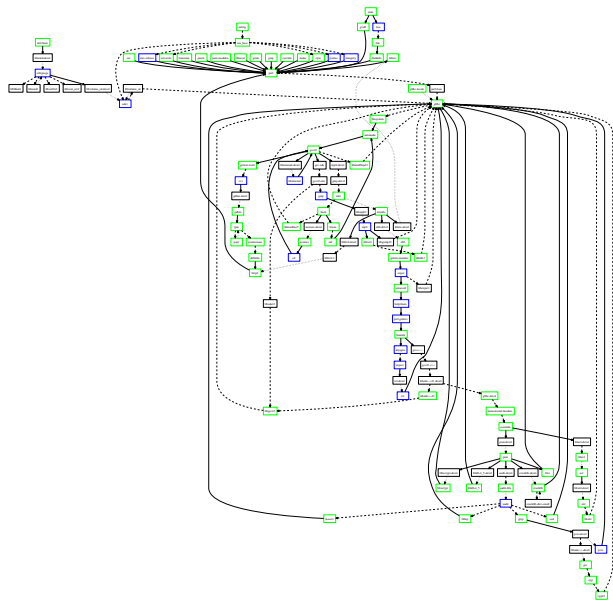
“Bootstraps” itself in half a day in the openSUSE build service.

Overview

Base:build consists of

- 80 source packages
- 164 generated binary packages (i586)
- for some reasons for example tcl, gpm and less.

“Bootstraps” itself in half a day in the openSUSE build service.



Base:install

More difficult to define than Base:build.

Basic requirement: fulfill all rpm dependencies.

What interests most is the size of the installed system.

Base:install

More difficult to define than Base:build.

Basic requirement: fulfill all rpm dependencies.

What interests most is the size of the installed system.

Base:install

More difficult to define than Base:build.

Basic requirement: fulfill all rpm dependencies.

What interests most is the size of the installed system.

Base:install

The minimal set of packages you need to setup a chroot where you can change into and have a shell.

You do not need rpm itself here.

Dependencies of aaa-base should be it.

Base:install

The minimal set of packages you need to setup a chroot where you can change into and have a shell.

You do not need rpm itself here.

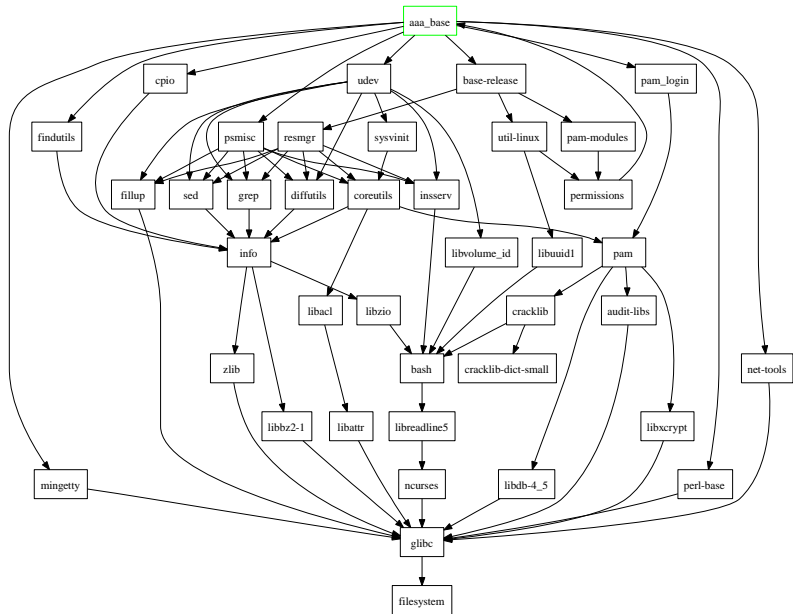
Dependencies of aaa-base should be it.

Base:install

The minimal set of packages you need to setup a chroot where you can change into and have a shell.

You do not need rpm itself here.

Dependencies of aaa-base should be it.



35MB

Base:install

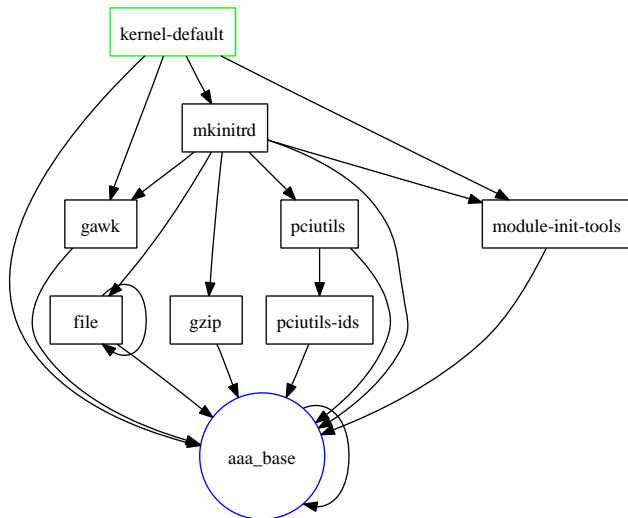
The minimal set of packages you need to setup a bootable image where you can login (and get a shell).

Kernel and initrd might not be required in the image itself, nor does the initrd need to be built inside the image. The same is true for the bootloader.

Base:install

The minimal set of packages you need to setup a bootable image where you can login (and get a shell).

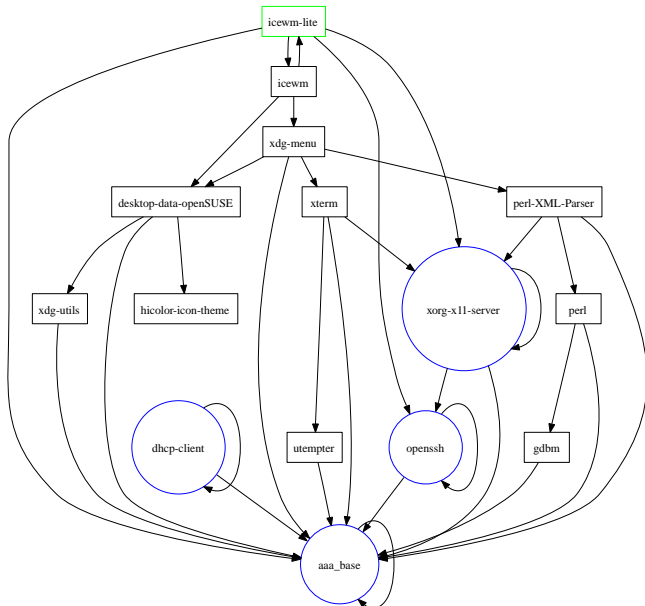
Kernel and initrd might not be required in the image itself, nor does the initrd need to be built inside the image. The same is true for the bootloader.



108MB (+73MB)

What about software installation and network connectivity? Add things like rpm, openssh, dhcp-client and wget so you can start installing additional packages from the network.

Add an X server and xdm to build a X terminal? Add a small window manager?



162MB

Hackweek Project base-bootstrap

<http://en.opensuse.org/Base-Project>

Base:install/base-installer contains yet another image builder (->kiwi).

base-bootstrap: setup the filesystem of the target system (like debootstrap). Seeds the filesystem with all rpm dependencies of a specified set of packages (for example aaa-base). Usable also for QA of package (pre-)requires as it uses a single rpm transaction for setting up the system.

base-installer: in addition to base-bootstrap includes building a bootable image including initrd building and bootloader installing (support for QEMU is implemented).

Hackweek Project base-bootstrap

<http://en.opensuse.org/Base-Project>

Base:install/base-installer contains yet another image builder (->kiwi).

base-bootstrap: setup the filesystem of the target system (like debootstrap). Seeds the filesystem with all rpm dependencies of a specified set of packages (for example aaa-base). Usable also for QA of package (pre-)requires as it uses a single rpm transaction for setting up the system.

base-installer: in addition to base-bootstrap includes building a bootable image including initrd building and bootloader installing (support for QEMU is implemented).

Hackweek Project base-bootstrap

<http://en.opensuse.org/Base-Project>

Base:install/base-installer contains yet another image builder (->kiwi).

base-bootstrap: setup the filesystem of the target system (like debootstrap). Seeds the filesystem with all rpm dependencies of a specified set of packages (for example aaa-base). Usable also for QA of package (pre-)requires as it uses a single rpm transaction for setting up the system.

base-installer: in addition to base-bootstrap includes building a bootable image including initrd building and bootloader installing (support for QEMU is implemented).

Hackweek Project base-bootstrap

<http://en.opensuse.org/Base-Project>

Base:install/base-installer contains yet another image builder (->kiwi).

base-bootstrap: setup the filesystem of the target system (like debootstrap). Seeds the filesystem with all rpm dependencies of a specified set of packages (for example aaa-base). Usable also for QA of package (pre-)requires as it uses a single rpm transaction for setting up the system.

base-installer: in addition to base-bootstrap includes building a bootable image including initrd building and bootloader installing (support for QEMU is implemented).

Packaging Rules

- Use file-requires to allow substitution with for example busybox.
- Prefer C and sh (maybe perl) for core functionality.
- Avoid %post scripts in core packages.
- Split packages appropriately.
- Follow the shared library policy.
- Do not provide excessive default configuration, but defer this to product policy packages.

Packaging Rules

- Use file-requires to allow substitution with for example busybox.
- Prefer C and sh (maybe perl) for core functionality.
- Avoid %post scripts in core packages.
- Split packages appropriately.
- Follow the shared library policy.
- Do not provide excessive default configuration, but defer this to product policy packages.

Packaging Rules

- Use file-requires to allow substitution with for example busybox.
- Prefer C and sh (maybe perl) for core functionality.
- Avoid %post scripts in core packages.
- Split packages appropriately.
- Follow the shared library policy.
- Do not provide excessive default configuration, but defer this to product policy packages.

Packaging Rules

- Use file-requires to allow substitution with for example busybox.
- Prefer C and sh (maybe perl) for core functionality.
- Avoid %post scripts in core packages.
- Split packages appropriately.
- Follow the shared library policy.
- Do not provide excessive default configuration, but defer this to product policy packages.

Packaging Rules

- Use file-requires to allow substitution with for example busybox.
- Prefer C and sh (maybe perl) for core functionality.
- Avoid %post scripts in core packages.
- Split packages appropriately.
- Follow the shared library policy.
- Do not provide excessive default configuration, but defer this to product policy packages.

Packaging Rules

- Use file-requires to allow substitution with for example busybox.
- Prefer C and sh (maybe perl) for core functionality.
- Avoid %post scripts in core packages.
- Split packages appropriately.
- Follow the shared library policy.
- Do not provide excessive default configuration, but defer this to product policy packages.

Questions?