Recipes for sabayon

cook your own Linux distro within two hours

Danilo Pianini
danilo.pianini@unibo.it

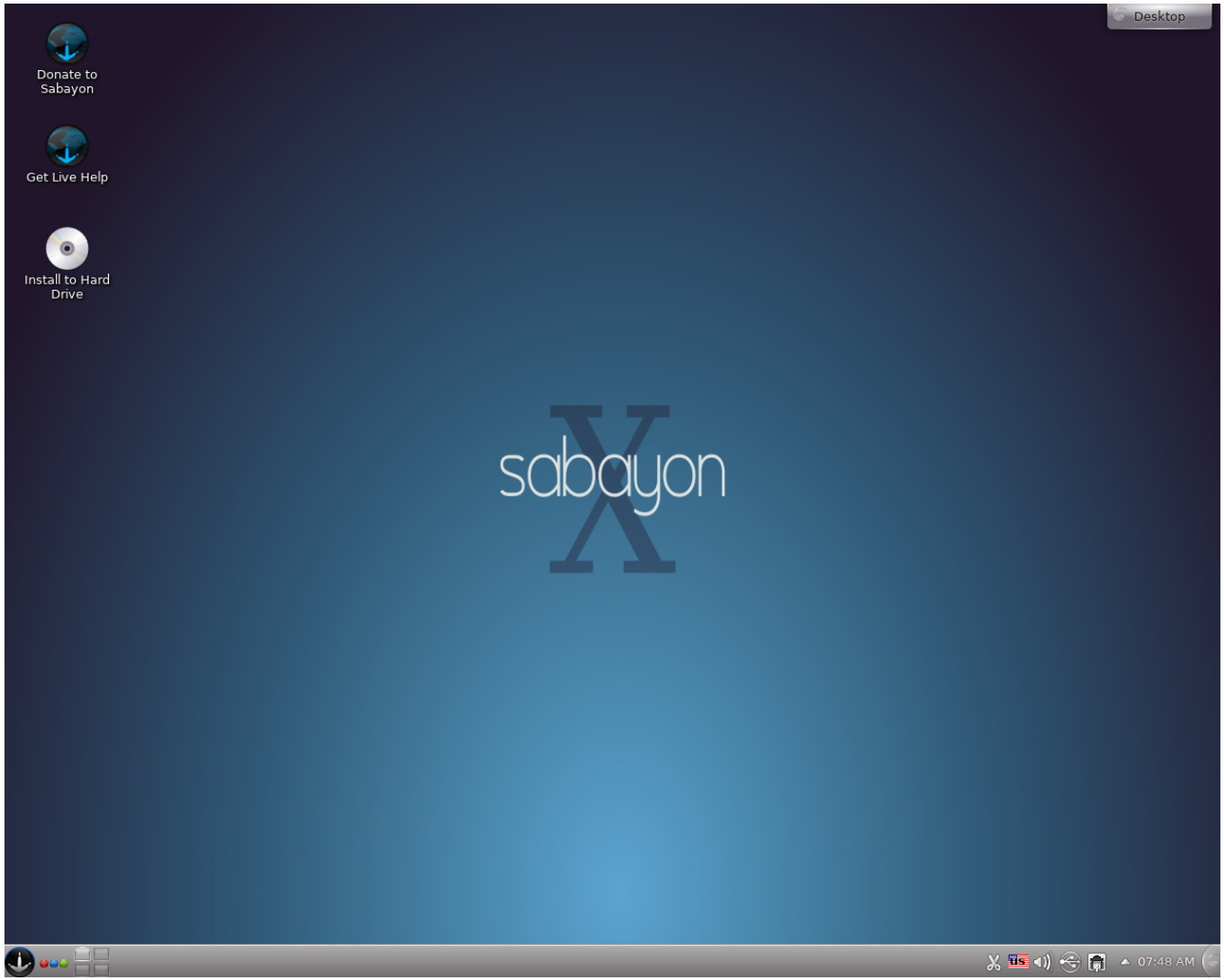
Linux Day 2012, Cesena

October 27, 2012

Main features

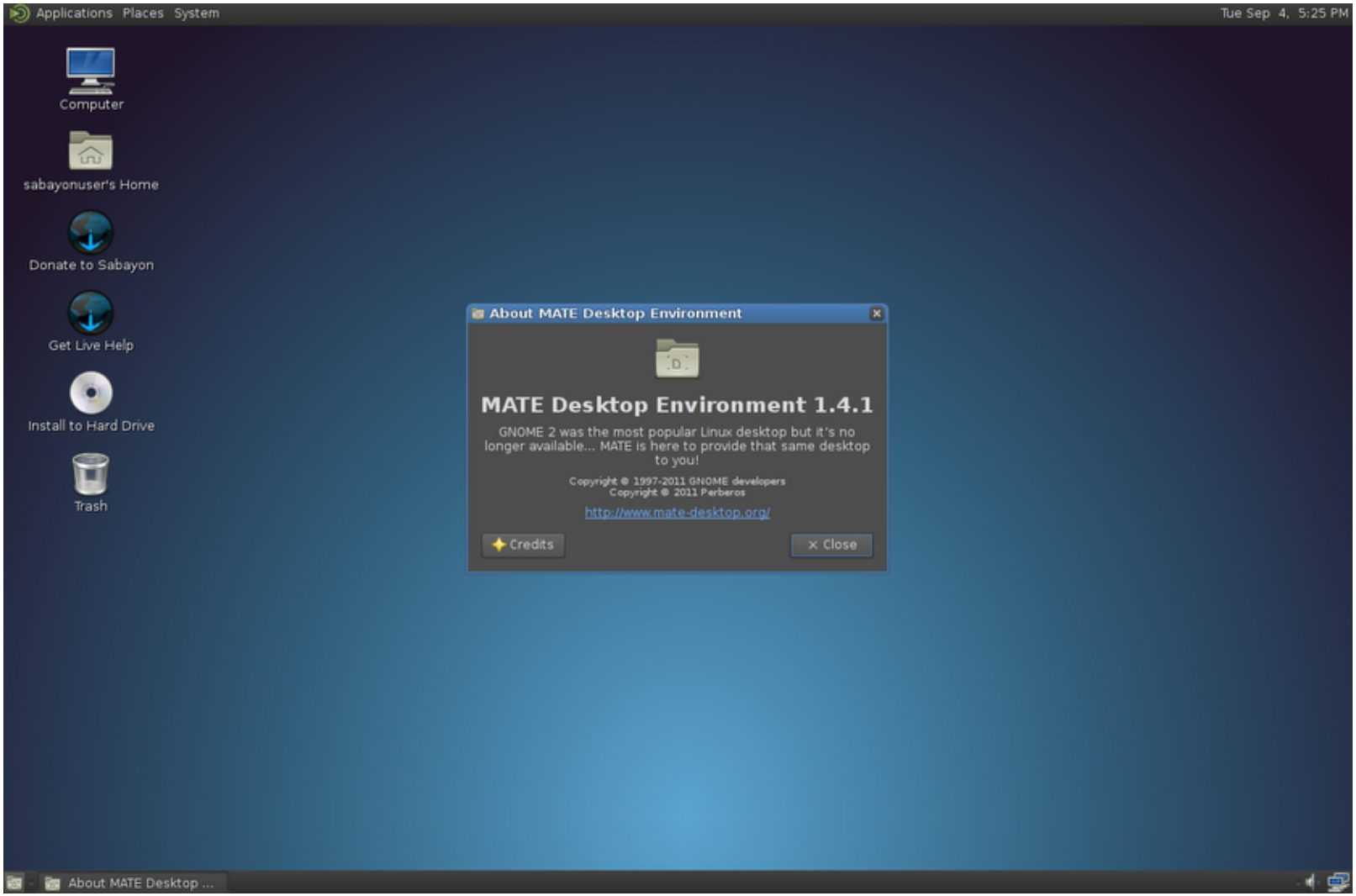
- Built upon Gentoo
- 100% Gentoo compatible
- Rolling release
- Proprietary drivers out of the box
- Short release circle
- Two package managers
- Weekly updates for everybody
- Daily updates for the skilled
- Experimental hourly updates for the advanced
- New applications hit the repo in short time
- Daily releases!

Desktop side (KDE)

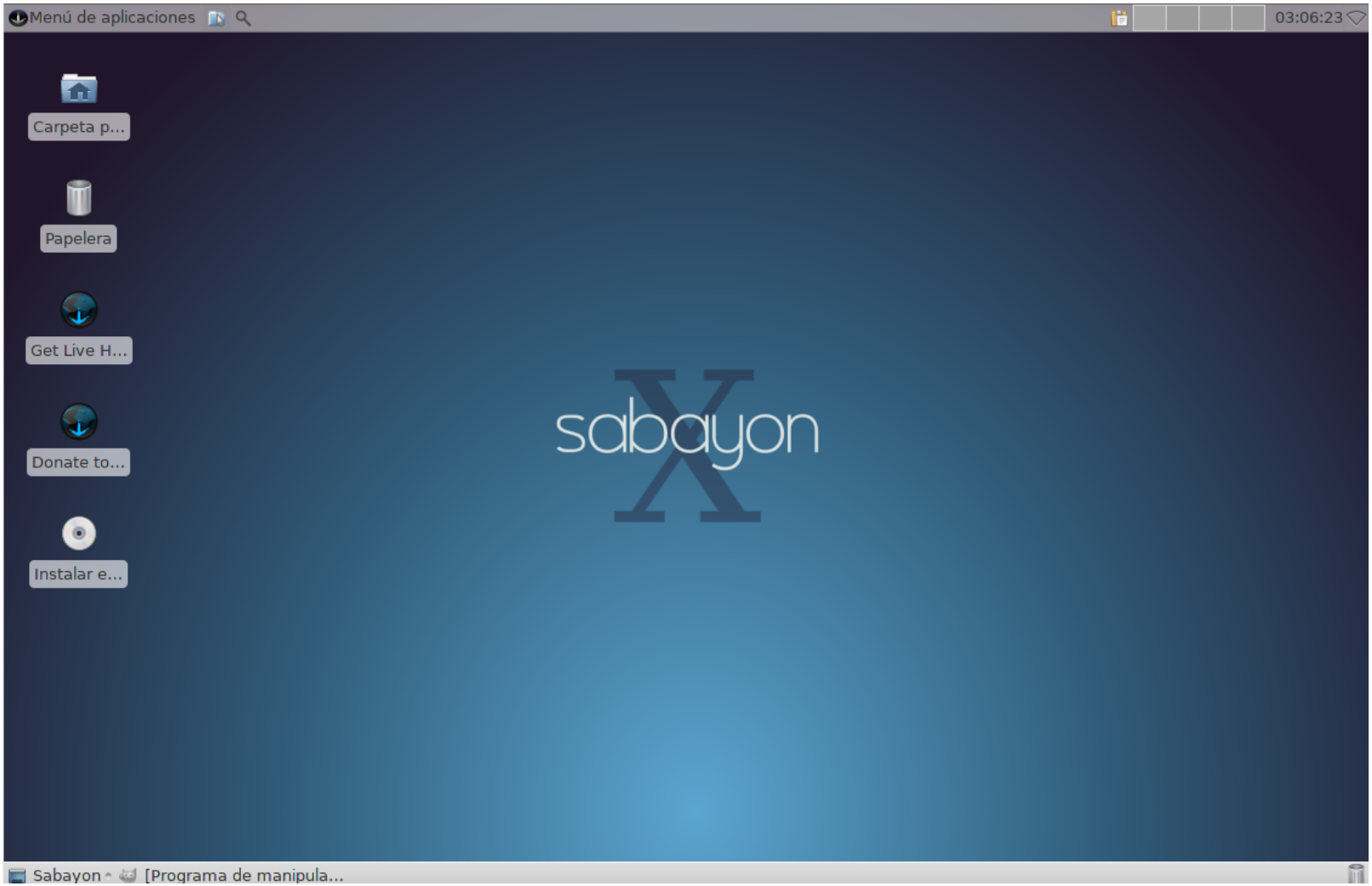


Desktop side (Gnome)

Desktop side (MATE)



Desktop side (XFCE)



Server side

Server Base

- Base system, optimized for server load
- No support for graphics
- Most of the Sabayon Project infrastructure runs on Sabayon Server
- A simulation server in APICe runs Server Base
- The Maven webserver in APICe runs Server Base

Hardened Server

- Built upon the Gentoo Hardened patch set
- Same as above, with strong focus on security

Distro developer side

Spin Base

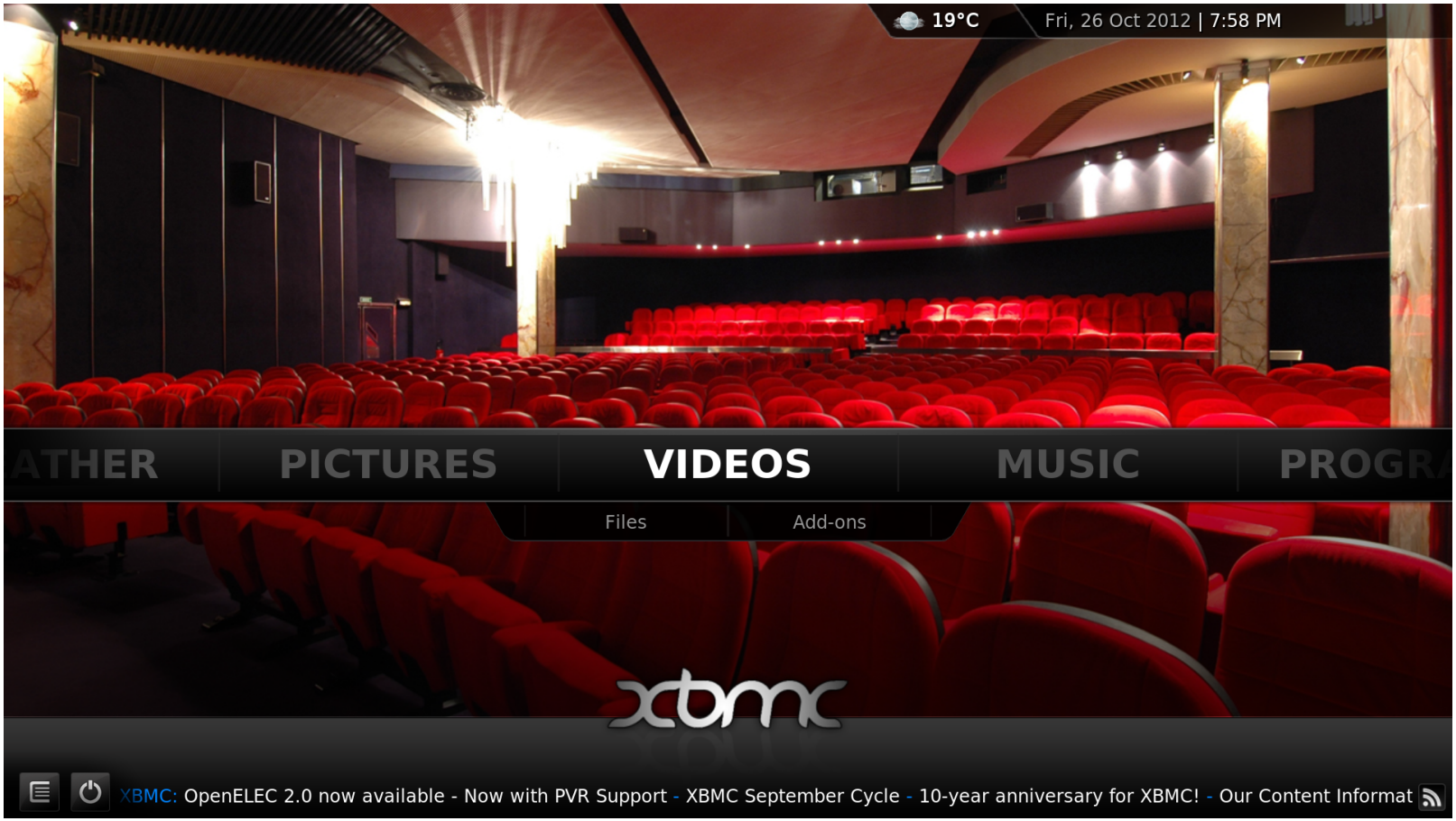
- Base system
- Desktop optimized kernel
- Ready for graphical support
- Base for all the other releases

LXDE

- Similar to XFCE
- Not officially released, but fully usable

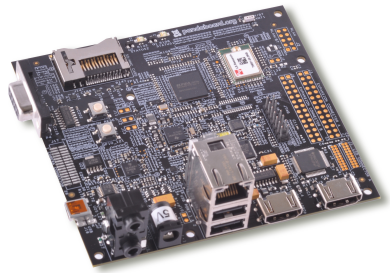
Multimedia

KDE and GNOME editions integrate XBMC, the system can be booted in mediacenter mode



ARM





ARMv7a support

- BeagleBoard xM (4GB and 8GB)
- BeagleBone (2GB, 4GB, 8GB, 16GB)
- Efika MX (4GB)
- Pandaboard (4GB, 8GB)

Why Gentoo?

Of thousands distros out there, only few are based on Gentoo

Pros

- Modularity
- Portability
- Extreme flexibility
- Comprehensive documentation

Cons

- Every package must be compiled
- Complex
- Remarkable skill level required to keep the system up to date

Early times

- A ready-to-use Gentoo
- Huge DVD (more than 4GB)
- Tons of pre-installed software
- Pre-configured drivers

Notably: first distro to provide Compiz support out of the box

A binary package manager

Problem: upgrading was a hell

Entropy

- Relies on a source package manager (Portage)
- Builds binary packages
- Fast
- Potentially, same packages of the SPM
- Upgrades become easy!

Rolling!

Automatization!

Now that a new package manager is on the way...

- Automated upgrades of non-critical packages
- Automated generation of Sabayon ISOs (daily builds)
- Multiple repositories in order to improve reliability
- Most work is completely automatic

What is Molecule?

Release metatool used for creating Sabayon (and Gentoo) releases

- Basically an automation tool
- Multiple profiles built-in
  - Use a Sabayon ISO as base for a new distro
  - Build a new distro from scratch
- Easy addition and removal of software
- Entry point for configuration
- Documentation is poor...
- ...but all the scripts used by the Sabayon team are available online

Getting started

- Install molecule
  - `equo install molecule`
- Download the examples
  - `git://git.sabayon.org/projects/molecules.git`

Photoyon Linux

Let's build a photographers oriented distro upon the normal Sabayon KDE!

- KDE environment
- Digikam
- GQView
- RAW Therapee
- Hugin
- GIMP
- Inkscape
- Kdenlive
- Google Picasa

Note: it would make more sense to start from Spin Base, but the files end up being a bit too long for being presented here!

Spec file

Let's take clue from the amd64 KDE spec file

```
%env %import PATH/TO/kde.common

release_version: 1.0

release_desc: amd64

%env source_iso: /PATH/TO/Sabayon_Linux_DAILY_amd64_K.iso

destination_iso_image_name: Photoyon_1.0_amd64.iso
```

kde.common

```
execution_strategy: iso_remaster

release_string: Photoyon Linux
release_file: /etc/sabayon-edition
iso_title: Photoyon 1.0
extra_mksquashfs_parameters: -comp xz -Xbcj x86

# Outer chroot script command - BEFORE
%env outer_chroot_script: /PATH/TO/scripts/remaster_pre.sh
# Inner chroot script - BEFORE
%env inner_chroot_script: /PATH/TO/scripts/remaster_generic_inner_chroo
# Inner chroot script command - AFTER
%env inner_chroot_script_after: /PATH/TO/scripts/remaster_generic_inner
# Outer chroot script command -AFTER
%env outer_chroot_script_after: /PATH/TO/scripts/remaster_post.sh

destination_iso_directory: /PATH/TO/OUTPUT/FOLDER/

...
```

kde.common

```
...
# List of packages that would be removed from chrooted system
packages_to_remove:
app-misc/sabayon-mce,
kde-base/libkdegames,
kde-base/kontact
kde-base/kopete,
kde-base/klipper,
kde-base/korganizer,
media-sound/clementine,
media-tv/xbmc,
net-irc/konversation,
net-libs/libktorrent
...
```

kde.common

```
...
# List of packages that would be added to chrooted system
packages_to_add:
media-gfx/digikam
media-gfx/gimp,
media-gfx/gqview,
media-gfx/hugin,
media-gfx/inkscape,
media-gfx/picasa,
media-gfx/rawtherapee,
media-video/kdenlive
...
```

remaster_pre

```
#!/bin/sh
PKGS_DIR="/var/lib/entropy/client/packages"
CHROOT_PKGS_DIR="${CHROOT_DIR}/var/lib/entropy/client/packages"

[[ ! -d "${PKGS_DIR}" ]] && mkdir -p "${PKGS_DIR}"
[[ ! -d "${CHROOT_PKGS_DIR}" ]] && mkdir -p "${CHROOT_PKGS_DIR}"

echo "Mounting packages over"
rm -rf "${CHROOT_PKGS_DIR}"/*
echo "Mounting bind to ${CHROOT_PKGS_DIR}"
mount --bind "${PKGS_DIR}" "${CHROOT_PKGS_DIR}" || exit 1
mount -t proc proc "${CHROOT_DIR}/proc"

exit 0
```

inner scripts

Inner scripts configure the various services
Modify them only if you need something different in your rc

Edit a skel

- The defaults for a new user are stored in `/etc/skel`
- You can copy them, then add new configurations, or modify the existing.
- New users will inherit your defaults, if you pull them within the ISO (use the remaster_post script to do it!)

Raw but effective way

- Create a first revision of your distro with the default skel
- Start the applications whose defaults are not satisfying
- Set them up as you wish
- Close the applications
- Copy the configuration within your personalised skel

remaster_post

```
#!/bin/sh
PKGS_DIR="/var/lib/entropy/client/packages"
CHROOT_PKGS_DIR="${CHROOT_DIR}/var/lib/entropy/client/packages"
CHROOT_SKEL_DIR="${CHROOT_DIR}/etc/skel"
SKEL_DIR="/PATH/TO/YOUR/skel"

rm -rf CHROOT_SKEL_DIR
cp -r SKEL_DIR CHROOT_SKEL_DIR

# make sure to not leak /proc
umount "${CHROOT_DIR}/proc" &> /dev/null
umount "${CHROOT_DIR}/proc" &> /dev/null
umount "${CHROOT_DIR}/proc" &> /dev/null

echo "Umounting bind to ${CHROOT_PKGS_DIR}"
umount "${CHROOT_PKGS_DIR}" || exit 1

exit 0
```

Sabayon Forensics

Grusha Linux

Final thoughts

Sabayon

- Rolling release
- Wide spectrum of applications
- Daily relases
- Wide repositories
- Many flavors
- Gentoo compatibility

Molecule

- Allows to create a new distro in a couple of hours
- Automatizes the creation process
- Little scripting skills required
- Ideal for small-team distro development

Join the team!

- The team counts about twenty people
- Different nationalities (English is a requirement)
- They normally hang on IRC (`#sabayon` and `#sabayon-dev`)
- Very collaborative environment
- If you want to improve the distribution:
  - Learn: `ssh`, `screen`, `rsync`, `Portage`, `layman`
  - Ask for joining the team
  - You will receive serverside-Entropy training
  - You will be among those who forge Sabayon

Recipes for sabayon

cook your own Linux distro within two hours

Danilo Pianini
danilo.pianini@unibo.it

Linux Day 2012, Cesena

October 27, 2012