Can we build a simple, cross-distribution installation framework?

Dieter Plaetinck

06-02-2011

TOC

- Why a shellscript based installation framework?
- AIF lessons learned
- Cross-distro collaboration?

AIF Overview

- a set of bash scripts to install Arch Linux

AIF Overview

- a set of bash scripts to install Arch Linux
- framework: procedures and libraries

AIF Overview

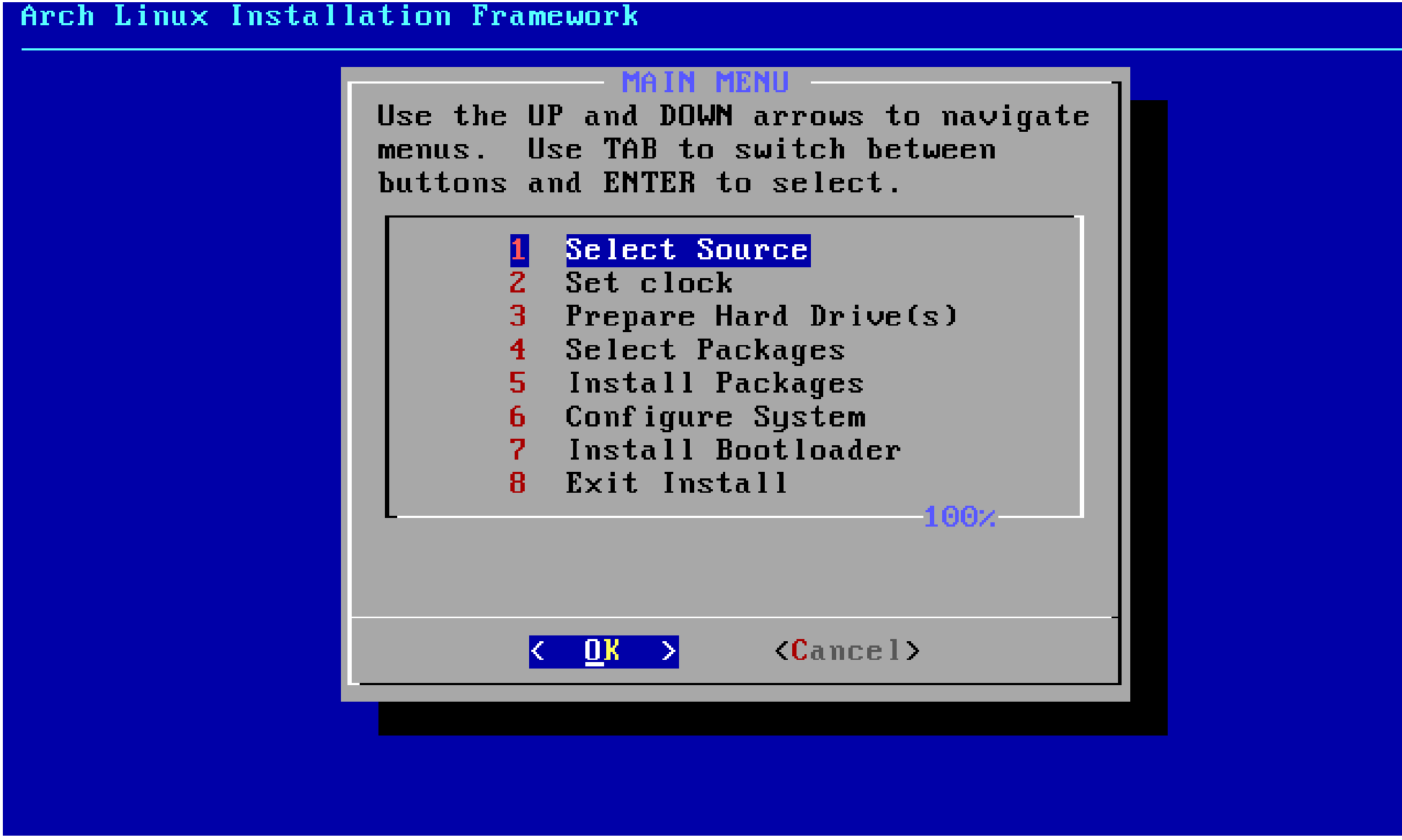
- a set of bash scripts to install Arch Linux
- framework: procedures and libraries
- customizable

AIF Overview

- a set of bash scripts to install Arch Linux
- framework: procedures and libraries
- customizable
- a filesystem model

AIF Overview

- a set of bash scripts to install Arch Linux
- framework: procedures and libraries
- customizable
- a filesystem model
- testsuite (build time and runtime)


Arch Linux Installation Framework
MAIN MENU
Use the UP and DOWN arrows to navigate menus. Use TAB to switch between buttons and ENTER to select.
1 Select Source
2 Set clock
3 Prepare Hard Drive(s)
4 Select Packages
5 Install Packages
6 Configure System
7 Install Bootloader
8 Exit Install
100%
< OK >
<Cancel>

Why?

- simple, hackable framework. same code for interactive and automatic installs
- KISS versus handholding users? (interactive procedure)

Why?

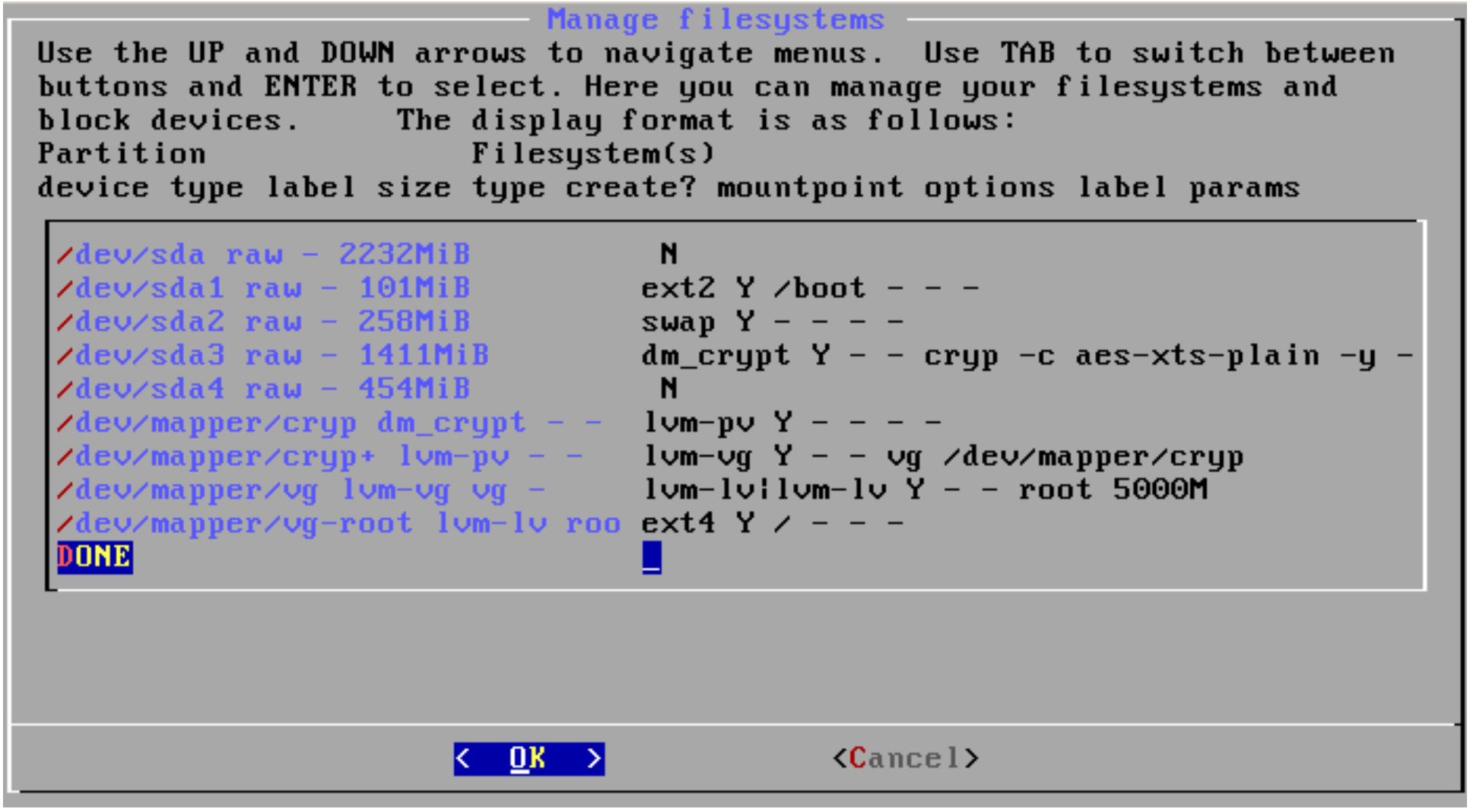
- simple, hackable framework. same code for interactive and automatic installs
- KISS versus handholding users? (interactive procedure)
  - required knowledge is different from running the OS
  - fast and {worry,concentration}less
  - casual users

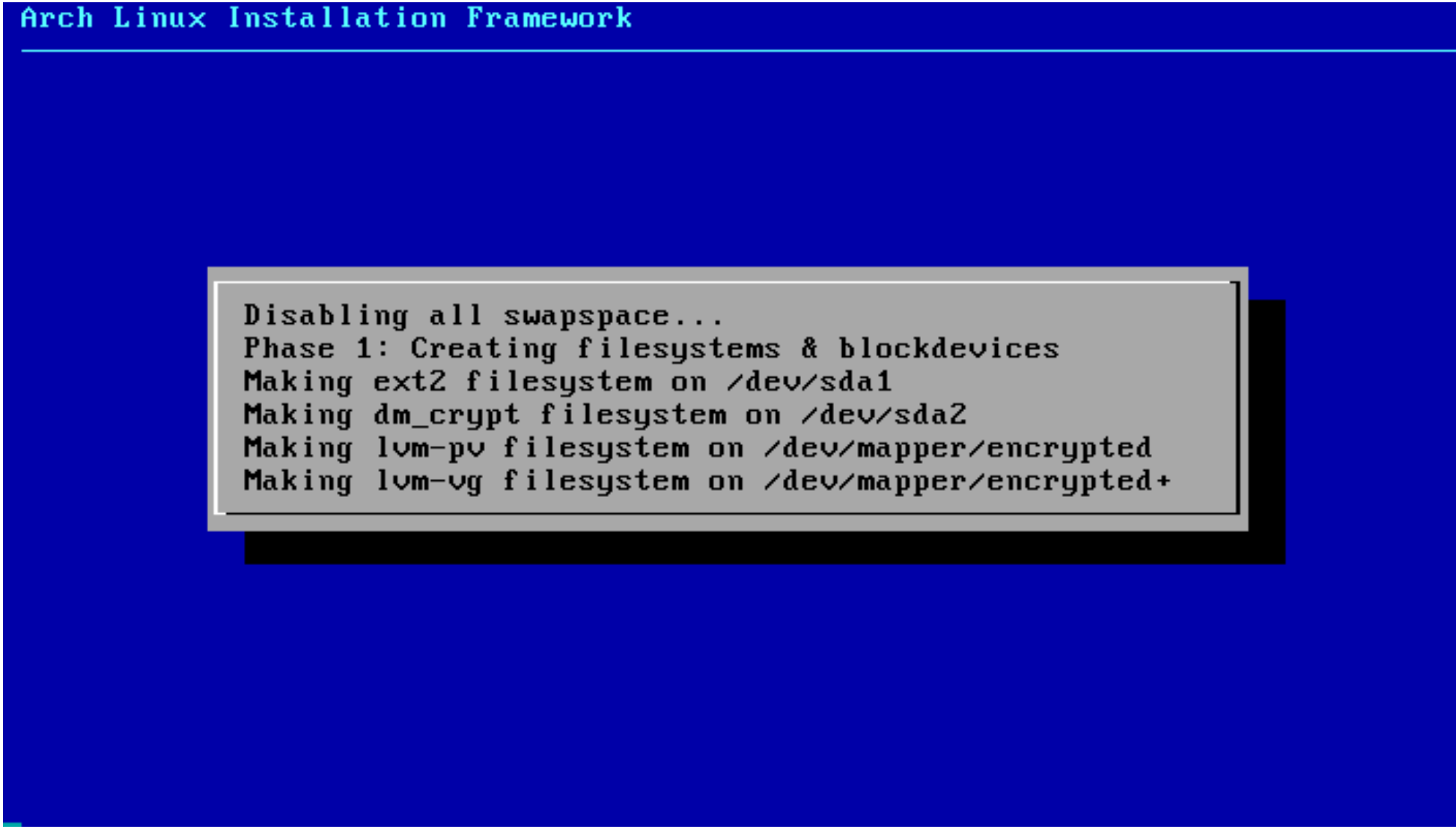
Dependencies

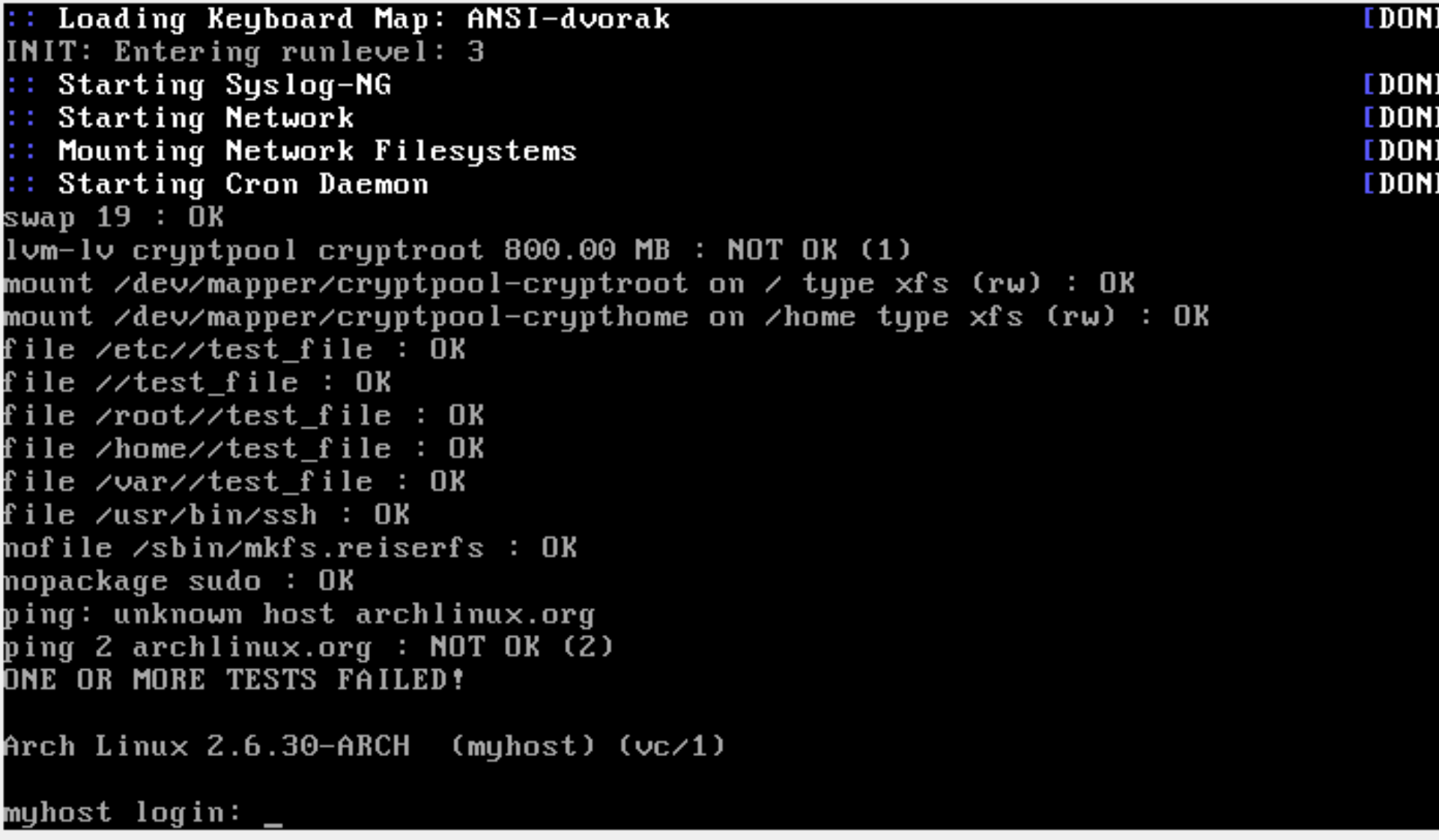
- Needed: bash, util-linux-ng, coreutils, grep, awk, sed, libui-sh
- Arch: pacman
- Optional: cryptsetup, lvm2, e2fsprogs, jfsutils, reiserfsprogs, xfsprogs, nilfs-utils
- Optional: dhcpd, ntp, dialog, grub, syslinux, markdown

Libui-sh

```
source /usr/lib/libui.sh

ask_yesno 'do you want to continue?' || return

ask_number 'Filesystem size' $lower $upper \
$default && echo picked $ANSWER_NUMBER

ask_checklist "Select packages" "${pkglist[@]}"
check_is_in openssh "${ANSWER_CHECKLIST[@]}"

ask_option no $title "Select an entry" required \
"${FSOPTS[@]}"
```


Manage filesystems
Use the UP and DOWN arrows to navigate menus. Use TAB to switch between buttons and ENTER to select. Here you can manage your filesystems and block devices. The display format is as follows:
Partition Filesystem(s)
device type label size type create? mountpoint options label params
/dev/sda raw - 2232MiB N
/dev/sda1 raw - 101MiB ext2 Y /boot - - -
/dev/sda2 raw - 258MiB swap Y - - - -
/dev/sda3 raw - 1411MiB dm_crypt Y - - cryp -c aes-xts-plain -y -
/dev/sda4 raw - 454MiB N
/dev/mapper/cryp dm_crypt - - lvm-pv Y - - - -
/dev/mapper/cryp+ lvm-pv - - lvm-vg Y - - vg /dev/mapper/cryp
/dev/mapper/vg lvm-vg vg - lvm-lv|lvm-lv Y - - root 5000M
/dev/mapper/vg-root lvm-lv roo ext4 Y / - - -
DONE
< OK >
<Cancel>


Arch Linux Installation Framework
Disabling all swapspace...
Phase 1: Creating filesystems & blockdevices
Making ext2 filesystem on /dev/sda1
Making dm_crypt filesystem on /dev/sda2
Making lvm-pv filesystem on /dev/mapper/encrypted
Making lvm-vg filesystem on /dev/mapper/encrypted+


:: Loading Keyboard Map: ANSI-dvorak [DONE
INIT: Entering runlevel: 3
:: Starting Syslog-NG [DONE
:: Starting Network [DONE
:: Mounting Network Filesystems [DONE
:: Starting Cron Daemon [DONE
swap 19 : OK
lvm-lv cryptpool cryptroot 800.00 MB : NOT OK (1)
mount /dev/mapper/cryptpool-cryptroot on / type xfs (rw) : OK
mount /dev/mapper/cryptpool-crypthome on /home type xfs (rw) : OK
file /etc//test_file : OK
file //test_file : OK
file /root//test_file : OK
file /home//test_file : OK
file /var//test_file : OK
file /usr/bin/ssh : OK
nofile /sbin/mkfs.reiserfs : OK
nopackage sudo : OK
ping: unknown host archlinux.org
ping 2 archlinux.org : NOT OK (2)
ONE OR MORE TESTS FAILED!
Arch Linux 2.6.30-ARCH (myhost) (vc/1)
myhost login: _

Lessons learned

- bash is limited
- filesystem model code is hard

Lessons learned

- bash is limited
- filesystem model code is hard
- grub is hard

Lessons learned

- bash is limited
- filesystem model code is hard
- grub is hard
- date-time setting is hard

AIF Missing features / Goals

- GPT/EFI/large block sizes
- btrfs
- Softraid

LIF: distro independent

Distro independent (mostly)

- UI

LIF: distro independent

Distro independent (mostly)

- UI
- filesystems

LIF: distro independent

Distro independent (mostly)

- UI
- filesystems
- bootloaders

LIF: distro independent

Distro independent (mostly)

- UI
- filesystems
- bootloaders
- autoinstall configs, blueprinting

LIF: distro independent

Distro independent (mostly)

- UI
- filesystems
- bootloaders
- autoinstall configs, blueprinting
- time/date setting

LIF: distro independent

Distro independent (mostly)

- UI
- filesystems
- bootloaders
- autoinstall configs, blueprinting
- time/date setting
- error handling/reporting/logging

LIF: distro dependent

Distro dependent (mostly)

- source selection

LIF: distro dependent

Distro dependent (mostly)

- source selection
- package selection

LIF: distro dependent

Distro dependent (mostly)

- source selection
- package selection
- package installation

LIF: distro dependent

Distro dependent (mostly)

- source selection
- package selection
- package installation
- configuration

From AIF to LIF

- plugins/libraries per distro
- branches per distro

questions?

ML: arch-releng

dieter@plaetinck.be

https://github.com/Dieterbe/aif

http://projects.archlinux.org/aif.git/