Docker: LAB 4 image de base ArchLinux

Objet	Création d'une image de base ArchLinux
Niveau requis	débutant, avisé
Débutant, à savoir	
Suivi	:DRAFT:

Création d'une image de base `Archlinux`

moby/contrib/mkimage-arch.sh

```
#!/usr/bin/env bash
# Generate a minimal filesystem for archlinux and load it into the local
# docker as "archlinux"
# requires root
set -e

# reset umask to default
umask 022

hash pacstrap &>/dev/null || {
    echo "Could not find pacstrap. Run pacman -S arch-install-scripts"
    exit 1
}

hash expect &>/dev/null || {
    echo "Could not find expect. Run pacman -S expect"
    exit 1
}

export LANG="C.UTF-8"

ROOTFS=$(mktemp -d ${TMPDIR:-/var/tmp}/rootfs-archlinux-XXXXXXXXXX)
chmod 755 $ROOTFS

# required packages
PKGREQUIRED=(
    bash
    haveged
    pacman
    pacman-mirrorlist
)

# packages to ignore for space savings
PKGIGNORE=(
    dhcpcd
    diffutils
    file
```

```
    inetutils
    iproute2
    iputils
    jfsutils
    licenses
    linux
    linux-firmware
    lvm2
    man-db
    man-pages
    mdadm
    nano
    netctl
    openresolv
    pciutils
    pcmciautils
    psmisc
    reiserfsprogs
    s-nail
    sysfsutils
    systemd-sysvcompat
    usbutils
    vi
    which
    xfsprogs
)

PKGREMOVE=(
    gawk
    haveged
    less
    linux-libre
    linux-libre-firmware
)

PKGREQUIRED="${PKGREQUIRED[*]}"
IFS=','
PKGIGNORE="${PKGIGNORE[*]}"
unset IFS
PKGREMOVE="${PKGREMOVE[*]}"

arch="$(uname -m)"
case "$arch" in
    armv*)
        if pacman -Q archlinuxarm-keyring >/dev/null 2>&1; then
            pacman-key --init
            pacman-key --populate archlinuxarm
        else
            echo "Could not find archlinuxarm-keyring. Please, install it
and run pacman-key --populate archlinuxarm"
            exit 1
```

```
        fi
        PACMAN_CONF=$(mktemp ${TMPDIR:-/var/tmp}/pacman-conf-archlinux-
XXXXXXXXX)
        version="$(echo $arch | cut -c 5)"
        sed "s/Architecture = armv/Architecture = armv${version}h/g"
'./mkimage-archarm-pacman.conf' > "${PACMAN_CONF}"
        PACMAN_MIRRORLIST='Server =
http://mirror.archlinuxarm.org/$arch/$repo'
        PACMAN_EXTRA_PKGS='archlinuxarm-keyring'
        EXPECT_TIMEOUT=1800 # Most armv* based devices can be very slow
(e.g. RPiv1)
        ARCH_KEYRING=archlinuxarm
        DOCKER_IMAGE_NAME="armv${version}h/archlinux"
        ;;
    *)
        PACMAN_CONF='./mkimage-arch-pacman.conf'
        PACMAN_MIRRORLIST='Server =
https://mirrors.kernel.org/archlinux/$repo/os/$arch'
        PACMAN_EXTRA_PKGS=''
        EXPECT_TIMEOUT=60
        ARCH_KEYRING=archlinux
        DOCKER_IMAGE_NAME=archlinux
        ;;
esac

export PACMAN_MIRRORLIST

expect <<EOF
    set send_slow {1 .1}
    proc send {ignore arg} {
        sleep .1
        exp_send -s -- \$arg
    }
    set timeout $EXPECT_TIMEOUT
    spawn pacstrap -C $PACMAN_CONF -c -d -G -i $ROOTFS base $PKGREQUIRED
$PACMAN_EXTRA_PKGS --ignore $PKGIGNORE
    expect {
        -exact "anyway? \[Y/n\] " { send -- "n\r"; exp_continue }
        -exact "(default=all): " { send -- "\r"; exp_continue }
        -exact "installation? \[Y/n\]" { send -- "y\r"; exp_continue }
        -exact "delete it? \[Y/n\]" { send -- "y\r"; exp_continue }
    }
EOF

arch-chroot $ROOTFS /bin/sh -c 'rm -r /usr/share/man/*'
arch-chroot $ROOTFS /bin/sh -c "haveged -w 1024; pacman-key --init; pkill
haveged; pacman-key --populate $ARCH_KEYRING"
arch-chroot $ROOTFS /bin/sh -c "ln -sf /usr/share/zoneinfo/UTC
/etc/localtime"
arch-chroot $ROOTFS /bin/sh -c "for pkg in $PKGREMOVE; do if pacman -Qi
\$pkg > /dev/null 2>&1; then pacman -Rs --noconfirm \$pkg; fi; done"
```

```
echo 'en_US.UTF-8 UTF-8' > $ROOTFS/etc/locale.gen
arch-chroot $ROOTFS locale-gen

# udev doesn't work in containers, rebuild /dev
DEV=$ROOTFS/dev
rm -rf $DEV
mkdir -p $DEV
mknod -m 666 $DEV/null c 1 3
mknod -m 666 $DEV/zero c 1 5
mknod -m 666 $DEV/random c 1 8
mknod -m 666 $DEV/urandom c 1 9
mkdir -m 755 $DEV/pts
mkdir -m 1777 $DEV/shm
mknod -m 666 $DEV/tty c 5 0
mknod -m 600 $DEV/console c 5 1
mknod -m 666 $DEV/tty0 c 4 0
mknod -m 666 $DEV/full c 1 7
mknod -m 600 $DEV/initctl p
mknod -m 666 $DEV/ptmx c 5 2
ln -sf /proc/self/fd $DEV/fd

tar --numeric-owner --xattrs --acls -C $ROOTFS -c . | docker import -
$DOCKER_IMAGE_NAME
docker run --rm -t $DOCKER_IMAGE_NAME echo Success.
rm -rf $ROOTFS
```

moby/contrib/mkimage-arch-pacman.conf

```
#
# /etc/pacman.conf
#
# See the pacman.conf(5) manpage for option and repository directives

#
# GENERAL OPTIONS
#
[options]
# The following paths are commented out with their default values listed.
# If you wish to use different paths, uncomment and update the paths.
#RootDir     = /
#DBPath      = /var/lib/pacman/
#CacheDir    = /var/cache/pacman/pkg/
#LogFile     = /var/log/pacman.log
#GPGDir      = /etc/pacman.d/gnupg/
HoldPkg     = pacman glibc
#XferCommand = /usr/bin/curl -C - -f %u > %o
#XferCommand = /usr/bin/wget --passive-ftp -c -O %o %u
#CleanMethod = KeepInstalled
#UseDelta    = 0.7
```

```
Architecture = auto

# Pacman won't upgrade packages listed in IgnorePkg and members of
IgnoreGroup
#IgnorePkg   =
#IgnoreGroup =

#NoUpgrade   =
#NoExtract   =

# Misc options
#UseSyslog
#Color
#TotalDownload
# We cannot check disk space from within a chroot environment
#CheckSpace
#VerbosePkgLists

# By default, pacman accepts packages signed by keys that its local keyring
# trusts (see pacman-key and its man page), as well as unsigned packages.
SigLevel    = Required DatabaseOptional
LocalFileSigLevel = Optional
#RemoteFileSigLevel = Required

# NOTE: You must run `pacman-key --init` before first using pacman; the
local
# keyring can then be populated with the keys of all official Arch Linux
# packagers with `pacman-key --populate archlinux`.

#
# REPOSITORIES
#   - can be defined here or included from another file
#   - pacman will search repositories in the order defined here
#   - local/custom mirrors can be added here or in separate files
#   - repositories listed first will take precedence when packages
#     have identical names, regardless of version number
#   - URLs will have $repo replaced by the name of the current repo
#   - URLs will have $arch replaced by the name of the architecture
#
# Repository entries are of the format:
#       [repo-name]
#       Server = ServerName
#       Include = IncludePath
#
# The header [repo-name] is crucial - it must be present and
# uncommented to enable the repo.
#

# The testing repositories are disabled by default. To enable, uncomment the
# repo name header and Include lines. You can add preferred servers
immediately
```

```
# after the header, and they will be used before the default mirrors.

#[testing]
#Include = /etc/pacman.d/mirrorlist

[core]
Include = /etc/pacman.d/mirrorlist

[extra]
Include = /etc/pacman.d/mirrorlist

#[community-testing]
#Include = /etc/pacman.d/mirrorlist

[community]
Include = /etc/pacman.d/mirrorlist

# An example of a custom package repository.  See the pacman manpage for
# tips on creating your own repositories.
#[custom]
#SigLevel = Optional TrustAll
#Server = file:///home/custompkgs
```