

```
#!/bin/sh

umask 0022
export PATH='/usr/bin:/sbin:/bin'

# Defaults
keep="n"
CONFDIR="/etc/initramfs-tools"
verbose="n"
test -e /bin/busybox && BUSYBOXDIR=/bin
test -e /usr/lib/initramfs-tools/bin/busybox && BUSYBOXDIR=/usr/lib/initramfs-
tools/bin
export BUSYBOXDIR

OPTIONS=`getopt -o c:d:ko:r:v -n "$0" -- "$@"`

# Check for non-GNU getopt
if [ $? != 0 ] ; then echo "W: non-GNU getopt" >&2 ; exit 1 ; fi

eval set -- "$OPTIONS"

while true; do
    case "$1" in
        -c)
            compress="$2"
            shift 2
            ;;
        -d)
            CONFDIR="$2"
            shift 2
            if [ ! -d "${CONFDIR}" ]; then
                echo "${0}: ${CONFDIR}: Not a directory" >&2
                exit 1
            fi
            ;;
        -o)
            outfile="$2"
            shift 2
            ;;
        -k)
            keep="y"
            shift
            ;;
        -r)
            ROOT="$2"
            shift 2
            ;;
        -v)
            verbose="y"
            shift
            ;;
        --)
            shift
            break
            ;;
        *)
            echo "Internal error!" >&2
            exit 1
            ;;
    esac
done

# For dependency ordered mkinitramfs hook scripts.
. /usr/share/initramfs-tools/scripts/functions
. /usr/share/initramfs-tools/hook-functions
```

```

. "${CONFDIR}/initramfs.conf"
EXTRA_CONF=''
for i in /usr/share/initramfs-tools/conf.d/* ${CONFDIR}/conf.d/*; do
    [ -e $i ] && EXTRA_CONF="${EXTRA_CONF} ${basename $i} \
        | grep '^[[:alnum:]]+[[:alnum:]]\._-]*$' | grep -v '\.dpkg-.*$)";
done
# FIXME: deprecated those settings on mkinitramfs run
# these conf dirs are for boot scripts and land on initramfs
for i in ${EXTRA_CONF}; do
    if [ -d "${CONFDIR}/conf.d/${i} ]; then
        echo "Warning: ${CONFDIR}/conf.d/${i} is a directory instead of
file, ignoring."
    elif [ -e "${CONFDIR}/conf.d/${i} ]; then
        . "${CONFDIR}/conf.d/${i}"
    elif [ -e /usr/share/initramfs-tools/conf.d/${i} ]; then
        . /usr/share/initramfs-tools/conf.d/${i}"
    fi
done

# source package configs
for i in /usr/share/initramfs-tools/conf-hooks.d/*; do
    if [ -d "${i}" ]; then
        echo "Warning: ${i} is a directory instead of file, ignoring."
    elif [ -e "${i}" ]; then
        . "${i}"
    fi
done

if [ -n "${UMASK:-}" ]; then
    umask "${UMASK}"
fi

if [ -z "${outfile}" ]; then
    usage
fi

touch "$outfile"
outfile="$(readlink -f "$outfile")"

# And by "version" we really mean path to kernel modules
# This is braindead, and exists to preserve the interface with mkinitrd
if [ $# -ne 1 ]; then
    version="$(uname -r)"
else
    version="${1}"
fi

case "${version}" in
    /lib/modules/*/[!/]*)
        ;;
    /lib/modules/[!/]*)
        version="${version#/lib/modules/}"
        version="${version%%/*}"
        ;;
    esac

case "${version}" in
    /*)
        echo "$PROG: ${version} is not a valid kernel version" >&2
        exit 1
        ;;
    esac

# Check userspace and kernel support for compressed initramfs images
if [ -z "${compress:-}" ]; then
    compress=${COMPRESS}

```

```

else
    COMPRESS=${compress}
fi

if ! command -v "${compress}" >/dev/null 2>&1; then
    compress=gzip
    [ "${verbose}" = y ] && \
        echo "No ${COMPRESS} in ${PATH}, using gzip"
    COMPRESS=gzip
fi

if dpkg --compare-versions "${version}" lt "2.6.38" 2>/dev/null; then
    compress=gzip
    [ "${verbose}" = y ] && \
        echo "linux-2.6 likely misses ${COMPRESS} support, using gzip"
fi

[ "${compress}" = lzop ] && compress="lzop -9"
[ "${compress}" = xz ] && compress="xz --check=crc32"

if [ -d "${outfile}" ]; then
    echo "${outfile} is a directory" >&2
    exit 1
fi

MODULESDIR="/lib/modules/${version}"

if [ ! -e "${MODULESDIR}" ]; then
    echo "WARNING: missing ${MODULESDIR}"
    echo "Ensure all necessary drivers are built into the linux image!"
fi

if [ ! -e "${MODULESDIR}/modules.dep" ]; then
    depmod ${version}
fi

# Prepare to clean up temporary files on exit
DESTDIR=
__TMPCPIOGZ=
__TMPEARLYCPIO=
clean_on_exit() {
    if [ "${keep}" = "y" ]; then
        echo "Working files in ${DESTDIR:-<not yet created>}, early
initramfs in ${__TMPEARLYCPIO:-<not yet created>} and overlay in ${__TMPCPIOGZ:-
<not yet created>}"
    else
        for path in "${DESTDIR}" "${__TMPCPIOGZ}" "${__TMPEARLYCPIO}"; do
            test -z "${path}" || rm -rf "${path}"
        done
    fi
}

trap clean_on_exit EXIT
trap "exit 1" INT TERM # makes the EXIT trap effective even when killed

# Create temporary directory and files for initramfs contents
[ -n "${TMPDIR}" ] && [ ! -w "${TMPDIR}" ] && unset TMPDIR
DESTDIR="$(mktemp -d ${TMPDIR:-/var/tmp}/mkinitramfs_XXXXXX)" || exit 1
chmod 755 "${DESTDIR}"
__TMPCPIOGZ="$(mktemp ${TMPDIR:-/var/tmp}/mkinitramfs-OL_XXXXXX)" || exit 1
__TMPEARLYCPIO="$(mktemp ${TMPDIR:-/var/tmp}/mkinitramfs-FW_XXXXXX)" || exit 1

DPKG_ARCH=`dpkg --print-architecture`

# Export environment for hook scripts.
#
export MODULESDIR
export version

```

```

export CONFDIR
export DESTDIR
export DPKG_ARCH
export verbose
export MODULES
export BUSYBOX
export COMPCACHE_SIZE

# Private, used by 'catenate_cpio.gz'.
export __TMPCPIOGZ

# Private, used by 'prepend_earlyinitramfs'.
export __TMPEARLYCPIO

for d in bin conf/conf.d etc lib/modules run sbin scripts ${MODULESDIR}; do
    mkdir -p "${DESTDIR}/${d}"
done

# Copy in modules.builtin and modules.order (not generated by depmod)
for x in modules.builtin modules.order; do
    if [ -f "${MODULESDIR}/${x}" ]; then
        cp -p "${MODULESDIR}/${x}" "${DESTDIR}${MODULESDIR}/${x}"
    fi
done

# MODULES=list case. Always honour.
for x in "${CONFDIR}/modules" /usr/share/initramfs-tools/modules.d/*; do
    if [ -f "${x}" ]; then
        add_modules_from_file "${x}"
    fi
done

# MODULES=most is default
case "${MODULES}" in
    dep)
        dep_add_modules
        ;;
    most)
        auto_add_modules
        ;;
    netboot)
        auto_add_modules base
        auto_add_modules net
        ;;
    list)
        # nothing to add
        ;;
    *)
        echo "W: mkinitramfs: unsupported MODULES setting: ${MODULES}."
        echo "W: mkinitramfs: Falling back to MODULES=most."
        auto_add_modules
        ;;
esac

# Resolve hidden dependencies
hidden_dep_add_modules

# First file executed by linux
cp -p /usr/share/initramfs-tools/init ${DESTDIR}/init

# add existant boot scripts
for b in $(cd /usr/share/initramfs-tools/scripts/ && find . \
    -regextype posix-extended -regex '.*[[:alnum:]]\._-]+$' -type f); do
    option=$(sed '/^OPTION=!/d;$d;s/^OPTION=//;s/[[:space:]]*$/' "/usr/share/
initramfs-tools/scripts/${b}")
    [ -z "${option}" ] || eval test -n "\"${option}\"" -a "\"${option}\"" !=

```

```

\"n\" || continue

[ -d "${DESTDIR}/scripts/${dirname "${b}"}" ] \
    || mkdir -p "${DESTDIR}/scripts/${dirname "${b}"}"
cp -p "/usr/share/initramfs-tools/scripts/${b}" \
    "${DESTDIR}/scripts/${dirname "${b}"}"
done
for b in $(cd "${CONFDIR}/scripts" && find . \
    -regextype posix-extended -regex '.*/[[:alnum:]]\._-]+$' -type f); do
    option=$(sed '/^OPTION=;!d;$d;s/^OPTION=//;s/[[:space:]]*$/' "${CONFDIR}/
scripts/${b}")
[ -z "${option}" ] || eval test -n "\"${$option}\"" -a "\"${$option}\"" !=
\"n\" || continue

[ -d "${DESTDIR}/scripts/${dirname "${b}"}" ] \
    || mkdir -p "${DESTDIR}/scripts/${dirname "${b}"}"
cp -p "${CONFDIR}/scripts/${b}" "${DESTDIR}/scripts/${dirname "${b}"}"
done

echo "DPKG_ARCH=${DPKG_ARCH}" > ${DESTDIR}/conf/arch.conf
cp -p "${CONFDIR}/initramfs.conf" ${DESTDIR}/conf
for i in ${EXTRA_CONF}; do
    if [ -e "${CONFDIR}/conf.d/${i}" ]; then
        copy_file config "${CONFDIR}/conf.d/${i}" /conf/conf.d
    elif [ -e "/usr/share/initramfs-tools/conf.d/${i}" ]; then
        copy_file config "/usr/share/initramfs-tools/conf.d/${i}" \
            /conf/conf.d
    fi
done

# ROOT hardcoding
if [ -n "${ROOT:-}" ]; then
    echo "ROOT=${ROOT}" > ${DESTDIR}/conf/conf.d/root
fi

if ! command -v ldd >/dev/null 2>&1 ; then
    echo "WARNING: no ldd around - install libc-bin" >&2
    exit 1
fi

# fstab and mtab
touch "${DESTDIR}/etc/fstab"
ln -s /proc/mounts "${DESTDIR}/etc/mtab"

# install wait-for-root binary.
copy_exec /usr/lib/initramfs-tools/bin/wait-for-root /sbin

# module-init-tools
copy_exec /sbin/modprobe /sbin
copy_exec /sbin/rmmod /sbin
mkdir -p "${DESTDIR}/etc/modprobe.d"
cp -a /etc/modprobe.d/* "${DESTDIR}/etc/modprobe.d/"

# util-linux, added by niuao
copy_exec /sbin/blkid /sbin
copy_exec /sbin/losetup /sbin
copy_exec /sbin/kpartx /sbin
copy_exec /bin/ntfs-3g /bin
copy_exec /sbin/vgscan /sbin
copy_exec /sbin/shutdown /shutdown
copy_exec /usr/bin/qemu-nbd /bin
touch ${DESTDIR}/etc/initrd-release
touch ${DESTDIR}/version

# workaround: libgcc always needed on old-abi arm
if [ "${DPKG_ARCH}" = arm ] || [ "${DPKG_ARCH}" = armeb ]; then

```

加在这里(其实
我也是乱放的)

这两句可以删除

```

        cp -a /lib/libgcc_s.so.1 "${DESTDIR}/lib/"
    fi

    run_scripts_optional /usr/share/initramfs-tools/hooks
    run_scripts_optional "${CONFDIR}/hooks

# cache boot run order
for b in $(cd "${DESTDIR}/scripts" && find . -mindepth 1 -type d); do
    cache_run_scripts "${DESTDIR}" "/scripts/${b#./}"
done

# generate module deps
depmod -a -b "${DESTDIR}" ${version}
rm -f "${DESTDIR}/lib/modules/${version}"/modules.*map

# make sure that library search path is up to date
cp -ar /etc/ld.so.conf* "${DESTDIR}/etc/"
if ! ldconfig -r "${DESTDIR}" ; then
    [ $(id -u) != "0" ] \
    && echo "ldconfig might need uid=0 (root) for chroot()" >&2
fi

# Apply DSDT to initramfs
if [ -e "${CONFDIR}/DSDT.aml" ]; then
    copy_file DSDT "${CONFDIR}/DSDT.aml"
fi

# Make sure there is a final sh in initramfs
if [ ! -e "${DESTDIR}/bin/sh" ]; then
    copy_exec /bin/sh "${DESTDIR}/bin/"
fi

# Remove any looping or broken symbolic links, since they break cpio.
[ "${verbose}" = y ] && xargs_verbose="-t"
(cd "${DESTDIR}" && find . -type l -printf '%p %Y\n' | sed -n 's/ [LN]$/p' \
    | xargs ${xargs_verbose:-} -rL1 rm -f)

# dirty hack for armhf's double-linker situation; if we have one of
# the two known eglibc linkers, nuke both and re-create sanity
if [ "$DPKG_ARCH" = armhf ]; then
    if [ -e "${DESTDIR}/lib/arm-linux-gnueabi/lib/ld-linux.so.3" ] || \
        [ -e "${DESTDIR}/lib/ld-linux-armhf.so.3" ]; then
        rm -f "${DESTDIR}/lib/arm-linux-gnueabi/lib/ld-linux.so.3"
        rm -f "${DESTDIR}/lib/ld-linux-armhf.so.3"
        cp -aL /lib/ld-linux-armhf.so.3 "${DESTDIR}/lib/"
        ln -sf /lib/ld-linux-armhf.so.3 "${DESTDIR}/lib/arm-linux-
gnueabi/lib/ld-linux.so.3"
    fi
fi

[ "${verbose}" = y ] && echo "Building cpio ${outfile} initramfs"

if [ -s "${__TMPEARLYCPIO}" ]; then
    cat "${__TMPEARLYCPIO}" >"${outfile}" || exit 1
else
    # truncate
    > "${outfile}"
fi

(
# preserve permissions if root builds the image, see #633582
[ "$(id -ru)" != 0 ] && cpio_owner_root="-R 0:0"

# work around lack of "set -o pipefail" for the following pipe:
# cd "${DESTDIR}" && find . | cpio --quiet $cpio_owner_root -o -H newc | gzip >>"$
{outfile}" || exit 1

```

```
exec 3>&1
eval `
    # http://cfaj.freeshell.org/shell/cus-faq-2.html
    exec 4>&1 >&3 3>&-
    cd "${DESTDIR}"
    {
        find . 4>&-; echo "ec1=$?;" >&4
    } | {
        cpio --quiet $cpio_owner_root -o -H newc 4>&-; echo "ec2=$?;" >&4
    } | ${compress} >>"${outfile}"
    echo "ec3=$?;" >&4
`
if [ "$ec1" -ne 0 ]; then
    echo "E: mkinitramfs failure find $ec1 cpio $ec2 $compress $ec3"
    exit "$ec1"
fi
if [ "$ec2" -ne 0 ]; then
    echo "E: mkinitramfs failure cpio $ec2 $compress $ec3"
    exit "$ec2"
fi
if [ "$ec3" -ne 0 ]; then
    echo "E: mkinitramfs failure $compress $ec3"
    exit "$ec3"
fi
) || exit 1

if [ -s "${__TMPPIOGZ}" ]; then
    cat "${__TMPPIOGZ}" >>"${outfile}" || exit 1
fi

exit 0
```