# The glucopy-glurestore system deployment scripts

## Table of Contents

## Introduction

When you need to make more than one copy of your installed and fine-tuned system in less than a lifetime, these scripts are for you. We have used them to back-up, clone, and individualize slightly complex working systems, including systems with disk arrays (RAID). Restoration from compressed data taking up a 700Mb CD or a 2Gb DVD takes about ten minutes (current era computers).

## History

20061203 – First release: tlgu.carmen.gr
20061223 – EOF correction
20080102 – GRUB syntax correction, distribution-specific considerations
20120329 – RAID backup – restore, additional considerations

## Acknowledgements

Spiros Georgaras (sngeorgaras, otenet gr) for reviewing the scripts and for providing useful checks.

All authors and contributors to the utilities listed below.

## References

(use **man** command or **info** command to find out more)

**dd**

History has it that dd was originally cc (convert and copy) and was renamed to dd as cc stood for c compiler.
It has always been device dump or disk dump for me...

**grub**

The modularized GRand Unified Bootloader is used in the systems supported by the copy/restore scripts.

An interactive mode allows you options like finding the partitions where the boot loader files reside:

```
find /boot/grub/stage1
```

This will return something like root(hd0,1) which means that grub's root directory was found in the first hdd, second partition.

As long as you have it running, you may then set up the system to boot:

```
setup(hd0)
quit
```

The setup line indicates where the boot loader will reside (here in the first disk's Master Boot Record - MBR) and the quit command will flush the information to disk.

To boot from interactive mode you must indicate the location of the GNU/Linux system (relative to the hard disk drive used for booting) and the root directory using real device names. For example:

```
kernel /vmlinuz root=/dev/sda2 [additional kernel
parameters]
initrd /initrd
boot
```

**sfdisk**

A comprehensive disk partitioner with sector save (-O), dump (-d) and restore options.

**tar**

The venerable tape archive command can be used to easily create compressed or uncompressed archives in forms too numerous to mention here. For added peace of mind compute the **md5sum** of the resulting “tarballs” and compare them to the respective files stored on the backup medium (CD/DVD).

**mdadm**

Multiple Device (MD) administration, used for systems implementing software RAID (Redundant Array of Inexpensive Disks).

**md5sum**

Computes a file's checksum or "fingerprint".

**udev**

User-space dynamic device management.

**cpio**

Archiving utility, also used in making initial RAM disk (**initrd**) images.

**mkisofs**
**cdrecord** (now **wodim** - write data to optical disk media)

These utilities can be used to make and burn an ISO CD, optionally including bootable images (-b).

**growisofs**

Used as a front end to **mkisofs** to make and burn DVD-RW and DVD-RAM images.

## Disclaimer

**BE CAREFUL! THESE ARE EXAMPLE SCRIPTS AND WILL DEFINITELY NEED ADAPTATION!**

**TEST USING SAMPLE DATA/HARDWARE BEFORE ACTUAL DEPLOYMENT!**

The usual disclaimer about misconfiguring your system beyond repair or oblueterating your work applies: Don't blame it on us. Do one thing at a time. Make sure you understand the steps involved by reading the respective command manuals. Write to tlgu, carmen gr, in case this document contains inaccuracies, errors or if you have some information that others can benefit from.

## The need

The need arose (a rose... a rose!) to install a specific GNU/Linux configuration on a number of identical computer boxes with (almost) blank, unformatted, disks. One option was to blindly clone the disks using **dd** but this would need a significant amount of time, even with a large block size, indeed unjustified, as the sub-Tb disks were mostly empty. Another drawback is that disks should be removed and re-installed. Even with “no-tools” boxes this is an added inconvenience and involves a risk of damaging the hardware. Last, but not least, added complexity, introduced with more recent operating system distributions, necessitated system-specific adjustments following the “disk cloning” operation.

## The solution

We eventually settled on backing-up directories from the GNU/Linux tree by making a compressed **tar** ball (tape archive), copying it onto a CD/DVD and then expanding it onto freshly-formatted partitions. Which means: boot a compact working system on the target machine, clone partitioning information, format selected partitions, untar-uncompress backed-up information, make system bootable and, optionally, run any required customization scripts; this has lately become mandatory, as current distributions include persistent device naming introduced with **udev**, such as disk Universally Unique IDentifiers (UUIDs) and Ethernet Media Access Control (MAC) addresses, as well as policy mechanisms for running system services.

## The back-up script

The back-up should (ideally) be made while the system is not running, which means that you need to boot with a rescue system disk. A number of suitable floppy, USB or CD-based system images can be downloaded from the net or, better, included in your distribution's boot options. Make sure that this rescue system supports the type of disks in your target system (SATA/SCSI or IDE).

*The not-ideal case is when the system is running in single-user mode (**init 1**). As some programs **are** running at this run level, make sure that this will have no adverse effects to your backup.*

The **glucopy** script will get partition information in a form that can be used to restore the system on new disks and will then collect the working system files in one (or more) compressed tar balls.

The **sfdisk** program is used for getting partition information (-d option). Take a look at **sys_partitions.txt** after the **glucopy.sh** script has finished.

In this application it is considered that all Master Boot Record (MBR) information will be overwritten by the partitioning and boot loader programs. If you need a copy of all or parts of the MBR, however, **dd** is your friend.

The script must be run by the **root** user, so that all system files can be copied and file attributes preserved. **tar**'s preserve permissions option (-p) is used to that effect.

System directories that contain transient information (e.g. **/proc**, **/sys**) are not copied. Directories or individual files containing unwanted information can be excluded using tar's –exclude=/directory/file option.

**<u>Important note</u>**: For a successful system restoration, a suitable configuration should be present on the back-up volume. Consult the **Considerations** section.

```
#!/bin/sh
##############################################################
# glucopy.sh
#
# Creates the /glucopy directory
# Copies partition information in sys_partitions.txt
# Optionally makes a copy of the MBR in mbr.bin
# Copies working GNU/Linux system to .tgz files
#
# USAGE:
# Boot system with a rescue CD or any bootable CD
# that supports your type of disks (SATA/SCSI or IDE).
# run the back-up script: ./glucopy.sh
#
# In this example the system is on /dev/sda
# /dev/sda1 is a small partition with HW manufacturer utilities
# /dev/sda2 has system root (/) -> mounted to /mnt
# /dev/sda3 is a spare (blank) partition
# /dev/sda4 has home (/home) and space to hold
#           the information to be copied -> mounted to /mnt2
#
# 061116 jp/dm
##############################################################
# Change the following to reflect the place where your system root (/)
# and the home directories (/home/*) are mounted
mkdir /tmp/mnt1
mkdir /tmp/mnt2
mount /dev/sda2 /tmp/mnt1
mount /dev/sda4 /tmp/mnt2

# The directory to hold the system copy is placed on the larger partition
mkdir /tmp/mnt2/glucopy

# Change to the working directory
cd /tmp/mnt2/glucopy

# Optionally copy MBR information
#dd if=/dev/sda of=mbr.bin bs=512 count=1

# Copy partition information
/sbin/sfdisk -d /dev/sda > sys_partitions.txt

# Make tarballs, preserving file permissions (default if run as root)
tar -C /tmp/mnt1 -cpvzf syscopy.tgz boot bin etc lib media opt root sbin usr var dev

# Move up a level
cd ..
# here we are under the /home directory level
# notice that --exclude will exclude anything that matches a _pattern_
# the glucopy directory will still be created in the tarball
```

```
tar -cpvzf ./glucopy/homecopy.tgz * --exclude=glucopy/*


# Copy the backup/restore scripts (here under root /), and the sfdisk
program,  as well
cp /glucopy.sh ./glucopy
cp /glurestore.sh ./glucopy
cp /sbin/sfdisk ./glucopy


# Notify user
echo -e “glucopy: finished \a”
```

## The RAID back-up script

A slightly more involved situation is when you have to back up a RAID-based system. Information about preparing such a system may be found in the **RAID Preparation** section, below. When you boot with your rescue or system disk, the **mdadm** utility has to be available, in order to start the disk array prior to commencing the back-up operation. The following example command will start array /dev/md2, comprising /dev/sda2 and /dev/sdb2:

```
# mdadm --auto=yes /dev/md2 /dev/sd[ab]2
```

Such commands may be included in the backup script, for automatic execution. Verify that the RAID has started, by running

```
# cat /proc/mdstat
```

The example **glucopy-raid** script for backing up both RAID and non-RAID partitions is presented below.

**Note** that for a successful system restoration, a suitable configuration should be present on the back-up volume. Consult the **Considerations** section where issues are discussed in more detail.

```
#!/bin/sh
##############################################################
# glucopy-raid.sh
#
# Creates the /glucopy directory
# Copies partition information in sys_partitions.txt
# Optionally makes a copy of the MBR in mbr.bin
# Copies working GNU/Linux system to .tgz files
#
```

```
# USAGE:
# Boot system with a rescue CD or any bootable CD
# that supports your type of disks (SATA/SCSI or IDE).
# run the back-up script: ./glucopy.sh
#
# In this example the system is on /dev/sda
# /dev/sda1 has boot (/boot) -> mounted to /mnt1
# /dev/sda2 (/dev/md2) has system root (/) -> mounted to /mnt2
# /dev/sda3 has home (/home) and space
# to hold the information to be copied -> mounted to /mnt3
# /dev/sda4 is a spare partition
# /dev/sdb1 is a copy of /boot
# /dev/sdb2 participates in /dev/md2
# /dev/sdb3 participates in /dev/md3
# /dev/sdb4 is a spare partition
# 061116 jp/dm
# 070503 sg - root user and program availability checks
# 091217 dm - raid (md) volumes
#
# Multiple-device volumes must be started before the script is run
# in single-user mode or after booting with a live CD:
#
# init 1
# mdadm --auto=yes /dev/md2 /dev/sd[ab]2
# mdadm --auto=yes /dev/md3 /dev/sd[ab]3
##############################################################
# Change the following to reflect the place where your system root (/)
# and the home directories (/home/*) are mounted

# Check that user is root
USER=`id -un`
if (test $USER != "root");then
        echo "Error: This script must be executed by root"
        exit
fi

# Check availability of programs
TAR=`which tar 2>/dev/null`
if (test -z "$TAR");then
```

```
        echo "Error: The \"tar\" program was not found"
        echo "       The live CD you are using does not contain the
program \"tar\"."
        echo "       The backup procedure cannot be executed without
it."
  echo "       Please use another live CD."
  exit
fi

# The sfdisk program may be available on your boot medium and/or
# your boot disk.  In the latter case, make sure that the sfdisk
# program is compatible with the boot system.
#
SFDISK=`which sfdisk 2>/dev/null`
#SFDISK=`pwd`\/sfdisk  2>/dev/null

if (test -z "$SFDISK");then
        echo "Error: The \"sfdisk\" program was not found"
        echo "       The live CD you are using does not contain the
program \"sfdisk\"."
        echo "       The backup procedure cannot be executed without
it."
        echo "       Please use another live CD."
        exit
fi

# The mdadm program may be available on your boot medium and/or
# your boot disk.  In the latter case, make sure that the mdadm
# program is compatible with the boot system.
#
#MDADM=`which mdadm 2>/dev/null`
MDADM=`pwd`\/mdadm

if (test -z "$MDADM");then
        echo "Error: The \"mdadm\" program was not found"
        echo "       The live CD you are using does not contain the
program \"mdadm\"."
        echo "       The backup procedure cannot be executed without
it."
        echo "       Please use another live CD."
        exit
```

```
fi

# Checkpoint - use the exit command to check parts of your script
#exit

mkdir /tmp/mnt1
mkdir /tmp/mnt2
mkdir /tmp/mnt3
mount /dev/sda1 /tmp/mnt1
mount /dev/md2 /tmp/mnt2
mount /dev/md3 /tmp/mnt3

# The directory to hold the system copy is placed on the larger (home) partition
mkdir /tmp/mnt3/glucopy

# Change to the working directory
cd /tmp/mnt3/glucopy

# Optionally copy MBR information
#dd if=/dev/sda of=mbr.bin bs=512 count=1
# Copy partition information
"$SFDISK" -d /dev/sda > sfdisk.sda
"$SFDISK" -d /dev/sdb > sfdisk.sdb

# Make tarballs
"$TAR" -C /tmp/mnt1 -cpvzf bootcopy.tgz `find .`
"$TAR" -C /tmp/mnt2 -cpvzf syscopy.tgz bin etc lib media opt root sbin usr var dev

# Move up a level
cd ..
# here we are under the /home directory level
# notice that --exclude will exclude anything that matches a _pattern_
# the glucopy directory will still be created in the tarball
"$TAR" --exclude=glucopy/* -cpvzf ./glucopy/homecopy.tgz *

# Copy the backup/restore scripts (in the present working directory), and the sfdisk program
cp `pwd`/* ./glucopy
cp "$SFDISK" ./glucopy
cp "$MDADM" ./glucopy

# Notify user
echo glucopy: finished
```

## CD/DVD preparation

The information in the glucopy directory can be burned onto a CD or DVD using the **mkisofs** and **cdrecord** (**wodim**) or **growisofs** or by using a GUI front end such as **k3b**. You can make the medium bootable by including a suitable system image.

The two example scripts below (**cdmake**, **cdburn**) will make a CD ISO image in the /tmp directory and burn it to a CD. Or you can use the **growisofs** command: to burn CD/DVDs e.g. growisofs /dev/hdc <mkisofs options>

Needless to say, the scripts need to be customized for your own equipment.

```
#-------------------------------------------------------
# Usage: cdmake VOLUME_LABEL DIRECTORY
# Will output /tmp/cdimage.iso
#
if (test -z $1 -o -z $2); then
        echo Usage: cdmake VOLUME_LABEL TOP_LEVEL_DIRECTORY
else
mkisofs -V $1 -rational-rock -full-iso9660-filenames \
  -iso-level 2 -o /tmp/cdimage.iso  $2
fi
#-------------------------------------------------------
Example: ./cdmake BACK061203 /home/glucopy
```

```
#-------------------------------------------------------
# cdburn
cdrecord gracetime=2 -v dev=/dev/hdc -speed=48 -multi -eject
driveropts=burnfree /tmp/cdimage.iso
#-------------------------------------------------------
Example: ./cdburn
```

## The restore script

A minimal (or not-so-minimal) system is loaded using a rescue CD/USB/floppy, at which point the restore script is run.

The **glurestore** script will partition the target disk using information obtained from the original system, will format the desired partitions and will then mount the partitions, restore the data, and make the target disk bootable. A partition re-read (using two methods) will make sure that the partitions will be available prior to formatting.

Note that **/proc** and **/sys** directories (which were not backed up) need to be created. The **/mnt** directory is created, as well.

System restoration may be followed by individualization commands for hard-coded information removal or system serialization.

**Important note:**

**KEEP YOUR REWRITABLE MEDIA IN A PLACE OTHER THAN THE COMPUTER BEING RESTORED.**
**THE BLUE MEANIES ARE LURKING ABOUT.**

```
#!/bin/sh
##############################################################
# glurestore.sh
#
# USAGE:
# Boot system with a rescue CD or any bootable medium
# that supports your type of disks (SATA/SCSI or IDE).
#
# Mount the medium containing the backed-up files and
# run the restore script:
# e.g. mount /dev/hda /mnt
#      /mnt/glurestore.sh
#
# In this example the system is on /dev/sda
# and the CD/DVD ROM drive is on /dev/hda
# There are four primary partitions:
# /dev/sda1 contains HW manufacturer utilities and will not be formatted
# /dev/sda2 will be formatted using ext3 and will hold the system root (/)
# /dev/sda3 will be formatted using ext3 (but will not be used)
# /dev/sda4 will be formatted using ext3 and will hold /home
#
# 061116 jp/dm
# 080102 dm - note that a space is needed after the GRUB commands
# 120207 dm - reread partitions, remove system-specific information
##############################################################

# Change to the restore medium directory
cd /mnt

# Optionally install Master Boot Record
# (uncomment the following two lines)
#echo "Installing MBR"
#dd if=mbr.bin of=/dev/sda bs=512 count=1

# Create Partitions
# The sfdisk program used has been copied
# on the backup medium.
echo "Creating Partition Table"
./sfdisk /dev/sda < sys_partitions.txt

# Reread partitions (with a short delay)
echo "Rereading partition tables (rescan, sfdisk)"
echo 1 > /sys/block/sda/device/rescan
sleep 1
"$SFDISK" -R /dev/sda
sleep 1

# Format partitions
# Change to reflect the number of partitions that need
# to be formatted and the type of formatting
echo "Formating Partitions"
mkfs.ext3 /dev/sda2
mkfs.ext3 /dev/sda3
mkfs.ext3 /dev/sda4

# Create directory to mount "/" partition
echo "Creating  directory to mount / partition "
```

```
mkdir /mnt2
mount /dev/sda2 /mnt2

# Create /proc /sys and /mnt directories
echo "Creating  proc, sys and mnt directories "
mkdir /mnt2/proc
mkdir /mnt2/sys
mkdir /mnt2/mnt

# Create directory and mount "home" partition
echo "Creating  directory and mounting home partition "
mkdir /mnt2/home
mount /dev/sda4 /mnt2/home

# Extract files
tar -C /mnt2 -xpvzf syscopy.tgz
tar -C /mnt2 -xpvzf homecopy.tgz

#####################################
# Optional commands may be placed here
# to remove hard-coded information or
# to add system-specific parameters
#####################################

rm -f /mnt2/etc/iftab
rm -f /mnt2/etc/udev/rules.d/70-persistent-net.rules
rm -f /mnt1/var/lib/dbus/machine-id
rm -f /mnt2/root/.bash_history
rm -f /mnt2/home/__user__/.bash_history

# Now make sure that correct boot information
# is present in the hard disk's MBR
# grub is run in batch mode
# hd0,1 is hard disk 0 (first hard disk of any kind), second partition
– for this example
/sbin/grub --batch <<EOF
root (hd0,1)
setup (hd0)
quit
EOF
```

```

# Notify user
echo -e “glurestore: finished! \a”

# Reboot and check your new system
reboot
```

## The RAID restore script

The **glurestore-raid** script in this example will partition the target disk using information obtained from the original system, will start the disk arrays, and will format individual partitions and volumes. The script will then mount the partitions, restore the data, and make the target disks bootable. A partition re-read (using two methods) will make sure that the partitions will be available prior to formatting.

Note that **/proc** and **/sys** directories (which were not backed up) need to be created. The **/mnt** directory is created, as well.

System restoration may be followed by individualization commands for hard-coded information removal or system serialization.

**VERY IMPORTANT RAID NOTE:**

**KEEP YOUR REWRITABLE MEDIA IN A PLACE OTHER THAN THE COMPUTER BEING RESTORED.**
**THE BLUE MEANIES ARE LURKING ABOUT; ACCIDENTS WILL HAPPEN AND ARE MORE LIKELY WHEN YOU ARE RESTORING RAID ARRAYS, COMPLETE, OR - MAKER FORBID - WITH MISSING DISKS. IF YOUR USB FLASH DISK OR USB HARD DISK IS INADVERTENTLY ASSIGNED A PARTITION IN THE RANGE OF THE ARRAY BEING RESTORED IT WILL CERTAINLY BE OBLUETERATED (A SAD THING TO HAPPEN).**

```
#!/bin/sh
##############################################################
# glurestore-raid.sh
#
# USAGE:
# Boot system with a rescue CD or any bootable medium
# that supports your type of disks (SATA/SCSI or IDE).
#
# Mount the medium containing the backed-up files and
# run the restore script:
# e.g. mount /dev/hda /mnt
#      /mnt/glurestore-pclinuxos-raid.sh
#
# In this example the system is on /dev/sda and /dev/sdb
# Two partitions (2 and 3) are in RAID1 (mirror) configuration
# There are four primary partitions:
# /dev/sda1 and /dev/sdb1 will hold /boot (non-RAID)
# /dev/sda2 and /dev/sdb2 will hold / (/dev/md2)
# /dev/sda3 and /dev/sdb3 will hold /home (/dev/md3)
# /dev/sda4 and /dev/sdb4 are free (non-RAID)
#
# Boot (GRUB) information is written on the
# Master Boot Records (MBR) of both disks
#
# 061116 jp/dm
# 061223 dm -- EOF correction
# 070503 sg -- root user and program availability checks
# 071220 dm -- remove hard-coded MAC address files (PCLinuxOS)
# 091217 dm -- raid (md) preparation
# 120207 dm - reread partitions, remove system-specific information
##############################################################
# Check that user is root
USER=`id -un`
if (test $USER != "root"); then
        echo "Error: This script must be executed by root"
        exit
fi

# Check availability of programs
TAR=`which tar 2>/dev/null`
if (test -z "$TAR"); then
        echo "Error: The \"tar\" program was not found"
        echo "       The live CD you are using does not contain the
program \"tar\"."
        echo "       The backup procedure cannot be executed without
it."
        echo "       Please use another live CD."
        exit
fi

# The sfdisk program may be available on your boot medium and/or
# your boot disk.  In the latter case, make sure that the sfdisk
# program is compatible with the boot system.
#
SFDISK=`which sfdisk 2>/dev/null`
#SFDISK=`pwd`\/sfdisk  2>/dev/null

if (test -z "$SFDISK"); then
        echo "Error: The \"sfdisk\" program was not found"
        echo "       The live CD you are using does not contain the
program \"sfdisk\"."
        echo "       The backup procedure cannot be executed without
it."
        echo "       Please use another live CD."
        exit
fi

# The mdadm program may be available on your boot medium and/or
# your boot disk.  In the latter case, make sure that the mdadm
# program is compatible with the boot system.
#
#MDADM=`which mdadm 2>/dev/null`
MDADM=`pwd`\/mdadm
#echo $MDADM
if (test -z "$MDADM"); then
        echo "Error: The \"mdadm\" program was not found"
        echo "       The live CD you are using does not contain the
program \"mdadm\"."
        echo "       The backup procedure cannot be executed without
it."
        echo "       Please use another live CD."
        exit
fi
GRUB=`which grub 2>/dev/null`
if (test -z "$GRUB"); then
        echo "Error: The \"grub\" program was not found"
        echo "       The live CD you are using does not contain the
program \"grub\"."
        echo "       The backup procedure cannot be executed without
it."
        echo "       Please use another live CD."
        exit
```

```
fi

MKEXT2=`which mkfs.ext2 2>/dev/null`
MKEXT3=`which mkfs.ext3 2>/dev/null`

if [[ -z "$MKEXT2" || -z "$MKEXT3" ]]; then
        echo "Error: Program \"mkfs.ext2\" or \"mkfs.ext3\" was not found"
        echo "       The live CD you are using does not contain this program."
        echo "       The backup procedure cannot be executed without it."
        echo "       Please use another live CD."
        exit
fi

# Checkpoint
#exit

THIS_DIR=`pwd`

# Optionally install Master Boot Record
# (uncomment the following two lines)
#echo "Installing MBR"
#dd if=mbr.bin of=/dev/sda bs=512 count=1

# Create Partitions
# If both disks have the same partitions
# there is no need for a separate sfdisk file.
echo "Creating Partition Tables"
"$SFDISK" -O "$THIS_DIR/disk_a_info" -f /dev/sda < sfdisk.sda
"$SFDISK" -O "$THIS_DIR/disk_b_info" -f /dev/sdb < sfdisk.sdb

# Reread partitions (with a short delay)
echo "Rereading partition tables (rescan, sfdisk)"
echo 1 > /sys/block/sda/device/rescan
sleep 1
echo 1 > /sys/block/sdb/device/rescan
sleep 1
"$SFDISK" -R /dev/sda
```

```
sleep 1
"$SFDISK" -R /dev/sdb
sleep 1

# Create RAID volumes
echo "Creating RAID Volumes"
modprobe raid1
"$MDADM" -C /dev/md2 -f -R -l1 -n2 /dev/sda2 /dev/sdb2
"$MDADM" -C /dev/md3 -f -R -l1 -n2 /dev/sda3 /dev/sdb3

# on a single disk...
#"$MDADM" -C /dev/md2 -f -R -l1 -n2 /dev/sda2 missing
#"$MDADM" -C /dev/md2 -R -l1 -n2 /dev/sda2 missing
#"$MDADM" -C /dev/md3 -R -l1 -n2 /dev/sda3 missing

# Checkpoint
#exit

# Format partitions
# Change to reflect the number of partitions that need
# to be formatted and the type of formatting
echo "Formatting disk 1 boot partition"
"$MKEXT2" /dev/sda1
echo "Formatting disk 2 boot partition"
"$MKEXT2" /dev/sdb1
echo "Formatting md2"
"$MKEXT3" /dev/md2
echo "Formatting md3"
"$MKEXT3" /dev/md3
#echo "Formatting disk1 empty partition"
#"$MKEXT3" /dev/sda4
#echo "Formatting disk2 empty partition"
#"$MKEXT3" /dev/sdb4

# Create directory to mount partitions; start with root (/) directory
echo "Creating directory to mount partitions "
mkdir /mnt2
mount /dev/md2 /mnt2
```

```
# Create /proc /sys and /mnt directories
echo "Creating proc, sys and mnt directories "
mkdir /mnt2/proc
mkdir /mnt2/sys
mkdir /mnt2/mnt

# Create directory and mount "home" partition
echo "Creating  directory and mounting home partition "
mkdir /mnt2/home
mount /dev/md3 /mnt2/home

# Create directory and mount "boot" partition
echo "Creating  directory and mounting boot partition (first disk)"
mkdir /mnt2/boot
mount /dev/sda1 /mnt2/boot

# Extract files
echo "Extracting files (root, home, boot1)"
"$TAR" -C /mnt2 -xpvzf syscopy.tgz
"$TAR" -C /mnt2/home -xpvzf homecopy.tgz
"$TAR" -C /mnt2/boot -xpvzf bootcopy.tgz

# Mount second boot directory and copy the boot files over
echo "Mounting and copying boot partition (second disk)"
mount /dev/sdb1 /mnt2/boot
echo "Extracting files (boot2)"
"$TAR" -C /mnt2/boot -xpvzf bootcopy.tgz

#####################################
# Optional commands may be placed here
# to remove hard-coded information or
# to add system-specific parameters
#####################################

rm -f /mnt2/etc/iftab
rm -f /mnt2/etc/udev/rules.d/70-persistent-net.rules
rm -f /mnt1/var/lib/dbus/machine-id
rm -f /mnt2/root/.bash_history
rm -f /mnt2/home/ats/.bash_history
```

```
# Now make sure that correct boot information
# is present in the hard disk's MBR
# grub is run in batch mode
# hd0,0 is hard disk 0 (first hard disk of any kind), first partition – for this example
"$GRUB" --batch <<EOF
root (hd0,0)
setup (hd0)
quit
EOF

"$GRUB" --batch <<EOF
root (hd1,0)
setup (hd1)
quit
EOF

# Notify user
echo glurestore: finished!
# Reboot, if desirable
#reboot
```

## Considerations

### Distribution-specific

**SuSE** disks include a RAM-based rescue system. So does the **Trinity Rescue Kit**, which is completely RAM-based.

**Ubuntu**, for security reasons, does not have a user named **root**. Login and execute the command **sudo su**, providing the default user's password, to gain persistent system administrator rights.

**PCLinuxOS** has a **copy2ram** boot option that can be used to free the **LiveCD** drive. If this does not fit in the available computer memory, try the **console** option.

### Persistent device attributes – MAC addresses

Recent GNU/Linux distributions employ the **udev** (dynamic device management) mechanisms to identify and assign persistent names to network interfaces and hard disk partitions. This is not desirable in a cloning process as different computer hardware are certain to have different MAC addresses and different storage drive UUIDs.

MAC addresses are known to be stored in **/etc/iftab**, **/etc/udev/rules.d/xx-persistent-net.rules** (xx is a number), **/etc/sysconfig/network-scripts/ifcfg-ethx** (x is a number) and similar. Entries may have been made manually (/etc/iftab) or dynamically.

The files containing dynamic information need and can be excluded. This may be done during back-up (e.g. add a pertinent tar --exclude= xxx in **glucopy**) or the unwanted files may be deleted following restoration. The latter method is used in the **glucopy/glurestore** variants in the examples discussed above. Note that, as an afterthought, the root and user **.bash_history** files are also purged. The same can be extended to **browser cache files** or other transient information.

## Persistent device attributes – storage media identifiers

A more involved process must be followed to eliminate the effects of storage media (e.g. hard disk drive) naming. Recent distributions refer to storage media and partitions by name (type – model – serial number) or by computed UUID. The relevant entries may be found in **/dev/disk/by-id** and **/dev/disk/by-uuid**. These symbolic links **should be replaced by "real" device node names or the "cloned" system will crash and burn!**

The persistent node names propagate to various files of an installed system, notably **/etc/fstab** (file system table), **initrd** (the initial RAM-disk image, used by many), and even the bootloader options file, **/boot/grub/menu.lst** if GRUB is used. In the case of RAID, UUIDs may be used in **/etc/mdadm.conf**.

The file system table and the GRUB bootloader files are the easiest to fix. Edit the files mentioned above using a plain text editor. Replace names or UUIDs by the corresponding device names.

## Initial RAM disk (initrd) and cpio

Fixing the initial RAM disk is more involved. The information is stored inside a file in the **/boot** directory, usually called initrd-*release_version*.img and (also usually) pointed to by a symbolic link called **initrd.img**. This file contains a compressed file system including libraries, modules, and a script that will take care of initial system startup. You may check the contents of the initial RAM disk image by running:

```
lsinitrd /boot/initrd.img
```

This will result in a listing of the image contents, followed by the contents of the startup script, called **init**. If this file contains commands with disk names or UUIDs, notably **mkrootdev** ... or **waitdev** ..., the initial RAM disk image must be re-generated.

**AFTER the /etc/fstab and /boot/grub/menu.lst files have been edited**, the initial RAM-disk image may be recreated using the following commands, noting that `uname -r` evaluates to the current kernel release:

```
# Change to the /boot directory
cd /boot

# Save the current RAM disk image file
cp  initrd-`uname -r`.img  initrd-`uname -r`.img.original

# Generate a new initrd-xxxxxx.img
mkinitrd initrd-`uname -r`.img `uname -r`
```

Check the initrd.img contents again by running **lsinitrd initrd.img**. If there are still UUID or name references in the **init** script part there are two options:

1. Fix the **mkinitrd** script

2. Edit the **init** script inside the RAM disk image.

The first option is obvious; as we do not want to interfere with distribution goodness, we will take option 2.

```
# Create a temporary work area in /boot
cd /boot
mkdir rdtmp
cd rdtmp
# Get the target ramdisk image and make it look like a gz archive
cp ../initrd-`uname -r`.img ./rdtmp.img.gz
# Uncompress the image
gunzip rdtmp.img.gz
# Un-archive the uncompressed image
cpio -i < rdtmp.img
```

This will result in the RAM disk file structure created in /boot/rdtemp. Edit the **init** file and replace name or UUID references with device node names. Re-archive, re-compress, and properly rename the initial RAM disk image:

```
# Remove the previous uncompressed image
rm -f rdtmp.img

# Create a new gzipped initrd archive
```

```
find . | cpio -o --'format=newc' | gzip > initrd.new

# Save the existing image and set the new one in its place
cd /boot
mv initrd-`uname -r`.img  initrd-`uname -r`.img.org
cp rdtmp/initrd.new /boot/initrd-`uname -r`.img
```

That's it. The resulting initrd.img should now be good for both the prototype as well as the cloned systems.

### File ownership

In the process of backing up and restoring computer systems, we have encountered at least two cases of services failing to start due to insufficient user privileges (read wrong group ID - GID) assigned to system services start-up scripts. One of them was the HAL (Hardware Abstraction Layer) daemon (**hald**), taking care of real-time hardware changes. The other one was the D-bus message daemon (**dbus-daemon**).

The ultimate workaround for **hald** is to set the not-so-well-known **--retain-privileges** option on the daemon's invocation line. This, for PCLinuxOS may be found in **/etc/init.d/haldaemon.**

```
daemon --check $servicename $processname --retain-privileges
```

For the **dbus** case, the group ID was wrong (or the Set User ID flag was not set). The commands below fixed this. Note that group IDs are numeric values that change from system to system. When executing such commands (**chown**, **chgrp**) keep in mind that the group ID numbers in **/etc/group** should match the desired group names, as the user ID numbers in **/etc/passwd** should match the desired user names.

```
chgrp messagebus /lib/dbus-1/dbus-daemon-launch-helper
chmod u+s /lib/dbus-1/dbus-daemon-launch-helper
```

## RAID preparation

The RAID system to be backed-up, has to be prepared somehow. Here are the steps used in preparing the system presented in the RAID back-up and restore scripts, employing two physical disks with three partitions each. The first partition in each disk is formatted separately and used for booting the system, while the other two partitions are used in corresponding multiple-device arrays (mirrored).

The **mdadm** package is needed – install using synaptic or download individual package and install as follows:

```
rpm -i mdadm-xxxx-xxxx.rpm
```

Load RAID1 (mirror) module:

```
modprobe raid1
```

Create RAID arrays:

```
mdadm -C /dev/md2 -l1 -n2 /dev/sda2 /dev/sdb2
mdadm -C /dev/md3 -l1 -n2 /dev/sda3 /dev/sdb3
```

or (verbose syntax)

```
mdadm --create /dev/md2 --level raid1 --raid-disks 2 /dev/sda2 /dev/sdb2
mdadm --create /dev/md3 --level raid1 --raid-disks 2 /dev/sda3 /dev/sdb3
```

Verify that the RAID arrays are active:

```
cat /proc/mdstat
(md2: active raid1 sda2 sdb2, md3: active raid1 sda3 sdb3)
```

Add array information to /etc/mdam.conf:

```
mdadm --examine --scan >> /etc/mdadm.conf
```

Edit **/etc/mdadm.conf**; UUIDs must be replaced with device names for the system to be portable, namely /dev/md2, /dev/md3. The file should read:

```
DEVICE /dev/sd[ab]2
DEVICE /dev/sd[ab]3
ARRAY /dev/md2 level=raid1 num-devices=2 devices=/dev/sda2,/dev/sdb2
ARRAY /dev/md3 level=raid1 num-devices=2 devices=/dev/sda3,/dev/sdb3
```

Format arrays:

```
mkfs.ext3 /dev/md2
mkfs.ext3 /dev/md3
```

Format boot partitions:

```
mkfs.ext2 /dev/sda1
mkfs.ext2 /dev/sdb1
```

Prepare fstab for booting with /dev/md2 as root:

```
cd /etc
cp fstab fstab.sda2
```

Edit fstab; change root device to /dev/md2; add the **noatime** option, if desirable.

**Note** that if /etc/fstab is not changed, the initrd image that will be prepared will have the wrong information.

**/etc/fstab** should read:

```
/dev/md2 / ext3 defaults,noatime 1 1
/dev/sda1 /boot ext2 defaults,noatime 1 2
/dev/md3 /home ext3 defaults,noatime 1 2
none /proc proc defaults 0 0
none /dev/pts devpts mode-0620 0 0
```

The initial RAM-disk image is used on boot, and must include the proper root device name (**/dev/md2**), the **mdadm** executable, which may not be present in the system startup disk, and the **raid1** kernel module.

Prepare an **initrd** image including the **raid1** module (`uname -r` evaluates to the current kernel-release version):

```
cd /boot
mkinitrd --with=raid1 initrd-`uname -r`-raid.img `uname -r`
```

Verify that the raid1 module is included, that the root device is /dev/md2 and that the init script contained in the initrd image is free from UUID references, as discussed in the **Initial RAM disk** section, above:

```
lsinitrd initrd-`uname -r`-raid.img
```

## GNU Free Documentation License

Version 1.2, November 2002

```
of this license document, but changing it is not allowed.
```

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

**A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

**B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

**C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.

**D.** Preserve all the copyright notices of the Document.

**E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

**F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

**G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

**H.** Include an unaltered copy of this License.

**I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

**J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

**K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

**L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

**M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

**N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

**O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it
does.>
    Copyright (C) <year>  <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.
    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301
USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show
w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.