PC-IRAF Installation Guide

Doug Tody
Mike Fitzpatrick

IRAF Group
National Optical Astronomy Observatories†
January 1998

ABSTRACT

This document describes how to install or update PC-IRAF on a Personal Computer running a recent version of Red Hat or Slackware Linux, or FreeBSD. Future versions of this document will discuss installation on other platforms supported as part of the PC-IRAF project.

Both stand-alone and networked, multiple architecture configurations are described. Only those issues which one must understand to install PC-IRAF are discussed here; a companion document, *Unix IRAF Site Manager's Guide*, deals with other issues such as interfacing new devices, configuring the IRAF networking system, adding layered software, and so on.

This document covers the standard installation procedure for IRAF on UNIX systems and has been tailored for the PC-IRAF distribution. Please refer to the README file in the PC-IRAF distribution directory (PCIX) for more detailed information on platform-specific issues.

January 27, 1998

†Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

Contents

1. **Introduction** 1

2. **Installing PC-IRAF** 2
   - 2.1. Prepare the root IRAF directory 2
     - 2.1.1. If updating an existing IRAF installation... 2
     - 2.1.2. If installing IRAF for the first time 3
   - 2.2. Install the files 4
     - 2.2.1. Installing from a network distribution 4
     - 2.2.2. Installing from tape 5
     - 2.2.3. Configuring the BIN directories 6
   - 2.3. Merge local revisions back into the new system 7
   - 2.4. Run the INSTALL Script 7

3. **System Checkout** 8

Appendix A. A Complete Example 10

PC-IRAF Installation Guide

Doug Tody
Mike Fitzpatrick

IRAF Group
National Optical Astronomy Observatories†
January 1998

1. Introduction

This document covers the general procedure for installing IRAF on a UNIX system. It has been tailored for PC-IRAF and is accurate so far as the basic installation is concerned, but we do not attempt to cover the new features of PC-IRAF here. Please see the README file in the PC-IRAF distribution directory (ftp://iraf.noao.edu/iraf/v211/PCIX/README) for information on the platform specific aspects of the release or other timely information. The PC-IRAF README also includes a summary of the installation process for those who are installing from disk and who have already installed IRAF on other platforms.

The PC-IRAF release should install and run on any recent Linux or FreeBSD system. Linux is evolving rapidly however, and there are a number of different Linux distributions out there - Slackware, Red Hat, Debian, and others. These are all basically the same thing and they all share the same Linux kernel but they differ in many details regarding what non-Kernel software is included and how it is configured. FreeBSD on the other hand is controlled by a core group of programmers and there are no variant distributions which may conflict.

Unlike the earlier Linux/IRAF port, the V2.11 PC-IRAF distribution contains a single IRAF system (directory tree) which will run on a number of different operating systems, where each host system is represented as a different binary architecture. As of this writing the PC-IRAF V2.11 system contains binaries for Slackware Linux, Red Hat Linux, and FreeBSD. Other architectures will be added in a future release, e.g. Solaris x86 and MkLinux (for the PowerPC Macintosh). Our expectation is that the statically linked linux binaries will *run* on any recent Linux system, however in general it is not possible to compile programs on a Linux system unless the IRAF libraries have been tailored for and compiled on the target system, due to numerous minor differences in the different flavors of Linux. Different Linux distributions such as Slackware and Red Hat are for all practical purposes different operating systems, except that they share the same Linux kernel.

Before installing PC-IRAF, one must 1) obtain an appropriate PC-IRAF distribution, e.g., from the IRAF network archive on `iraf.noao.edu` (or by ordering a tape or CD-ROM distribution from NOAO), 2) select the server or node on which the system is to be installed and arrange for sufficient disk space to hold the system, and 3) set aside sufficient time to do the installation. If these directions are followed carefully and mistakes are avoided the basic installation should only take a few minutes on a fast PC. Additional time may be required to customize the system, to configure the local tape drives and other devices, set up IRAF networking, and so on.

The amount of disk space required to install a full IRAF system depends upon the system configuration, including the number of layered packages installed on top of the core IRAF

---

†Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

system. The main system, including both the core IRAF system and NOAO package sources, requires from 32 to over 200 Mb depending on what architectures or other options are installed. See the PC-IRAF README for more information on installation options.

2. Installing PC-IRAF

Although the details of how PC-IRAF is installed or updated depend upon the type of distribution and the desired local system configuration, the basic procedure is always the same. First one obtains the distribution files, then one follows the procedure outlined below to install the system. Most of these steps should be performed while logged in as IRAF; superuser permission is required only in the final stages of the installation, to run the *install* script.

System managers familiar with the installation of other UNIX packages should beware that IRAF has its own unique history and does not always follow the usual UNIX conventions. It is wise to read and adhere to the installation instructions to avoid problems.

```
# Prepare the root IRAF directory.
if new installation
    create iraf account
else if updating an old installation
    save locally modified files; delete old system

# Install the files.
login as iraf
unpack the core system distribution
configure the BIN directories

# Merge local revisions into new system.
if updating an old installation
    merge locally modified files back into new system

run the iraf install script to complete the installation
checkout the new system
```

If problems should arise during the installation help is available by posting a query to one of the *adass.iraf* newsgroups on USENET, by sending email to `iraf@noao.edu`, or via the IRAF HOTLINE (520-318-8160). The IRAF Web page `http://iraf.noao.edu` is a good source of information, and includes access to the IRAF FAQ list. The README file in the distribution directory is a good source of information on late-breaking developments, particularly problems found or issues arising since the initial release.

2.1. Prepare the root IRAF directory

2.1.1. If updating an existing IRAF installation...

If you are updating an existing IRAF installation then you will be replacing IRAF by the new version, and IRAF should already have an account and root directory on the desired host system. You should save any locally modified files and delete the old system, e.g., login as IRAF and enter something like the following.

```
% cd $iraf†
% tar -cf /d0/oiraf.tar local dev unix/hlib
% /bin/rm -rf *
```

There are many possible variations on this, e.g., you could use *mv* to move the above directories to someplace outside the main IRAF directory tree. Although it is probably simplest

† $iraf symbolizes the UNIX pathname of the root IRAF directory. If no "iraf" environment variable is defined just supply the actual pathname.

and safest to save entire directories as in the example, in practice only a few files are likely to have been modified. These are the following.

```
dev/graphcap
dev/hosts
dev/tapecap
dev/termcap
hlib/extern.pkg
hlib/login.cl
hlib/zzsetenv.def
local/.login, .cshrc, etc.
```

Once the old system has been deleted you are ready to install the new one, as described in §2.2. It is important to delete the old system first to avoid creating junk files or directories when the new system is installed (due to file or directory name changes or deletions). Once the new system has been restored to disk, do *not* merely restore the files saved above, as you will need to carefully merge local changes into the versions of the files supplied with the new IRAF release (more on this later).

2.1.2. If installing IRAF for the first time...

If you are installing IRAF for the first time then the first step is to set up a new account for the user '`iraf`'. This is necessary for IRAF system management, which should always be done from the IRAF account. The IRAF account has login files which set up a custom UNIX environment for IRAF system management. Having an IRAF account provides a convenient place (the IRAF system manager's login directory) to keep scratch files created during system configuration.

The location of the IRAF root directory is arbitrary. Our practice here is to locate the software in a system file storage area separate from the host operating system files (to simplify OS upgrades), and then use a symbolic link such as /iraf to point to the actual root directory. We recommend using /iraf as some of the startup files in the IRAF system account assume this (e.g. debugging support). This makes life simpler if IRAF is NFS mounted on several machines and it is later necessary to move the IRAF files. Try to keep the path to the physical IRAF root directory short to avoid filename truncation problems when IRAF is run.

The login directory for the iraf account should be $iraf/local (e.g., /iraf/iraf/local), rather than the IRAF root directory $iraf as one might expect. This is done to provide a work area for local files separate from the main IRAF directory tree, to simplify updates and make it easier to keep track of what has been locally added and what is standard IRAF. This also helps to ensure that the proper environment for the iraf account is created through the distributed .login and .cshrc files. In any case, make sure that when the IRAF account is set up the login directory is set correctly, or the IRAF environment will not be set up properly and later problems are sure to result.

A typical IRAF installation consists of the main IRAF release, a number of BIN directories (the IRAF binaries), and additional directories for layered software such as STSDAS, PROS, and so on. If sufficient disk space is available to keep everything in one area the following directory structure is recommended (note this assumes all architectures are installed):

```
/iraf/iraf                        # iraf root directory ($iraf)
/iraf/iraf/local                  # iraf login directory (~iraf)
/iraf/irafbin                     # iraf BIN directories
/iraf/irafbin/bin.freebsd         # FreeBSD binaries iraf core system
/iraf/irafbin/noao.bin.freebsd    # FreeBSD binaries NOAO package
/iraf/irafbin/bin.linux           # Slackware binaries iraf core system
/iraf/irafbin/noao.bin.linux      # Slackware binaries NOAO package
/iraf/irafbin/bin.redhat          # RedHat binaries iraf core system
/iraf/irafbin/noao.bin.redhat     # RedHat binaries NOAO package
/iraf/extern/tables               # layered package
/iraf/extern/stsdas               # layered package
/iraf/extern/xray                 # layered package
    (etc.)
```

For the purpose of this example we assume that the IRAF files are stored in /iraf; as we mentioned this is normally a link and the actual directory is arbitrary. Given this directory the IRAF root $iraf would be "/iraf/iraf/" and the login directory for the IRAF account would be /iraf/iraf/local. The Slackware Linux binaries for the core IRAF system would be in /iraf/irafbin/bin.linux, with a link $iraf/bin.linux pointing to this directory (more on this later). Similar links would exist for other architectures.

Given the above directory structure the `/etc/passwd` file entry for the IRAF account would be something like the following.

```
iraf:xxx:123:456:IRAF system account:/iraf/iraf/local:/bin/csh
```

2.2. Install the files

If you are installing from tape skip forward to §2.2.2. If you are installing from a network distribution (i.e., from hard disk or CD-ROM) continue with the next section.

2.2.1. Installing from a network distribution

PC-IRAF is available over the network via anonymous ftp from `iraf.noao.edu`, in the subdirectory `iraf/v`*nnn*`/PCIX`, where "v*nnn*" is the IRAF version number, e.g., subdirectory `iraf/v211/PCIX` for all versions of V2.11 PC-IRAF.

If IRAF is being installed from a network distribution all the architecture independent IRAF files for both the core IRAF system and the NOAO packages will be in the distribution file `as.pcix.gen`. This "file" is stored in the network archive as a directory wherein the large distribution file has been split into a number of smaller pieces, e.g.,

```
% ls as.pcix.gen
CHECKSUMS          as.pcix.gen.gz.06  as.pcix.gen.gz.14  as.pcix.gen.gz.22
FILES.Z            as.pcix.gen.gz.07  as.pcix.gen.gz.15  as.pcix.gen.gz.23
as.pcix.gen.gz.00  as.pcix.gen.gz.08  as.pcix.gen.gz.16  as.pcix.gen.gz.24
as.pcix.gen.gz.01  as.pcix.gen.gz.09  as.pcix.gen.gz.17  as.pcix.gen.gz.25
as.pcix.gen.gz.02  as.pcix.gen.gz.10  as.pcix.gen.gz.18  as.pcix.gen.gz.26
as.pcix.gen.gz.03  as.pcix.gen.gz.11  as.pcix.gen.gz.19
as.pcix.gen.gz.04  as.pcix.gen.gz.12  as.pcix.gen.gz.20
as.pcix.gen.gz.05  as.pcix.gen.gz.13  as.pcix.gen.gz.21
```

Assume that the directory `as.pcix.gen` as shown above has been recreated somewhere on the machine on which IRAF is to be installed. We can restore the main IRAF source tree as follows.

```
% whoami
iraf
% cd $iraf
% cat /path/as.pcix.gen/as.* | zcat | tar -xpf -
```

After the above finishes the root IRAF directory should appears as follows (this is for V2.11).

```
HS.PCIX.GEN  bin.generic  bin.redhat   lib          mkpkg        sys
bin          bin.linux    dev          local        noao         unix
bin.freebsd  bin.linuz    doc          math         pkg
```

The files `bin.linux`, `bin.freebsd`, and `bin.redhat` are links to the actual BIN directories (for binary executables), which may not exist yet. Configuring the BIN directories is discussed in section §2.2.3.

2.2.2. Installing from tape

If you have not already done so, log into the IRAF account so that the files when restored will belong to IRAF. Mount the distribution tape, which may be any supported tape drive, e.g. a DAT tape or Exabyte.

IRAF distribution tapes consist of multiple files separated by tape marks, with a TOC (table of contents) file as the first file on the tape. To find out what is on the tape, rewind it and read out the TOC file as follows (the actual device name may be different than that shown in the examples - on Linux systems /dev/tape is an optional link pointing to the actual tape device, e.g. /dev/nst0).

```
% mt -f /dev/tape rewind;  cat /dev/tape
```

This should cause a TOC file to be listed similar to the following, except for the file names which will vary depending upon what type of distribution you have. The example below is for a distribution of PC-IRAF.

```
0   Table of Contents

1   AS.PCIX.GEN   76.3Mb   IRAF, NOAO packages and sources
2   IB.FBSD.X86   37.1Mb   FreeBSD binaries for IRAF core system
3   NB.FBSD.X86   42.6Mb   FreeBSD binaries for NOAO packages
4   IB.LNUX.X86   34.4Mb   Linux binaries for IRAF core system
5   NB.LNUX.X86   37.1Mb   Linux binaries for NOAO packages
6   IB.RHLX.X86   37.6Mb   Red Hat Linux binaries for IRAF core system
7   NB.RHLX.X86   35.4Mb   Red Hat Linux binaries for NOAO packages
```

Here, the first column is the file number on the tape, the TOC file being file zero (the first distribution file is number one), the second column is the name of the tape file, the third column is the installed (uncompressed) disk usage for the package in megabytes, and the last column is a description of the file contents.

There are three types of tape files in the example shown: the `AS` file, which is all the IRAF sources (the core IRAF system, NOAO packages, and the host system interface), the `IB` files, or IRAF core system binaries, and the `NB` files, or NOAO package binaries, for each architecture. The NOAO package sources are included in the `AS` file since most people requesting IRAF are expected to want the astronomical reduction software, although IRAF can be configured without the NOAO packages if desired. For PC-IRAF a fourth type of distribution file exists, the `DB` file, which contains debugging version of the IRAF core system libraries; this is optional and is not required to run IRAF. All of the file objects are UNIX *tar* format files, with the exception of the TOC file which is a simple text file. The distribution files may be compressed if this was necessary to fit all the files on a tape.

The following commands would suffice to restore the main IRAF system to disk, given the distribution tape described by the TOC file in our example above. Once again, the tape device file and block size shown in the example will very likely have to be changed to whatever is needed for the tape device being used.

```
% whoami
iraf
% cd $iraf
% mt -f /dev/tape rewind;  mt -f /dev/tape fsf 1
% tar -xpf /dev/tape
```

After the above tar file read operation, the tape is left positioned to *just before the EOF of the file just read*, since *tar* stops reading the file data before reading the physical EOF. Hence, an *mt fsf* will be required to position to the next file on the tape. Any combination of *fsf* (forward skip file) or *bsf* (backward skip file) operations may be used to position to a file on a 9 track tape, DAT, or Exabyte. On a cartridge tape, it is best to plan things so that only forward file skips are used, using a rewind and forward skip if it is necessary to position to an earlier file on the tape.

Once the main system, containing only sources, is installed it is possible to create one or more empty BIN directories for the executables, then compile and link the full system. More commonly one will merely read the precompiled executables off the distribution tape, as we discuss in the next section.

2.2.3. Configuring the BIN directories

In IRAF all the files specific to any particular architecture are contained in a single directory called the BIN, or "binary", directory. To run IRAF you must install not only the AS (all-sources) directory tree, but the BIN directory for each architecture. The IRAF core system and the NOAO packages have separate BIN directories.

The BIN directories for the IRAF core system or a layered package (such as NOAO) are located, logically or physically, in the root directory of the IRAF core system or layered package. Every layered package has its own set of BIN directories. In the distributed V2.11 system you will find the following BIN files (directories or symbolic links) at the IRAF root.

```
link        bin -> bin.generic
directory   bin.generic
link        bin.freebsd  -> ../irafbin/bin.freebsd
link        bin.linux  -> ../irafbin/bin.linux
link        bin.redhat  -> ../irafbin/bin.redhat
```

If the IRAF directory structure is set up as described in §2.1.2, with $iraf located at iraf/iraf and the BIN directories stored in iraf/irafbin, then these links will not have to be modified. If a different directory structure is used you will have to modify the links accordingly.

The *bin* link and the *bin.generic* directory are required for the correct operation of IRAF and are maintained automatically by the IRAF software management utilities. *Under no circumstances should "bin" or "bin.generic" be modified or deleted*! It is a common error to manually delete the bin link and manually set it to bin.linux or some other architecture, e.g. to link external programs (people get confused by the library links in lib which point to ../bin). The bin.<arch> link can be modified as desired but bin and bin.generic should be left alone. The proper way to change the IRAF architecture (*i.e.*, set *bin* to a particular architecture) is to run "mkpkg *arch*" at the IRAF root. See the *Site Manager's Guide* for a complete discussion of package architecture support.

In the following example we will demonstrate the installation of the Slackware Linux binaries, the process is the same for the other architectures as well. Assume that the bin.linux directory has been created somewhere, e.g. in the iraf/irafbin directory, and that the `ib.lnux.x86` distribution files for the core IRAF system linux binaries have been downloaded from the network archive. We can restore the linux binaries with the following commands.

```
% cd $iraf/bin.linux
% cat /path/ib.lnux.x86/ib.* | zcat | tar -xpf -
```

Similarly, to restore the NOAO package linux binaries:

```
% cd $iraf/noao/bin.linux
% cat /path/nb.lnux.x86/nb.* | zcat | tar -xpf -
```

The procedure for restoring a BIN directory from a tape distribution is similar to that described in §2.2.2 for the core system. For example,

```
% cd $iraf/bin.linux
% mt -f /dev/tape rewind;  mt -f /dev/tape fsf 2
% tar -xpf /dev/tape
```

would restore the core system bin.linux directory from a tape containing an uncompressed `ib.lnux.x86` as file 2 on the tape.

2.3. Merge local revisions back into the new system

If this is a new IRAF installation this step can be skipped. Otherwise, once the new system has been restored to disk any local revisions made to the previous IRAF installation should be merged back into the new system. See §2.1.1 for a list of the files most likely to be affected. When propagating revisions made to these files, be sure to not replace the entire file with your saved version, as the version of the file in the new release of IRAF will often contain important additions or changes which must be preserved. It is best to merge your revisions into the version of the file which comes with the new IRAF.† This task will be easier if the revisions have been localized as far as possible, e.g., keep all `termcap` additions together at the head of the file, so that they may merely be transferred to the new file with the editor. The task of propagating revisions will also be much easier if detailed notes have been kept of all revisions made since the the last release was installed.

Beginning with IRAF version 2.8, one should no longer install locally added software in the core system LOCAL package. This significantly complicates updates and is no longer necessary as, due to the layered software enhancements introduced in V2.8 IRAF, it is now straightforward for each site to maintain their own custom LOCAL package external to the core IRAF system. The core system LOCAL is now only a *template-local* to be copied and used as the starting point for a custom LOCAL layered package. The layered software enhancements, and the procedure for building a custom LOCAL, are discussed further in the *PC-IRAF Site Manager's Guide.*

2.4. Run the INSTALL Script

Once all of the IRAF files have been restored to disk the PC-IRAF installation script must be run to complete the system installation. The install script modifies the system as necessary to reflect the new root directory and new default image storage and local bin directories, checks the mode and ownership of a number of files, installs a small set of IRAF commands in UNIX, and so on.

To make a trial run of the install script, enter the following commands:

```
% setenv iraf /path/iraf/
% cd $iraf/unix/hlib
% source irafuser.csh
% ./install -n
```

and answer the questions (don't forget the trailing '/' in the "setenv iraf"). The "-n" argument tells install to go through the motions without actually doing anything, so that one can see what will be done before committing to it.

Installing IRAF requires a few changes to be made to system directories outside the IRAF directory tree. Two fifo device entries are made in /dev. A symbolic link "iraf.h" is created in /usr/include. A number of links (cl, mkiraf, etc.) are made in /usr/local/bin or some similar directory which most users can be expected to have in their search path. The tape allocation

---

†The UNIX utility *diff* is useful for comparing files to see what has changed.

task alloc.e is made suid root (there are no known security loopholes, although we cannot make any guarantees). A symbolic link imtoolrc is created in /usr/local/lib (which is assumed to already exist).

Following one or more trial "no execute" ("-n") runs to see what the install script will do, the install script should be run without the "-n" to complete the installation. This must be done by the superuser as superuser permission is required to carry out the necessary additions to UNIX.

It is necessary to run the install script separately on each node from which IRAF will be used. If a single version of IRAF is installed on a server and NFS mounted on one or more clients, the install script must be run first on the server and then on *each* client (when installing on a client there will be warnings about insufficient permission to make changes to files on the NFS mounted partitions, which can be ignored). To install IRAF on a diskless client it may be necessary to run the install script *on the server* to do the install for the client, since the client's /usr/include and /dev directories may only be writable by root on the server. On some systems /usr is mounted read-only, and must be unmounted and remounted read-write before doing the installation to allow an entry to be made in /usr/include. Once the installation is complete the default mount access mode may be restored.

The exchange with the install script will be along the lines of the following (this example is for a sparc server):

```
% ./install -n
new iraf root directory (/iraf/iraf):
default root image storage directory (/d0/iraf):
local unix commands directory (/usr/local/bin):
install iraf for machine type linux
old iraf root = /usr/iraf, old imdir = /d0/iraf
installing iraf at /iraf/iraf, imdir=/d0/iraf, lbindir=/usr/local/bin
proceed with installation? (yes):
```

The "iraf root directory" is the value of $iraf (minus the trailing '/'in this case). The "root image storage directory" is the default place to put image data for users; the program may prompt with /tmp if it cannot find any likely looking data storage areas on your system, but /tmp is not a good place to put image data as the contents can be deleted whenever the system reboots. The value entered should be the path to a public iraf subdirectory of a designated data or scratch disk on your system. Lastly, the "local unix command directory" is where the UNIX callable IRAF startup commands will be defined. This should be a UNIX directory which is in the default path of anyone who might want to use IRAF; /usr/local/bin is the most common value.

After answering with "yes" or hitting return in response to the "proceed with installation" query, the script will issue a series of messages as it checks the system and performs the installation, possibly answering additional questions in the process.

3. System Checkout

The basic IRAF system should be usable once the files have been restored to disk, the binaries have been configured or generated, and the install script has been run. To verify that the basic system comes up and runs successfully, login as iraf and startup the CL (IRAF command language) from the iraf account. You should be able to login as IRAF and type "cl" to start IRAF, using the login files which come with the distributed system.

The IRAF system account for PC-IRAF has been customized for Linux and FreeBSD and should autoconfigure for whichever system is being run. Some default window system startup files are provided, but the iraf account is configured mainly to run in an *xgterm* or *xterm* (shell-oriented) window. This account is intended for IRAF system management and as a working example showing how to use the various system components. Normal IRAF use should be done from a user account configured using the examples given with the IRAF system account.

```
% login iraf
% cl
```

To more thoroughly test the installation it is a good idea to set up a user account to run IRAF and test IRAF from this user account. To do this you login to a user account and run the *mkiraf* task to set up the IRAF login files. This will create or initialize the user's `uparm` (user parameter) directory, and create a new `login.cl` file. It may also be desirable to edit the user's `.login` file to modify the way the environment variable `IRAFARCH` is defined. This variable, required for software development but optional for merely using IRAF, must be set to the name of the desired machine architecture, e.g., 'redhat' for Red Hat 5.0 systems, 'linux' for all other Linux distributions, and 'freebsd' for FreeBSD systems.

```
% mkiraf
Initialize uparm? (y|n): y
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc."
Enter terminal type: xgterm
A new LOGIN.CL file has been created in the current directory.
You may wish to review and edit this file to change the defaults.
```

The *cl* command should now start up the CL, which will clear the screen and print out a startup message. The standard test procedure included in Volume 1A of the *IRAF User Handbook* should be run to verify the installation.

Appendix A. A Complete Example

Assume we are installing IRAF for the first time on our spiffy new PC. The IRAF directories will be located at /u1/iraf using a symbolic link /iraf to point to this location. We will configure only the linux binaries, locating the BIN directories in the directory /iraf/irafbin (installing other architectures is just a matter of unpacking the binaries and creating the links). The local user commands will be placed in /usr/local/bin. We will be installing from a network distribution with the distribution files located in /d0.

The first step is for the superuser to create an account for the fictitious user '`iraf`', with home directory /iraf/iraf/local and shell /bin/csh. The /u1/iraf directory is created owned by IRAF, and pointed to by the link /iraf. We then login as IRAF (a warning message will be printed since there is no login directory) and proceed as follows.

```
% whoami
iraf
%
% setenv iraf /iraf/iraf/                 # set root directory
% mkdir /iraf/iraf
%
% cd $iraf                                # unpack main IRAF distribution
% cat /d0/as.lnux.gen/as.* | zcat | tar -xpf -
%
% cd /iraf                                # create BIN directories
% mkdir irafbin
% mkdir irafbin/bin.linux
% mkdir irafbin/noao.bin.linux
%
% cd $iraf/bin.linux                      # unpack core bin.linux
% cat /d0/ib.lnux.x86/ib.* | zcat | tar -xpf -
%
% cd $iraf/noao/bin.linux                 # unpack NOAO bin.linux
% cat /d0/nb.lnux.x86/nb.* | zcat | tar -xpf -
%
% cd $iraf/unix/hlib                      # run the INSTALL script
% source irafuser.csh
% ./install -n
% su
# ./install
# exit
%
% cd
% source .login                          # read new .login
% rehash                                 # pick up new iraf commands
% cl                                     # verify that the CL runs
```

This will fully install IRAF on a server or a standalone system. If this version of IRAF will be accessed via NFS by client nodes then the IRAF install script must be run on each client node as well. Installing IRAF does not allow one to access local tape drives, printers, and so on. Refer to the *PC-IRAF Site Manager's Guide* for information on how to configure IRAF for the local site.