LINUX Installation Guide for GAMIT/GLOBK Users

1. Introduction.

The Linux operating system is a Unix look-alike devised and built in 1991 by Linus Torvalds, then a computer science student in Helsinki, to meet his personal needs of a compact operating system that had "grunt". Assisted by gratuitous contributions from Net users world-wide, it has matured into a robust alternative both for DOS and Windows users and users of full commercial Unix systems. Its most appealing assets are the low cost of hardware (Intel platform) and system software (mostly free), and the power it delivers for the expert user. The drawbacks are the plethora of Linux versions around the lack of corporate backing for patches, updates, and the gamut of other services that inflate the cost of conventional software.

If you are willing to invest some time in installing the Linux operating system, along with X-windows and a C and Fortran compiler on your PC, you can run GAMIT and GLOBK quite efficiently. For example, the Pentium Pro chip running at 200 MHz is comparable in speed to an HP 735 at a fraction of the cost, and 4 times faster than a Sun Sparc 5. A Linux machine can be networked successfully with other workstations, binary files (GAMIT C- and T-files and GLOBK solution files) will not be compatible with those produced on Sun and HP machines (but will be compatible with those written on a DEC). It is possible to maintain both Linux and a DOS Windows 95 system on the same PC if there is sufficient disk space available (Windows 95 requires up to 500 Mb depending on the type of installation).

To successfully install Linux and the required compilers, you will need to follow closely the instructions provided with the versions you obtain. The purpose of this memo is to provide some initial insight into the choice versions and an outline of the steps involved.

Incidentally, Torvalds pronounces his name "LEEN-oos", so the appropriate pronunciation of his baby is "LEEN-ux", not "LINE-ux".

2. Hardware Requirements

The Linux operating system for Intel X86 cpu's requires the following hardware components:

1. CPU's: All Intel 386DX and above CPU's are supported. We recommend that you consider the more recent 586 and 686 CPUs with a preference for the 686. The X86SX processors must be supplemented with a math coprocessor. We do not recommend that you upgrade an old SX CPU for GAMIT/GLOBK processing.

2. Memory: The minimum recommended memory is 16 Mbytes of RAM. This is not a Linux restriction as Linux will compile in 4 Mbytes but a general system requirement. In general the more memory the better. Certainly GAMIT and GLOBK will appreciate the extra memory. This extra memory will reduce paging and hence provide speed and throughput for the user. We recommend at least 32Mb.

3. System Bus: Linux supports the ISA and EISA busses found in the older Intel PCs and the newer local and/or VESA bus system used in some 486s to improve communication between the hardware peripherals and the CPU. Most high end Pentiums now use the PCI bus. This is also supported by Linux. GAMIT and GLOBK make extensive use of files. Thus it is the users interest to transfer data.
between the CPU and the peripherals as fast as possible. We recommend that users choose a system with the PCI bus.

4. Disk Space: Linux is frugal on disc space. It is possible to construct an installation in 20 Mbytes. However this system will not support the GAMIT/GLOBK suite. You need a minimum of 200 Mbytes for such a system, including the X-windows libraries. Other system elements such as the GNU C library require a further 100 Mbytes. Then there is the GAMIT/GLOBK suite requiring about 60 Mbytes. We recommend that your minimum disc capacity be 1 Gb.

Linux supports most common controllers including those implementing the SCSI standards, including the wide SCSI standard. Wide SCSI discs have double the width which dramatically improves throughput. SCSI suppliers for which drivers exist include Seagate, UltraStor, Western Digital, Adaptec and Future Domain. Other disc suppliers for which drivers exist include Qlogic, Seagate, Buslogic, Adaptec and NCR. Like Unix, Linux supports multiple drives so it is possible to spread your system across a number of volumes.

A significant issue with the hard disc is the assignment of swap space. The general rule of thumb is that it must be at least twice the size of the main memory. Linux uses a predefined swap space which cannot be encroached upon by the system. The maximum size is 128 Mbytes consisting of eight 16 Mbyte partitions. With 64 Mb of RAM and 128 Mbytes of swap, you will be able to run the largest of the GAMIT modules (CVIEW) dimensioned for up to 40 stations and 2880 epochs.

5. The Monitor: Linux will support all the common standards, CGA, EGA, VGA and Super VGA. However, in order to run X-windows for CVIEW you will need a video card. We recommend that the monitor be at least a 15 inch multi-sync color monitor with at least 1 Mb of RAM. The following video chip sets are supported by LINUX: Tseng ET3000, ET4000/W32 and ET4000AX; Western Digital, most of the WD90C?? series; Trident, most of the TVGA? Series; NCR: 77C22, 77C22E, 77C22E+; Cirrus Logic: most of CLGD? Series; S3, most of 86C? series; Compaq AVGA; Raradise PVGA1; MATRCX MG???. X-Free86 is continually being upgraded and augmented, so the above list is neither definitive nor up to date.

6. CD-ROMs CD-ROMS are considered part of most systems partly because of the ease with which CD-ROM containing the software can be acquired and used. We recommend that you use a SCSI CD-ROM and that you attach it to the SCSI controller that works with your disc drive. However Linux also supports the new EIDE and ATAPI CD-ROMs. Beware of propriety drivers and CD-ROMS included with multimedia packages. Some work and some do not. The Creative Labs Soundblaster line is supported. The following partial list are known to be supported: NEC CDR--541, SONY CDU-541, SONY CDU-31a and CDU-33a, Aztech, Orchid, Okano, Matsushita, Mistsumi.

7. The Mouse Linux supports most mice. The most common are the Logitech and the Microsoft. For ease of use with CVIEW and X-windows in general, you should get a 3-button mouse.

8. Network Linux supports network access via the the IEEE803 standard for the Ethernet. There is Linux support for 3Com, Novell, Western Digital and Hewlett Packard.

9. Modems Linux supports the SLIP and PPP standards.
3. Choosing Linux Software

There are a plethora of sources for Linux software, in a variety of flavors and levels of support. The most popular versions are Slackware, Red Hat, and Debian. All three of these are available in a single 6 CD-ROM set from InfoMagic.

Slackware The home site for slackware is http://www.cdrom.com. A complete list of ftp sites can be found in http://unr.edu/homepage/ccs/SlackwareSites.html. It is available in the following languages: English, French, Italian, German, Japanese, and others. There is no charge for Slackware via ftp; it can be obtained also on CD-ROM at a nominal charge.


Debian There is no "official" Debian GNU/Linux CD-ROM. There are several major contributors from whom you can get CD-ROMS and ftp copies. A source of information is http://www.us.debian.org/. This web site also has links to ftp sites from which a copy may be downloaded. Debian is also available from Walnut Creek CDROM, the home of Slackware.

Other Sources Perhaps the most significant academic/research source of information, including software is the University of North Carolina's Sunsite, http://sunsite.unc.edu/. The depth and breadth of UNC's holdings are indicated by the following topics:
- General Linux Information - General and introductory Linux information.
- Linux Documentation Project and HOWTOs - Information about the Linux Documentation Project, and listing of LDP resources.
- Linux Development Projects - Listing of Linux-related development projects (hardware ports, software, research areas, etc.).
- Commercial Linux Products - Commercial Linux distributions, books, software and hardware.
- Linux Links - Lots of links to all sorts of Linux information.
- Powered by Linux! - See what Linux can do for you. Listing of some sites using Linux.
- Linux Users Groups - World-wide listing of Linux users groups.
- Linux Usenet Newsgroups - Quick list of Linux newsgroups with descriptions.
- LDP Mirrors - Mirrors of these pages.

The range of information and the extent of the HOWTO documents, part of the Linux Documentation Project, is the same as those found in the specific documents of Walnut Creek CDROM and InfoMagic.

The following is a list of the HOWTO documents with a brief explanation of their contents:

- Bootdisk-HOWTO How to create a boot/root maintenance disk for Linux
- Busmouse-HOWTO Information on bus mouse compatibility with Linux
- CDROM-HOWTO Information on CD-ROM drive compatibility for Linux
- COPYRIGHT HOWTO Copyright Information
- Commercial-HOWTO Listing of commercial software products for Linux
<table>
<thead>
<tr>
<th>HOWTO</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSEMU-HOWTO</td>
<td>HOWTO about the Linux MS-DOS Emulator, DOSEMU</td>
</tr>
<tr>
<td>Danish-HOWTO</td>
<td>How to configure Linux for use with the Danish characterset</td>
</tr>
<tr>
<td>Distribution-HOWTO</td>
<td>A list of mail order distributions and other commercial services</td>
</tr>
<tr>
<td>ELF-HOWTO</td>
<td>Information on ELF binaries for Linux</td>
</tr>
<tr>
<td>Ethernet-HOWTO</td>
<td>Information on Ethernet hardware compatibility for Linux</td>
</tr>
<tr>
<td>Firewall-HOWTO</td>
<td>How to set up a firewall using Linux</td>
</tr>
<tr>
<td>Ftape-HOWTO</td>
<td>Information on ftape drive compatibility with Linux</td>
</tr>
<tr>
<td>German-HOWTO</td>
<td>Information on using Linux with German-specific features</td>
</tr>
<tr>
<td>HAM-HOWTO</td>
<td>HOWTO configure amateur radio software for Linux</td>
</tr>
<tr>
<td>HOWTO-INDEX</td>
<td>Index of HOWTO documents about Linux</td>
</tr>
<tr>
<td>Hardware-HOWTO</td>
<td>A list of hardware known to work with Linux</td>
</tr>
<tr>
<td>INFO-SHEET</td>
<td>Generic introduction to the Linux operating system</td>
</tr>
<tr>
<td>Installation-HOWTO</td>
<td>How to obtain and install the Linux software</td>
</tr>
<tr>
<td>JE-HOWTO</td>
<td>Information on JE, a set of Japanese language extensions for Linux</td>
</tr>
<tr>
<td>Kernel-HOWTO</td>
<td>Upgrading and compiling the Linux kernel</td>
</tr>
<tr>
<td>META-FAQ</td>
<td>A listing of Linux sources of information</td>
</tr>
<tr>
<td>MGR-HOWTO</td>
<td>Information on the MGR graphics interface for Linux</td>
</tr>
<tr>
<td>Mail-HOWTO</td>
<td>Information on Linux-based mail servers and clients</td>
</tr>
<tr>
<td>NET-2-HOWTO</td>
<td>HOWTO configure TCP/IP networking, SLIP, PLIP, and PPP under Linux</td>
</tr>
<tr>
<td>NIS-HOWTO</td>
<td>Linux NIS network information service, yellow pages, YP</td>
</tr>
<tr>
<td>News-HOWTO</td>
<td>Information on USENET news server and client software for Linux</td>
</tr>
<tr>
<td>PCI-HOWTO</td>
<td>Information on PCI-architecture compatibility with Linux</td>
</tr>
<tr>
<td>PCMCIA-HOWTO</td>
<td>How to install and use PCMCIA Card Services</td>
</tr>
<tr>
<td>PPP-HOWTO</td>
<td>Information on using PPP networking with Linux</td>
</tr>
<tr>
<td>Printing-HOWTO</td>
<td>Information on how to set up printing under Linux</td>
</tr>
<tr>
<td>Printing-Usage-HOWTO</td>
<td>How to use the printing system for a variety of file types and options</td>
</tr>
<tr>
<td>SCSI-HOWTO</td>
<td>Linux SCSI drive tape CD-ROM HOWTO</td>
</tr>
<tr>
<td>SCSI-Programming-HOWTO</td>
<td>Information on programming the generic Linux SCSI interface</td>
</tr>
<tr>
<td>Serial-HOWTO</td>
<td>Information on use of serial devices and communications software</td>
</tr>
<tr>
<td>Sound-HOWTO</td>
<td>Sound hardware and software for the Linux operating system</td>
</tr>
<tr>
<td>Term-HOWTO</td>
<td>HOWTO use the `term' communications package on Linux systems</td>
</tr>
<tr>
<td>Tips-HOWTO</td>
<td>HOWTO on miscellaneous tips and tricks for Linux</td>
</tr>
<tr>
<td>UMSDOS-HOWTO</td>
<td>How to install and use the UMSDOS filesystem</td>
</tr>
<tr>
<td>UPS-HOWTO</td>
<td>Information on using a UPS power supply with Linux</td>
</tr>
<tr>
<td>UUCP-HOWTO</td>
<td>Information on UUCP software for Linux</td>
</tr>
<tr>
<td>WRITING</td>
<td>Information on writing a HOWTO</td>
</tr>
<tr>
<td>XFree86-HOWTO</td>
<td>How to obtain, install, and configure XFree86 3.1.1 (X11R6)</td>
</tr>
</tbody>
</table>

Academic and research libraries have generally been reluctant to acquire books of computer system material since they are quickly outdated. Those that are available for Linux fall into two categories: 1) higher level reference and generic material that describe Linux principles, theory and philosophy of implementation; and 2) a mixture of procedures and theory generally directly connected with a specific implementation. The following list is comprehensive but not exhaustive.
4. Installing the Slackware Version of Linux

As an example of the steps and potential problems involved in a Linux installation, we describe installation of the Slackware version as provided by the April 1997 LINUX Developer's Resource distributed by Info Magic. We acquired the LINUX TOOLBOX from Info Magic, (web:http://infomagic.com or email:info@infomagic.com). This toolbox consists of a 6 cd-rom set, the book Running LINUX, and a small quickstart guide, QUICKSTART version 3.1.

The installation of the Slackware flavour of Linux from this resource presupposes that you have access to a conventional MS-DOS/Windows machine that can generate the "Slackware
Boot Disk" and the "Slackware Root Disk". We envisage sites that will not be able to
mount the cd-roms for a wide variety of reasons including the desire not to get a Windows
95 license. Consequently these disks can be supplied by MIT or by Peter Morgan. These
two disks do need to be tailored to the hardware environment of the platform on which
Linux is to be installed. In particular the necessary drivers for both a SCSI bus system,
recommended for GAMIT users, and the conventional IDE bus system will not fit on a
single 1.44 Mb diskette.

The following is a list of directory structure of cd-rom 2: Slackware V3.2.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOOTING.TXT</td>
<td>Filelist of the boot disk</td>
</tr>
<tr>
<td>FILELIST.TXT</td>
<td>Filelist of the filelist</td>
</tr>
<tr>
<td>TRANS.TBL</td>
<td>Table of the Transfers</td>
</tr>
<tr>
<td>install/</td>
<td>Directory of the install</td>
</tr>
<tr>
<td>slakware/</td>
<td>Directory of the slakware</td>
</tr>
<tr>
<td>CDROM.TXT</td>
<td>Directory of the CDROM</td>
</tr>
<tr>
<td>INSTALL.GER</td>
<td>Directory of the Install GER</td>
</tr>
<tr>
<td>UPGRADE.TXT</td>
<td>Directory of the Upgrade</td>
</tr>
<tr>
<td>kernels/</td>
<td>Directory of the kernels</td>
</tr>
<tr>
<td>slakware.doc/</td>
<td>Directory of the slakware.doc</td>
</tr>
<tr>
<td>COPYING</td>
<td>Directory of the Copying</td>
</tr>
<tr>
<td>INSTALL.TXT</td>
<td>Directory of the Install</td>
</tr>
<tr>
<td>bootdsks.12/</td>
<td>Directory of the bootdsks.12</td>
</tr>
<tr>
<td>locatedb.2</td>
<td>Directory of the locatedb.2</td>
</tr>
<tr>
<td>source/</td>
<td>Directory of the source</td>
</tr>
<tr>
<td>COPYRIGHT.TXT</td>
<td>Directory of the Copyright</td>
</tr>
<tr>
<td>LOWMEM.TXT</td>
<td>Directory of the LOWMEM</td>
</tr>
<tr>
<td>bootdsks.144/</td>
<td>Directory of the bootdsks.144</td>
</tr>
<tr>
<td>ls_lr2</td>
<td>Directory of the ls_lr2</td>
</tr>
<tr>
<td>v2.1/</td>
<td>Directory of the v2.1</td>
</tr>
<tr>
<td>ChangeLog.txt</td>
<td>Directory of the ChangeLog</td>
</tr>
<tr>
<td>README32.TXT</td>
<td>Directory of the README32</td>
</tr>
<tr>
<td>contrib/</td>
<td>Directory of the contrib</td>
</tr>
<tr>
<td>rootdsks/</td>
<td>Directory of the rootdsks</td>
</tr>
</tbody>
</table>

(Note: These listings were extracted using Linux. The case of the files and directories is
different under MS/DOS. MS/DOS, like Linux, is case sensitive. You must use the
appropriate convention.)

To make the boot disk it is necessay to enter the boot directory bootdsks.144.

A listing of this directory now follows:

```
7000fast.s  am53c974.s  cdu535.s  goldstar.i  n_5380.s  sbpcd.i
RAWRITE.EXE  aztech.i  cm206.i  goldstar.s  n_53c7xx.s  sbpcd.s
README.TXT  aztech.s  cm206.s  hpfs.i  net.i  scsi.s
TRANS.TBL  bare.i  dtc3280.s  in2000.s  optics.i  scsinet.s
WHICH.ONE  bareapm.i  eata_dma.s  iomega.s  optics.s  scsipnp.s
advansys.s  barepnp.i  eata_isa.s  mcd.i  pas16.s  seagate.s
aha152x.s  buslogic.s  eata_pio.s  mcd.s  qlog_fas.s  trantor.s
aha1542.s  cdu31a.i  fat32.i  mcdx.i  qlog_isp.s  ultrastr.s
aha1740.s  cdu31a.s  fat32.s  mcdx.s  sanyo.i  ustor14f.s
aha2x4x.s  cdu535.i  fdomain.s  n53c406a.s  sanyo.s  xt.i
```

There are two types of files in this list. The .i files are for IDE platforms. A boot disk of an
IDE system is readily made by entering the following commands commencing from a DOS
prompt in the cd-rom directory assumed to be mounted at D.

D:\> cd bootdsks.144
D:\> rawrite

Enter disk image file name: bare.i
Enter target diskette drive a:
Please insert a formatted disk into drive A: and press-Enter-:

In order to make the correct SCSI boot disk it is necessary to know what is the make and model of the SCSI adapter. Most SCSI adapters are made by Adaptec. The current series are 29xx. The installation described in this document used an Adaptec 29XX series and thus the appropriate command sequence was:

D:\> cd bootdsks.144
D:\> rawrite
Enter disk image file name: aha2x4x.s
Enter target diskette drive a:
Please insert a formatted disk into drive A: and press-Enter-:

It now remains to make the "Slackware root" disk. This root disk has a number of functions including the ability to rescue your system from special problems. The procedure is similar to that detailed for the boot disk. It is first necessary to place yourself in the rootdsks directory. The directory listing is as follows:

RAWRITE.EXE   README_T.APE   pcmcia.gz       text.gz
README        TRANS.TBL     rescue.gz       umsdos.gz
README.UMS    color.gz      tape.gz

The required file is color.gz which contains a nice 16 colour menu interface for entering options. The sequence of commands to place this file on the appropriate floppy is:

D:\> cd rootdsks
D:\> rawrite
Enter disk image file name: color.gz
Enter target diskette drive a:
Please insert a formatted disk into drive A: and press-Enter-:

Before building your Linux system you must make some decisions about the nature of your file system. In particular you must make a decision as whether the hardware is to be configured for a single operating system, Linux, or a mixed system in which Windows 95 and Linux will coexist. You must also be aware that some of the installation procedures will "destroy" parts or all of your current hard disk file system. If the hardware is "new", this is not generally a problem. Figure 1 (not yet included) shows some of the common decisions that need to be made. It must be stressed that the Linux version of fdisk can only operate on Linux partitions just as the MS/DOS version must be used on MS/DOS partitions. The use of the Linux version of fdisk on a MS/DOS partition will destroy the contents of that partition.
There are some significant gains in operating a mixed system, especially when only one high order PC is available. For example, receiver observation files can be decoded on the Windows 95 side of the PC and seen from the Linux side for processing. The operation of a mixed system, in this case fully supported, is generally not recommended by most system managers. A reasonable Windows 95 system will generally require about 500 Mb of disk. It is recommended that you make the Windows 95 partition the lowest numbered partition or the first disk in a multi disk system.

You must partition the disk before you commence the setup procedure. In general, unless you want to resize your partitions, you only need to run fdisk once. Linux demands at least two partitions—the swap partition and one other. It is common for the swap partition to be the first partition. Linux doesn't support swap partitions greater than 128 Mbytes. A good rule of thumb is that this partition should be twice the installed main memory, but if you have less than 64 Mb of RAM, you may need to exceed this ratio to obtain adequate total memory to run GAMIT and GLOBK.

Most Linux systems then have one partition for the system and the user areas. We recommend that you design a system with specific partitions for specific tasks. This is especially so in a multi-user environment since it prevents the system from running out of space for common group functions such as editing, mail, spooling etc. The maximum number of partitions allowed is 8, with each byte of a 32-bit word used to label each partition. The first byte is used to define the type of bus architecture. Thus it will be s for SCSI or h for a conventional IDE drive. The second byte is used for the device, d is from drive while the third is used to indicate the number of devices. In a system with a single SCSI disc the drive will have the label of sda. In a system with two SCSI drives the first is labelled sda while the second is labelled sdb. Since we are dealing with bytes we have eight possible partitions. Linux is organised so that the low, first four partitions are real or absolute while the high or last four partitions are relative. Thus a single drive mixed system will look like:

<table>
<thead>
<tr>
<th>Partition</th>
<th>Size</th>
<th>Reference</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>sda1</td>
<td>500 Mb</td>
<td></td>
<td>Windows 95 system including Office</td>
</tr>
<tr>
<td>sda2</td>
<td>128 Mb</td>
<td>82</td>
<td>Linux swap space</td>
</tr>
<tr>
<td>sda3</td>
<td>tbd</td>
<td>83</td>
<td>LINUX root or other space to be named</td>
</tr>
<tr>
<td>sda4</td>
<td>tbd</td>
<td>83</td>
<td>LINUX Extended</td>
</tr>
</tbody>
</table>

Thus it is possible to have only one named partition, usually associated with the root kernal, if partitioning into logical partitions for users, software and system functions is required. These logical partitions all exist with the space defined for the extended area. A large complex LINUX system utilizing non standard hardware and or many hardware components will need up to 350 Mb for the root area whereas a stripped down systems will require less than 100 Mb for this area. The extended area then contains all the remaining disk space.

It is now seen that only four logical partitions can be generated within the extended partition. Thus some compromise is necessary. In order to understand these issues Figure 2 (not yet included ) is presented which shows a typical Unix/Linux file system.

It is recommended that the extended or logical file system be invoked and that consideration be given to the establishment of partitions associated with /usr, /tmp and /home.
good idea if additional drives are available for data is a partition /usr/src to hold software sources which can undergo an independent updating cycle.

The `fdisk` routine is quite cryptic to run. Please use the supplied help system which is activated with "m". Note that "p" can be used in a repetitive manner to view your progress. You will also need to change the reference number associated with you Linux swap partition. Enter "t" to do this.


**C. Installation**

Installing LINUX is relatively simple using the supplied menu driven tools. The main menu is activated by typing "setup". You are stepped through this menu. Unfortunately there are no recovery procedures so care selecting items should be exercised. The system marks all defaults required so that apart from the selection of the type of kernel you are only supplying optional or enhancement routines. The first task to be completed is the initialization of the swap space. This is item 3 in the menu. Follow the menu. You are automatically lead to item 4 in the menu after completing the installation of the swap space. You are now asked to initialize your partitions. You must have a root partition. You are asked to assign logical names, including root, "/", to hardware partitions. We recommend you use the default sizes and that you perform full error and bad block checking on any new system. You are then lead into the selection menu. Before getting to the Selection process you will need to answer questions as to where the source code is to be placed, target, and where to find the source. The target should be the hard disk while the source is a CD-ROM.

Slackware is organised along a series line which has its origins in the days when it was distributed on floppy disks. We recommend that you select A, AP, D, E, N, T, X, XAP, XV. You will only be offered packages with each group selected at this stage.

The following is meant to be a guide as to the packages that you will need to run an effective GAMIT/GLOBK system with auxilaries for analysing GPS data and communicating with your colleagues. It is not exhaustive and can be considerably tailored to the individuals needs and requirements.

1. CUS The custom disk sets. In general you will not have any of these.
2. A The Base LINUX system. Deselect the ide Kernal but make sure you select scsi kernal. recommended additions are: ibsc2, getty, scsimods.
3. AP Various applications that do not use X. Recommend that you add ispell, sudo, ghostscr, jpeg, bc, groff.
4. D Program Development (C, C++, etc). Recommend that you add objc2721, g77, gdb, gcl, m4, man, ncurses, perl, rcs, svgalib, strace, terminfo.
5. E GNU EMACS. No additions required.
6. F FAQ Lists, HOWTO documentation. Not recommended at initial building. It is assumed that you will have a hard copy of the important HOWTO documents in one of the class 2 manuals. The manual *Using LINUX*, highly recommended for Slackware users, contains these documents. Once you know the limits of your system then feel free to install this group.
7. Networking (TCP/IP, Mail, etc). Recommend that you add procmail, dip, ppp, rdist, pine, smailcfg. If news is important then you should select the following until you make a decision as to which system you are most comfortable with, cnews, inn, tin, trn, trn-nntp, nn-spool, nn-nntp, netpipes.


9. The TeX documentation system. If you are into TeX then you should select ALL the options in this item. Note that TeX formatted documents are accepted/required by many journals.

10. TCL Tcl/Tk/TclX language and toolkits. Not recommended.

11. XFree.86 X windows system. Please be careful in this menu. There are no selected defaults. It is recommended that you select the X-server appropriate for your card and one of the generic servers. The "best" generic server is 32svga. Note this menu presumes that you know the make, model and chip set used on your video card. A second menu is presented in this section that requests general information. It is recommended that you select x32doc, fnon and x32fcyr. If you are going to use web and html then you will need to install these options.

12. X Applications. We recommend that you add gnuplot, xfm, xpaint, xspread.


14. XView (Open look window managers). No additions are necessary unless active development is to take place.


Once you have installed these packages from the CD-ROM to your hard disk you will be asked to install the Linux kernel and make the LILO, the Linux loader. The menu should be followed. Note that you will need your initial Linux boot disk to start this process. You should use a third floppy to hold the floppy copy of the Linux loader which will enable you to boot from a floppy if necessary. Please be careful in installing LILO to ensure that you make both the Windows 95 and Linux bootable.

It is recommended that you allow the installation process to format your diskette and that you then make your lilo boot disk. Pressing continue will then walk you through setup procedures for a modem, mouse, and custom fonts. It is to be noted that skipping over these procedures is not damaging as they can be installed at a latter stage by activating the appropriate installation program.

Eventually you are brought to a LILO installation menu which must be started with the "Begin" sequence. After each sequence you are returned to the menu. You then skip down to the next item that is required and again follow the procedure. In a mixed system you will need to enter Linux and DOS. Finally Install is selected. The procedure asks for network information and then time zone information. Both of these items are also reinstallable by post installation procedures. You are quickly presented with a screen indicating that the installation process is complete and that you may exit and reboot your machine.

We caution you not to think that you have finished the process. You still need to bring up the X Windows system and many of the ancillary packages but you do indeed have a basic system. This basic system MUST be protected. Please reboot the hardware and logon as root and assign root a password using passwd.

D. Adding users
This is an essential part of the installation process. It is considered bad practice to run "normal" operations as root, hence even single user systems need to establish user accounts. It is unfortunate that many Linux/Unix administrators create user accounts by hand and are sufficiently proud of this skill to say so in their guides on system administration without providing sufficient information on the process. It is also unfortunate that Version 3.2 of Slackware is not robust in this area.

The full procedure for adding users is as follows:

1. Ensure that all the files to be transferred to the user's home directory at establishment time are resident in etc/skel. The minimum set includes the following files:
   
   .Xdefaults .Xmodmap .fvwm2rc95 .less .lesskey .lessrc
   
   If these files are not present in /etc/skel then they can be copied there by root from his home directory with the command
   
   cp ./.* /etc/skel/.
   
   Note that this set does not include any customization and path setting script.

2. Add the new user to /etc/group. This assigns the group, GID, identifier. The format of each line in the file is
   
   group name:password:GID:members
   
   Generally the password is not active. A * is inserted in this field. Users are separated by commas. It is recommended that a number above 100 be used for all GIDs set up by the system administrator.

3. Add the new user to /etc/passwd. The format for entries in the password file is
   
   user name:password:UID:GID:full name:home directory:login shell
   
   In systems that use the shadow password concept, LINUX is one such system, the password field is supplied with * and the encrypted password entered into /etc/shadow. Systems assign UID's according to various rules. It is recommended that you start with a large number, say 1000.

4. Add the user to /etc/shadow. The format for entries is
   
   login name:password:fields describing the time limits for user and password.
   
   It is recommended that you enter, login name:*:9797:0::::: and exit /etc/shadow. You resolve the default password entry by running the password script with the new user's login name as the first argument. Many system administrators use a change_me type password in executing this command.

5. Create the home directory for this new user with
   
   mkdir /home/new_user

6. Copy the startup file in /etc/skel into this new user's home directory
   
   cp /etc/skel/.* /home/new_user/.

7. Transfer ownership of this new directory from root to the new user
   
   chown new_user /home/new_user

The supplied script, adduser2, which resides in sbin accomplishes steps 2 through 7 in a simple prompt manner. Don't use adduser, as recommended in all the guides, as this script clearly is out of date and non functional with Slackware Version 3.2.
5. Installing C and Fortran Compilers

The logical choice for compilers are the “GNU” versions developed by the Free Software Foundation:

Free Software Foundation, Inc. Telephone: +1-617-542-5942
59 Temple Place - Suite 330 Fax: (including Japan) +1-617-542-2652
Boston, MA 02111-1307 Free Dial Fax (in Japan):
USA 0031-13-2473 (KDD)
Electronic mail: gnu@prep.ai.mit.edu 0066-3382-0158 (IDC)

The site prep.ai.mit.edu also is an anonymous ftp site with all the GNU software in the directory /pub/gnu. This is a large directory so it is advisable to get this listing into a file or be sure you know what you are searching for. The C and Fortran compilers bundled in two .gz files; e.g. gcc-2.7.2.2.tar.gz and g77-5.20.tar.gz. The software can also be obtained on CD-ROM for a small fee which covers the cost of the CD-ROM and their effort to send it to you.

The G77 compiler must be itself compiled using the lastest GCC, which itself must be compiled with a more primitive C compiler included with your Linux distribution. To begin the installation, you should place a copy of the Linux C compiler in /usr/bin (which is where GNU will place its version if you use the recommended defaults). Then extract the directories and files of the Gnu distribution using gunzip filename.tar.gz and and tar xvf filename.tar. This produces under /usr/local/src the directories gcc and g77 with the following structure:

```
grcc/gcc-2.7.2.2
  config cp ginclud objc stage1 stage2 f many files
```

```
g77/g77-0.5.20
  COPYING. g77 README.g77 f
  gbe runtime INSTALL
  many .h, .c, .info files
```

Under gcc are many files ending with .c, .bat, .com, and .h, plus four additional directories:

config is a directory that contains only sub-directories leading to the hardware types supported by the gcc compiler. The directory has a README file that makes this statement. The machine named sub-directories do not have README or information
files. However, it is seen that there are file with subscripts corresponding to the various compiler options invoked. That is there is a function called .as for the option to compile with the GNU assembler.

`cp` is the C++ directory. It contains files associated with this object oriented compiler.

`ginclude` is a directory that contains machine (host and target) dependant code. The files include the stdio files and math files needed by the various hardware specified in the configuration process.

`objc` is the Objective C subdirectory. It contains a small `README` file explaining the rationale behind the directory and its parentage.

The most important document is the text file `INSTALL`. Other text files are meant to be compiled along with the code into a searchable document which is accessed after the compilation has succeeded.

It is very important to know what is the exact configuration of your system. In general the system is characterized by three variables: CPU, manufacturer, and operating system. For most applications the CPU will be one of ix86 but GNU also supports SUN and HP and other hardware. These notes are geared to an Intel system. The next two parameters are much harder. For Intel machines we recommend you use unknown rather than * for this field. Finally there is the operating system. Unfortunately you need to know more than Linux. To obtain the BEST guess of your system run `config.guess` A typical output will be `686-unknown-linuxlibc1`. Note that the C library extension (c1) has been added to the Linux operating system label. This extension is very important.

It is first necessary to configure the Makefile. Since you will be writing in a system area you will need to be logged on as root. As of the current time, March 21, 1997, version 2.7.2.2, the system does not know about x686 machines. Thus if your are attempting to build on a X686 machine you must first modify `config.sub`. This can be done as follows using vi.

```bash
vi config.sub
to initialize the vi editor
:set nu
to turn on line numbers if not already turned on
:/132
to go to line 132 which has a piece that looks like i[345]86
position cursor to character 5
a
to enter the append mode
6
add the 6. Line now looks like i[3456]86............

lines 147 needs a similar modification.
:wq
to write the file and quit the editor
```

A comment on line 422 says that 686 machines are not explicitly supported at the moment, but you have voided this restriction with the edits on lines 132 and 147. An alternative is to fool the compiler by using `i586` as the machine type rather than `i686`. Now configure the Makefile by entering

```bash
./configure --build=i686-unknown-linuxlibc1
```

This command tells the configure process to configure the Makefile for an Intel 686 machine of unknown manufacturer using the linux operating system and the libc1 library.
Running the command will invoke the i386 machine type in the config subdirectories. A successful config run will produce the following:

```
links are now set up to build a native compiler for i686-unknown-linux.
```

In performing this task config sets up a small number of one line files for the Makefile which it uses to access information in the various sub directories. It also writes out the substantial Makefile. This Makefile has many options, but none need to be explicitly specified as long as you are building a "native" compiler—that is, one that will run on the same architecture as the machine you are using for the build. The compiler is built by issuing the following command

```
make LANGUAGES=c
```

Please note that the INSTALL instructions are not uniformly rigorous about the syntax of the above and following make instructions. Rigorously the Makefile expects to see `LANGUAGES="list"` where list is any combination of c, C++ and objective-c. Multiple entries are separated by the blank character. That is the rigorous form would be `LANGUAGES="c"`. This philosophy is also true for the list that accompanies CFLAGS and other key words. Fortunately all the instructions either default correctly or are correctly interpreted as typed. There is no explicit completion message. If you don't exit cleanly then there is a problem. The last part of the output is the line

```
cxxmain.o underscore.o getopt.o getopt1.o
```

These routines were built with the resident compiler. It is now necessary to build the gcc compiler, which is bootstrapped through several stages. The command

```
make stage1
```

copies the object files to a directory stage1. The command

```
make CC="stage1/xgcc -Bstage1/" CFLAGS="-g -O2"
```

then builds the compiler using the .c files in the main directory using the new compiler. This process is quite long with much output scrolled to the screen. A successful completion is indicated by the following lines. Note that the first line is the same line from the first build.

```
cxxmain.o underscore.o getopt.o getopt1.o
xgcc: file path prefix `stage2/' never used
```

A stage 2 compiler is build by executing the commands

```
make stage2
make CC="stage2/xgcc -Bstage2/" CFLAGS="-g -O2"
```

The compilers in stage1 and stage2 should be the same. If they aren't then the build is in error. The command

```
make compare
```

performs the comparison by comparing the last 16 lines of all the files. The process uses the diff command which only reports differences. Thus there is no explicit statement that
the compilers are the same and the process is complete and error free. The most common
error is that the process cannot use or find genattr. Failure to proceed past here is a sure
indication that you have not guessed your configuration correctly and therefore run
configure correctly to make the appropriate Makefile.

The final step for gcc is installing it in /usr/bin or usr/local/bin. Compilers are
considered top level entities and hence their usual path is /usr/bin. Compilers such as C,
FORTRAN, PASCAL and other third generation languages exist at the highest level and
are considered global to the running and organization of the system. Other compilers and
systems, especially the high level systems such as MATLAB usually exist in /usr/local.
Libraries are usually installed in /usr/lib although other locations such as /usr/local/lib are
often used. To install the compilers in /usr/bin, include in the Makefile the line

    configure --prefix=/usr

Clearly any existing copy of gcc, g77 or f2c and their associated libraries will be
overwritten. To safeguard old versions it is desirable to move or copy these files to another
area or name. The INSTALL document discusses this issue in depth near the end of the
notes, after the numbered steps. To install the gcc compiler, enter

    make install CC="stage2/xgcc -Bstage2/" CFLAGS='"-g -O2" LANGUAGES="c"

Under the /g77 directory you find a sub-directory with the name of the specific
compiler (e.g. g77-0.5.20). Under this sub-directory are two documentation files—
COPYING.g77, and README.G77 and a directory /f containing the source (.h, .c, and
.info files), an INSTALL documentation file, and two directories, /gbe and /runtime. It is
essential that you read COPYING.g77, README.g77, INSTALL, and the README file under
the /gbe directory.

The README file under /gbe explains the application of the necessary patches. It also
refers to gcc source directory. In this case the gcc source directory is
/usr/local/src/gcc/. The patches are to be applied to the various sub directories and
files that exist off this tree. Thus the patches are applied from the gcc directory not the gcc-
2.7.2.2 directory. However updating the Info documentation must be done from the gcc-
2.7.2.2 level. There are only directories of the tree at the gcc level. Successful completion
of the patching will result in the message

    Patching file version.c using Plan A...
    Hunk #1 succeeded at 1.
    done

Next you should specify what compiler names are to be installed by creating zero-length
files (using the touch command) with filenames recognized as keywords by Make. The
existence of the file f77-install-ok is used to indicate that an installation using the
name f77 is required. If f77 is pre-existing in /usr/bin then the old version will be
overwritten installation phase. The file f2c-install-ok is used to indicate that the
installation process will install or overwrite this file. Since the GAMIT/GLOBK suite does
not use f77 as a default for g77, there is no requirement to install f77 or to modify f2c.

One option that must be specified to run GLOBK successfully is an override of the default
(99) maximum number of Fortran units that can be open at one time. Although you are
unlikely to need over a few hundred (corresponding to the number of h-files) there is no
harm in allowing 9999 files—set by changing MXUNIT in /f/runtime/libI77/fio.h.
This change is described in the INSTALL notes in the section dealing with Changing Settings Before Building.

The g77 compiler can be built using a bootstrap or direct method. The direct method is recommended because you have already built a native gcc environment and GNU’s FORTRAN is really an extension of this environment. The only problem that you may encounter is where the accessible modules lie. Copies certainly do lie in the stage1 and stage2 sub directories. Copies may not currently lie in the main gcc-2.7.2.2 directory. If gcc is correctly linked then there should be no problems. Build g77 by entering

```
make -k CC=gcc LANGUAGES=f77 all g77
```

The process ends with the line make: 'g77' is up to date.

It is recommended that you perform some validation of the system before you install the compiler. Several methods are presented in the INSTALL notes. We recommend running the two commands detailed in the INSTALL notes. Note that these commands are run from gcc-2.7.2.2 and not gcc since gcc is not the source directory.

Compare these outputs. We have noticed that the loader may be difficult to find in a completely new build. In all of our tests the problem is resolved at installation. Resolution is indicated by running a small test program rather than compiling null source files. If the problem is not resolved then there is an error in the building procedures.

Finally, install g77 by entering

```
make -k CC="stage2/xgcc -Bstage2/" CFLAGS="-g -O2" LANGUAGES=f77 install
```

This command must include the same definitions for CC as you used in building the last stage of the gcc compiler. A mistake here will produce the message

```
make: Target `install' not remade because of errors.
```

There are no messages associated with a successful completion. However the following checks can be made to validate the system:

1. Examine the date stamps in /usr/bin and /usr/lib. The file should exist and have a date just a few moments before. The following is the output of ls -l g?* from the /usr/bin directory.

```
-rwxr-xr-x   2 root     bin         41671  Mar 28 08:53 g7*
```

2. In your own home space compile the following FORTRAN program and then run the resulting a.out

```
program test_build
  do i=1,10
    print *, 'Hello World: ', i
  end do
  stop
end
```

Don’t forget to add the .f after the file name. We recommend that you compile with the g77 name. That is use g77 test_build.f.

We recommend that you now update your Info directories and clean up the src directory structure where you may have aborted and other safety structures.