DESY Computing Newsletter
No. 5 December 1993

A Publication of the DESY User Support Group

Contents

DESY Computing Newsletter No. 5

Editorial

by Michael Behrens

Central computing at DESY is changing rapidly, and most of us have to learn and become accustomed to new ways of doing our computing. The screen on the desktop is in many cases an X-terminal, or will be replaced by such a device in future, and physicists are getting used to working on multiple systems simultaneously. On the computing side we have corresponding changes, namely

- more Unix systems
- less IBM MVS
- significant changes for VAX/VMS

Several articles in this newsletter should help to find out

- where and how to get accounts
- how to move from the IBM to Unix
- what new services are available

At the end of this newsletter you will also find an index of all available USG publications. They are available in the self–service area of the computer center or via www/mosaic (*see DCN4*).

CERN documentation is being provided by the User Consulting Office, UCO. This service is heavily used, and we occasionally ran out of stock in the past. Several people then developed the habit of pulling these documents from the CERN server over the network and printing them on the DESY printers. This is seen as bad practice, as CERN documentation is often quite lengthy. Pulling multiple copies over the network is bad usage of expensive bandwidth, and printing them individually blocks printers for a considerable time, as some documents are quite demanding on printer resources. The latest CERN Newsletter e.g. had blocked various printers on the site for more than 30 minutes and then even caused them to crash.

The UCO has already reacted. It will in future pull the documents once, print them once locally and produce the required copies on a photocopier. This should help to keep our resources available for those tasks that really need them.

This Newsletter, as well as previous issues and all USG Documentation, is also available online. You may browse through it using a WWW browser (see the article in this Newsletter) or get them via anonymous ftp from ftp.desy.de in PostScript format.

Editors of the DCN are:		
Katherine Wipf	wipf@desy.de	ext. 3222
Michael Behrens	M.Behrens@desy.de	ext. 2556
Jan Hendrik Peters	petersjh@desy.de	ext. 2583

New Accounts on Central Computers

by Michael Behrens

In the User Consulting Office we often get asked:

"I want to use the DESY computers. What accounts do I need ?"

The general answer for new users is:

The standard account should be on the x4u, our Unix system from Silicon Graphics (SGI), or on the central VAX cluster if you want to work on VMS.

Both systems will offer available cpu–cycles for the next years and at least the **x4u** can take more users. More detailed information, and information on other systems available, will be given later in this article.

How to get an account

The User Consulting Office – in short UCO – handles all requests for accounts on central systems. You can get application forms at the UCO. You fill in the form, sign it, and get the administrator's section filled out and signed by your group administrator. UCO will help you to identify your administrator if you do not know who s/he is. You return the completed form to the UCO for installation of the new accounts. New forms can be picked up from, and completed forms returned to, the appropriate box provided outside the UCO. Allow a few days for registration to be completed. You will not get notified upon completion, you simply try if your accounts are already available.
x x x A warning: Fill out the form carefully and follow the instructions on the form. This will make user registration proceed quickly and smoothly.

Central Unix systems at DESY

The central Unix systems at DESY are available for interactive usage for all people working at DESY and for batch usage by those users who do not have dedicated machines for their groups. Monte Carlo production is exspected not to run on central machines at DESY. The large experiments H1, ZEUS, and HERMES have dedicated machines for reconstruction and data analysis. This type of work must not be done on the public systems. Currently several different Unix platforms are available at DESY:

The Silicon Graphics system named **x4u** is our third generation of Unix systems. It consists of 2 systems with 8 processors each. This makes more processing power available to the general community than we had ever before. The machines have fast cpus as well as good I/O performance. They are well suited for interactive and data–intensive HEP applications.

The Hewlett Packard 9000-700 cluster was the second generation of public Unix systems. They are still healthy and in good shape, and they offer especially good floating point performance. Therefore users from machine physics and from theory use HP systems quite heavily. Currently the HP systems are heavily loaded. Only users with low I/O and high cpu demands should work on HP's for the time being.

The Apollo systems were the first public Unix systems at DESY. By today's standards they are pretty slow, and the best part of their lifetime has passed already. You should not try to get an account there unless you have special reasons.

The central VAX cluster

We have a large user base from many different groups on the central VAX cluster, and this community is still growing relatively fast. Besides this some groups are heavily VAX – based, e.g. the synchrotron radiation people. They will probably want to stay in a VMS environment as long as possible. It is also clear that with the move to Alpha stations currently planned (*see the article in this newsletter*) there are no reasons for moving from the VAX to other systems. The VAXes are intended to be used as interactive machines and for program development, but currently there is little room for heavy computations. CPU intensive tasks should be run on the Unix systems.

The DESY IBM

In the past the IBM had been the traditional platform for most DESY users. Currently DESY is planning a reduction of the IBM, and many tasks now performed on the IBM will have to move to other platforms, mainly to the Unix systems. If you nevertheless want to get an account on the IBM, you may find that you may have to move off the machine again quite soon. If you are going to start new work and are free to choose, it would be better to apply for an account on the other systems.

Terminal access

The current default terminal at DESY is a small b/w X–terminal equipped with a *chooser* program that connects you to any DESY host. Larger grey-scale or color X-terminals are available in some offices or terminal pools. If you need a terminal and have none available, you should ask via your group for an X–terminal.

whenever you receive an X–terminal, the old terminal(s) must be returned to the computer center. There are also a large number of Falco terminals on the site. These are vt–compatible terminals with limited graphics capabilities. They also allow you to connect to all DESY systems. DESY will not acquire new Falco terminals in future.
The IBM is accessible from 3270 terminals too, but new 3270 terminals will not be deployed, and they offer access to the IBM only.

Recommendations

- Ask for an account on the public Silicon Graphics cluster named **x4u**, if you have no specific arguments for another system, or if you are looking for a system with free capacity.
- Ask for a Vax account is you are used to Vaxes and VMS, if your group is Vax based, or if you simply like this environment. Do not select the Vax for mass data processing.
- Ask for an HP account if you have an HP machine available for your group.
- Ask for an account on your experiments machine (H1, Zeus, and soon HERMES) for work for your experiment. Your group administrators will know best if you qualify for this.
- Ask for an IBM account only if you need it for future work and if your collaborators ask you to do so.

How to Transfer Applications from IBM to Unix

by Michael Behrens

Transferring applications from the IBM to Unix should not be too hard in most cases, as HEP programs are usually written to run on multiple platforms, and most physicists are used to avoid the more common traps people tend to fall into. There are a few traps left however, and this article should help to avoid them.

Getting rid of line numbers

Many people still have line numbers in their sources on the IBM, as this is the default when NEWLIB creates a new member. These line numbers have to be removed by the command *renum 0*.

If you want to remove line numbers from ALL members of a library, the following sequence should do it:

```
change A Z9
    TO VERIFY CHANGE FROM A TO Z9 ENTER YES, ELSE NO
yes
    YOU ARE NOW ENTERING COMMANDS IN STACK
    CHANGE GOES INTO ACTION IF YOU SIMPLY PRESS ENTER
renum 0
    ENTER NEXT STACK COMMAND:
```

If you simply press ENTER, renumbering will start quietly and line numbers for all members with names in the range from *A* to *Z9* will be removed. This means precisely that the contents of columns 73–80 will be erased. You will get warnings when there was some "real" contents in these columns before, but it will be erased nevertheless. So you should be careful if you have already unnumbered members, or if columns 73 to 80 are used in some members.

Transfer and Conversion

Sequential files and some libraries on the IBM are in "standard" formats and can be transferred easily by ftp in ascii mode. NEWLIB libraries are usually in a special internal compressed format (the COSY format) that is readable only on the DESY IBM under NEWLIB. They are characterized by
Recfm V Blksize 6233 Lrecl 6229
in the NEWLIB LDS screen. The import of COSY files must then be done using FTP in **binary** mode. When you are logged in on your Unix system, you start the ftp connection with
`ftp ibm`
You then enter userid and password and continue:

`binary`	*select binary mode*
`case`	*new file names in lower case*
`cd` *<your IBM library>*	
`get` *<member>*	*for single member*
...	

or for a full library

`prompt`	*suppress prompts*
`mget *`	*get all members of library*
...	
`quit`	*terminate ftp*

If you don't specify **binary**, the binary compress information is destroyed by the EBCDIC to ASCII conversion. The latter is done automatically during the decompression.

`uc` *<input–file>* *<output–file>*

converts COSY infile to a card image outfile without the need of converting it to FB format on the IBM. **uc** has been installed on the x4u machine and the HP cluster.
Note: See news uncosy on x4u and hp or Help uncosy under NEWLIB for details.

FORTRAN changes

Once you have your files on the Unix system, you have to make some more changes in order to compile properly on Unix.

Depending on how you had produced the source on the IBM, you may have to change names of subroutines, functions, and common blocks in your sources.

Besides upper/lower case, the following items have to be considered:

- Usage of IBM specific libraries
  If you use libraries that are available only on the IBM, you will have to replace the calls by (probably different) calls to a library that is available on Unix. For example you will have to replace calls to DESYLIB routines by calls to CERNLIB routines, if equivalent ones are available. The same holds for functions specific to the IBM, e.g. full-screen output. No general advice is possible, as this depends completely on the application.

- DD cards and OPEN statements
  On the IBM DD–statements are usually used to connect a file to a FORTRAN unit and to specify file parameters. This is not possible under Unix. For every file you need an OPEN statement in your program with all the necessary details and the explicit filename.

- NEWLIB %MACRO statements
  %MACRO statements have to be replaced by

FORTRAN INCLUDE statements of the form

```
INCLUDE "filename"   for SGI and HP, or
INCLUDE 'filename'   for HP only.
```

On SGI, the search order for INCLUDEs is:

- /usr/include
- current directory

on HP, the search order is:

- directory of source
- directory as specified by –I option
- current directory
- /usr/include

Note that the details of the file specifier format and the search strategy are system dependent. Please check the documentation if you experience problems.

If you relied on IBM specific extensions to FORTRAN, you may in some cases have to make some more changes. No general advice can be given here. The compiler will complain in these cases, so this should not be a major problem.

Running Jobs

When you have moved your code to Unix, you will eventually want to produce an executable program and run a job to execute this program. On the IBM, you either used NEWLIB with Compile, Link and Go commands, or you submitted a job executing VFORTCG or an equivalent procedure. An example job could have looked as follows:

```
//       JOB  CLASS=A,TIME=(,20)
//*MAIN RELPRI=MED
//*
//MYJOB EXEC VFORTCG,
//         LLB1='F99ABC.MY.PERSONAL.LIB',
//         LLB2='R01UTL.CERN.PRO.PACKLIB'
//*
//C.SYSIN DD *
      PROGRAM MAIN
      ......
      READ(5)  data,from,std,input
      ......
      READ(10)  DATAIN
      ......
      WRITE(11) DATAOUT
      CALL  SUB1(I,X)
      STOP
      END
//*
//G.FT10F001  DD DSN=F99ABC.FORTIN,DISP=SHR
//G.FT11F001  DD DSN=F99ABC.FORTOUT,
//               DISP=(,CATLG),DCB=R01DCB.VBS,
//               UNIT=FAST,SPACE=(TRK,(5,5))
//G.FT05F001  DD DSN=F99ABC.INPUT,DISP=SHR
```

The FORTRAN code on Unix should then be

```
      PROGRAM MAIN
      ......
C
      OPEN(UNIT=10,FILE='..../data/input')
      OPEN(UNIT=11,FILE='..../data/output')
C
      READ(5)  data,from,std,input
      ......

      READ(10)  DATAIN
      ......
      WRITE(11) DATAOUT
      CALL  SUB1(I,X)
      STOP
      END
```

Note the extra OPEN statements specifying explicitly the files used. It is compiled and linked with the command

```
f77 example.f -o example
   -L .               -lpersonal
   -L cern/pro/lib    -lpacklib
```

if personal.a is your personal object module library,and is executed by the command

```
example < input/file/ft05f001
```

The New Central SGI Machine x4u2

by Jan Hendrik Peters

At the beginning of November the SGI machine x4u (Power Series) has been replaced by two machines, x4u1 and x4u2, of the Challenge Series. First the x4u2 went into production. Later the x4u1 will be added to form a cluster. The new machines are each equipped with 8 R4400 processors instead of the 6 old R3000 processors on the x4u. Thus the public central unix cluster will provide an enormous cpu power to the users. This new x4u cluster is intended for all users who have large cpu and i/o demands in their jobs and for all those who need access to the large variety of information services provided on this machine. Users of the H1, ZEUS and HERMES collaborations, however, should use their own Challenge machines (dice, zarah and hermes, resp.) for production and analysis jobs and use the public machine only for those software packages not available on their machines. For the future we aim at having a single home directory for every user on all Unix systems at DESY. Further steps in this direction will be announced in due time.

The transition from the old to the new technology was done in two steps. First, system and library files as well as the home directories of the users were moved from the x4u to the x4u2. Then, in a second step, the new machine x4u2 was made available for login. When the transition is finally completed, the old x4u will no longer be available for login.

In this article we will describe the steps taken by the system administrators in the transition and what effect this will have on the users. Then we will give some advice on how to fully utilize this new machine.

The Relocation of Home Directories

The home directories of users on the x4u were moved to the x4u2 step-by-step with the following constraints:

- All home directories formerly residing on the **x4u** were automatically copied to the x4u2 by the system managers. The old directories are now no longer accessible when you login to the x4u. Instead, the new directories are mounted on the x4u over the network from the x4u2. You should see no difference working with this new scheme, neither on the x4u nor on the new machine x4u2.
- All home directories which reside on the **Apollo Cluster** (i.e. all those which show /home/apollo/user1/*userid* or /home/local/user1/*userid* or something similar when you type *pwd*) will no longer be accessible from the x4u2. All users with such an old home directory will get an empty new home directory on the x4u2. You will have to copy all your files from the Apollo to the x4u2 to use them. (See below how to copy all your files to the new directory structure.) If you need help, ask Thomas Finnern, R2.
- Users who already had a home directory on the x4u2 (System and Oracle test users) have to copy all their files manually from the x4u. (See below how to copy all your files to the new directory structure.)
- Users of the H1 and HERMES collaboration will get a home directory on their machines dice2 and hermes. These home directories will then be mounted on the x4u2.

Once all home directories from the old SGI machines have been moved or copied to the x4u2, the old disks will be emptied and then transfered to the new machines directly after Christmas.

Moving the Logins to the x4u2

In a second step the x4u2 was opened for public login and for servicing the X-terminals. The chooser menu now contains the new destination x4u2 in the xdmcp and telnet section (see USG/93/05 for how to use the chooser). X-terminals which were directly connected to the x4u are now being moved to the x4u2. Once all home directories and all X-terminals have been moved, the old x4u will be taken out of service and the machine will no longer be available for users.

Copy files to the new machines

If you want to work with the new x4u2 but have no home directory or cannot login, although you did have files on the x4u or were able to login there, then one of the following things might have happened:

- The system manager forgot to copy your files from the x4u to the x4u2. Action: send an email to the system manager (r02nug@x4u.desy.de) and ask him to do this for you.
- You had all your files cross mounted from the Apollo. Action: Copy all the files you need to the new platform as shown below.
- You already had a home directory on the x4u2 which intentionally was not overwritten by the automatic transfer. Action: Copy all the files you need to the new platform as shown below.
- You are a new user on the x4u2 or have been a new user on the x4u and never logged in. Action: Copy files from other machines to this new machine as shown below.
- If you are an H1 user with an old home directory on rec06, you will get a home directory on the machine dice2 which will be cross mounted to the x4u2. Action: You **don't** have to copy any files to the x4u2. In case of problems contact the x4u system manager (r02nug@x4u.desy.de).

There are several mechanisms to copy files to the new platform. If you need to transfer files from another Unix platform use the commands **rcp** (remote copy), **rdist** (remote file distribution, works between SGI machines and from Apollo), or **cp** (copy, on the Apollo), and from other platforms use **ftp** to transfer the files. If you feel uneasy using any of these utilities, come to the UCO or send an email to Thomas Finnern to help you copy files from the Apollo to the x4u2.

- **rdist** on any SGI or Apollo machine will copy all files from your home directory or from any other directory to the new home directory on the x4u2 with the following command:
  rdist -c *name* **x4u2:***newname*
  This will copy the file/directory *name* to the x4u2 into a (new) directory of name *newname*.
  rdist -c *path-name-of-your-home-dir* **x4u2:**
  This will copy the complete home directory to the x4u2.
  The command rdist will overwrite existing files of the same name on the destination host when the modification dates of source and target files are different. If a file to be copied does not exist on the destination host, it will be added. To avoid overwriting of younger files use the option -y: **rdist -y -c ...**

  You should be aware that **rdist** *works fine between Apollo and SGI, but this command is version and manufacturer dependant and should be used with care in other environments.*

- **rcp** will copy single files to the new host with the following command:
  rcp *old-file* **x4u2:***new-directory*
  or if you are on the x4u2 in the destination directory:
  rcp *old-host:old-file new-file*
  rcp will always overwrite existing files of the same name.

- **ftp** allows you to copy a single or multiple files from another machine. Remember that binary files have to be copied in the *binary* or *image* mode.

- With **cp** on the Apollo you can copy files or directories from any central to any other central Unix machine. All home directories are mounted on the Apollo in the file tree /home/*cluster-name*. Use the following options for the copy command: **cp –Ppvrsi**.

What is Different on the new Machines?

The new machines are binary compatible to the old SGI machines. That means, that all your jobs should run on the new platform but will not show the expected gain in speed. To fully utilize the new processor technology you have to recompile all your sources on the new machine.

The library format, however, is different on the new machines. Therefore you cannot link a newly compiled program with an old library. You will have to recreate all your personal libraries on the new machine. When you are using the make command (see USG/93/11) you have to *touch* all your source files, or remove your object files before you issue your usual *make* command. Besides this difference in the library structure, you should not experience any other problem.

Whenever you observe that files or utilities are missing on the x4u2, please contact the UCO (uco@desy.de). We will try to have the missing items reinstalled as soon as possible.

New Tapes in Tape Robot

by Michael Behrens

At the end of October the StorageTek (STK) tape robot was upgraded. The tape drives have been replaced by new drives that write 36 tracks, instead of the former 18 tracks, on the tapes, thus doubling the capacity of the existing tapes. In addition, longer cartridges have been made available, increasing the capacity even further. Together with the (hardware) compaction feature introduced earlier this year, each normal tape will hold a total of about 1 GB (1000 MB), each long tape about 1.6GB (1600MB).

Note: The capacities mentioned assume an average compaction factor of 2, consistent with measurements of the computing center.

All this is transparent to the user. Nobody has to change his/her JCL or programs in order to cope with the upgrade.

The new drives can use the existing cartridges in the STK robots, and the system knows if it is old or new data. It will read all tapes, old and new ones, and it will write all tapes using 36 tracks and the compaction feature. The catalogue listing on the IBM shows if the tape was written on the old or the new drives.

Most datasets on tape are much smaller than the full capacity of a tape. In the past the computer center has therefore regularly run a job to move many small datasets onto a single tape in order to utilize the full capacity of the tapes. This compression job is being continued as before.

Most groups have internal rules on preferred or maximal size of their datasets. These rules are the sole responsibility of the groups and will probably stay the same.

The manual tape drives are not affected by this upgrade.

The long cartridges mentioned above will not be usable directly by you but only by the compression job run by the computing center. Thus your files will eventually migrate from one of the normal cartridges to a long one in a transparent fashion.

The Future of VMS at DESY – Introducing the New Alpha Line

by The VXDESY Systems Group: Volker Heynen, Wolfgang Krechlok, Till Poser

Last year Digital announced a new line of computers that are capable of running **VMS**, amongst other operating systems, and are based on RISC architecture CPUs. This line is called the ALPHA line and is many times more powerful and much cheaper in initial price and maintenance cost than the VAX CISC Hardware. The ALPHAs are data- and source-compatible, but not binary compatible to the old VAX architecture. The ALPHA architecture, for instance, is purely 64 bit, compared with VAX 32 bits.

The low cost, high performance, and modern architecture of the new line of processors has led the VXDESY Systems Group to draft plans for moving to this inexpensive, powerful hardware platform while still sustaining and maintaining the high level of reliability, stability, and comfort that our users are used to under the **VMS** Operating System.

The existing central DESY VAX Cluster, with its 14 CERN units worth of computing power and 35 GByte of diskspace, currently serves 1450 active users with an average constant load of > 180 users and > 250 user processes during normal office hours.

Based on the profile of the existing operations, the following configuration has been proposed to extend and gradually replace the existing hardware:

- The DESY **AXP** Cluster will consist of 4 CPU servers, **AXP** 3000/500S, running **OpenVMS AXP**. Each server will be equipped with 256 MByte of memory and has 20 CERN units of CPU power.

- The servers will be connected via FDDI to a GIGASWITCH network switching device with a throughput capacity of 3.6 G bit/second. The GIGASWITCH will essentially serve as an external computer backplane.

- Each server will be equipped with 3 GByte of internal disks for system, library, paging and swapping on the internal SCSI-2 bus. The user- and scratch disks, 59 GByte in all, will initially be attached to the external SCSI-2 buses connected to the turbochannel It is foreseen that during the next year the disks will be moved to a DEC StorageWorks Device Array, connected via FDDI to the GIGASWITCH.

The new **AXP** cluster will be connected to the CISC hardware over the network. It is planned to run a so called mixed architecture cluster, where both ALPHAs and VAXes will coexist transparently.

The initial configuration as outlined above has already been approved by the DESY management and placed on order. It will be delivered at the end of this year. The VXDESY Systems Group foresees a span of several months, during which experience with the systems, the configurations, and the operations will have to be gained. During this time a limited number of users will have access to the new machines. The overall time scale of **AXP** introduction is expected to follow the outline below:

End of '93: Installation of the new hardware.

1st quarter '94: Test operations and limited access for users for test purposes.

2nd-4th quarter '94: Normal operations and migration of users.

End of '94: Begin of gradual phase-out of the current VAX hardware.

A direct connection of the **AXP** cluster to the StorageTEK cartridge silos currently available only on the IBM is being considered. This would allow the user to access the data stored in those silos without detours over other systems.

As has been mentioned before, the new computers are not binary compatible to the old hardware. For the users this means that they will have to recompile and relink all their programs. Normally this does not pose any problems for standard programs. For executables the possibility exists to "translate"

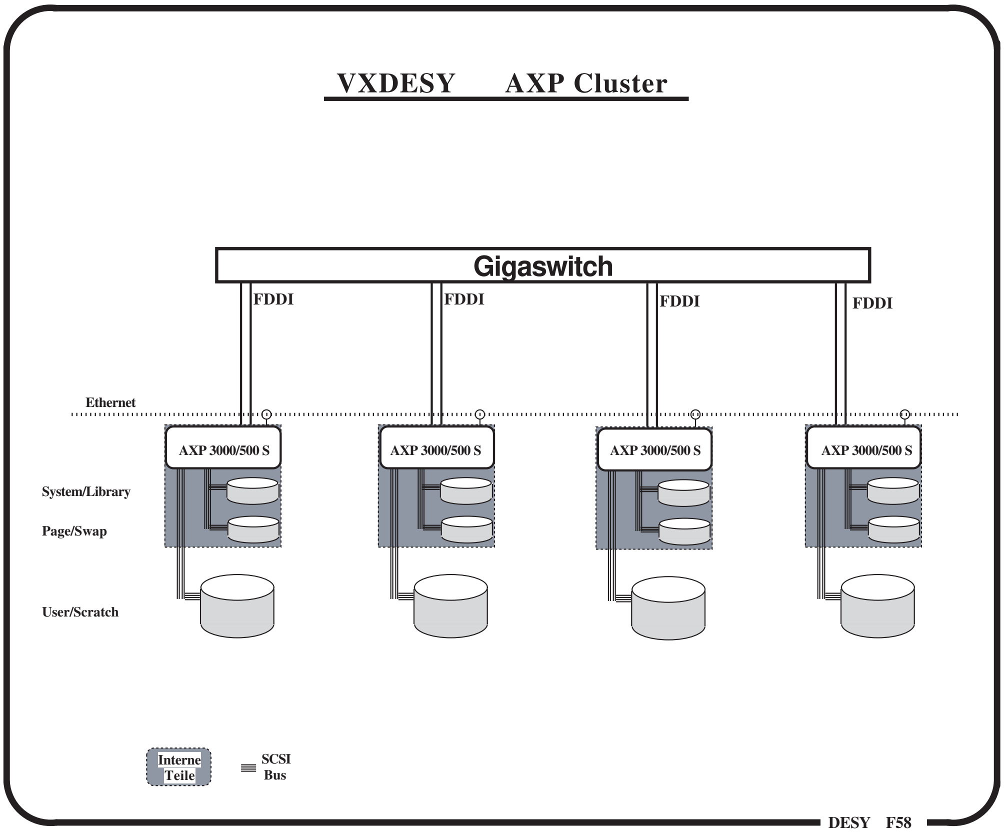


Figure 1: Basic Configuration of an **AXP** Cluster for VXDESY

the VAX image into an **AXP** image. This works for most of the programs, though the translation naturally will carry a penalty in loss of efficiency of the translated program and in file size. Translated programs run only at $\sim 40\%$ of the speed of the same programs compiled native. The users will be provided with a modicum of help and assistance in porting their programs during the migration phase. For applications where the migration is not possible, a small VAX system will still be available in the cluster after the end of '94.

The outlined configuration has the advantage that it allows easy scaling of all important resources to meet any further user demand. It will also allow the easy introduction of future generations of ALPHA processors.

It is envisaged that at the end of '94, the VAX users will have moved, preferably painlessly and transparently, to the new architecture, where they will enjoy much more computing power with the same reliability and level of comfort they have come to expect and demand from a **VAX-VMS** system.

If you have any comments, inquiries, and suggestions, send mail to VXDESY::SYSTEM or system@vxdesy.desy.de. It will be forwarded to all members of the VMS system group.

E-Mail and Privacy

by Michael Behrens

Thousands of mail messages are created or received each day at DESY. Most of these mail messages relate to our user's professional work, but many people send and receive personal mail as well, least occasionally.

I will not touch the issue of whether this is a legitimate use or misuse, but there is another caveat: E-mail is not as private as some of you seem to believe. Most people rely on the privacy of classical mail,and they believe that the privacy of e-mail is more or less at the same level, but this is not true.

If a mail message is transferred successfully to the intended recipient at the other end, then in most cases suitable privacy is provided. But if something goes wrong - and most people have some experience with failed mail - there will probably be other people having access to your mail.

If, for example, an address is mistyped, or if the receiver is not able to receive mail because of space problems, your mail will most likely be forwarded to the "postmaster" on one of the systems involved. The same will happen if a network outage occurred or if an intermediate system is misconfigured (and there are many misconfigured systems worldwide).

There are good reasons for forwarding undeliverable mail to postmasters, as they should be made aware of delivery problems and might be able to assist their users e.g. by forwarding undeliverable mail to the (hopefully) correct address.

Postmasters are certainly trustworthy people in general, but you might not want them to read your personal mail nevertheless. They might also prefer not to stumble across other people's private communications. While acting as postmaster I have seen very personal mail on a few occasions already, and I do not feel very comfortable with this situation.

I will not cover the more malignant ways of compromising privacy that also exist. The general message should be clear by now anyway:

Do not trust to much in e-mail privacy.
If you would not send a message on an open postcard, you had better not use e-mail but an ordinary letter instead, and transmit it via the traditional postal services.

WWW Information Services

by Makoto Ikeda

The World Wide Web (WWW) is an information discovery and retrieval system developed at CERN. The original aim of the WWW project was to help promote the availability of information within the high energy physics community. Currently the role of WWW has expanded by reaching throughout the whole Internet. Information sources are found in many different fields including biology, meteorology, literature, and law. The advantage of using the Web lies mainly in the large number of platforms supported and the ability to access a diverse range of information sources. The DESY User Support Group now supports a number of clients and servers available on all Unix and VAX central computers.

An Overview of the Web

The World Wide Web project was first proposed in 1989 by Tim Berners-Lee (CERN). The system has a client-server architecture; it requires a client or a "browser" on the user-end and a server for the information source. TCP/IP protocol suites provide the communications necessary to transfer various types of files.

Originally developed on a NeXT computer, the first line-mode browsers appeared in March 1991 running on various Unix platforms. These were terminal-based, but X11-based browsers soon followed thereafter. Although development of the main common library and the server remained at CERN, many institutes in Europe and the US participated in the development of clients.

In early 1993 an X11/Motif client, Mosaic for X, was developed by the National Center for Supercomputer Applications (NCSA). The beta versions were released to the public and it was rapidly accepted by the internet community. New aspects such as the support of in-line graphics were introduced. Shortly thereafter CERN released the common library, line-mode browser, and server into the public domain. This accelerated work on the server end and new general purpose servers were soon developed that also increased the functionality of WWW.

There are now currently over 100 servers operating in universities and national laboratories throughout the world, covering subjects in the sciences, arts, and engineering. The WWW project has now gone beyond the original scope of providing an information system solely for high energy physics.

Some Technical Details

The profusion of acronyms used in computing is only too familiar to most users. The Web, unfortunately, adds three more to the growing list, but they are necessary in understanding how the Web works.

1. HTML - Hypertext Markup Language

When you start a particular browser, you should see some formatted text on your screen and perhaps some special text that is highlighted in a particular way (perhaps underlined or in a different color). The special text can be accessed either by the mouse or a particular key,and this results in a new document or image appearing on your screen. The browser is reading a *hypertext* file and the highlighted text are *links* to other files. In order for the browser to understand how to format the file and display the links properly, a special language, HTML, was created for this purpose. It is called a markup language because the main text is interspersed or "marked up" with instructions that tell the remote client to format the text in some way. HTML is compliant with the Standard Generalized Markup Language (SGML), which is the ISO standard meta-language of markup languages. This allows one to define the function and attributes of each markup tag and any special entities (like national character sets). In addition, adhering to the SGML standard allows HTML files to be translated fairly easily to and from other formatting languages.

2. HTTP - Hypertext Transfer Protocol

Files that are located on a remote machine are made available by a server running a WWW-native protocol called HTTP. This protocol not only transfers files from the server to the client, but it also allows for the client to send keywords to a server. The keyword can then be used by the server to perform search queries.

Although HTTP provides additional flexibility, the design of WWW does not exclude other information sources which are available through other protocols. On the contrary, the common WWW library provides for the ability to transfer data from other servers like FTP, usenet news, and other information systems like gopher and WAIS. The data is transferred and the proper HTML markup is automatically included.

3. URL - Uniform Resource Locator

The URL gives the location of the server and the protocol to be used to access the document. Sample URL's are given below:

- http://info.desy.de/HEP/HEP.html
- ftp://ftp.desy.de/pub
- news:desy.usg
- telnet://x4u2.desy.de
- gopher://gopher.micro.umn.edu/

One can see that the protocol is given on the left side of the colon, while the location is given on the right. The URL is embedded as an attribute in a HTML markup tag so that the client can be instructed to go to that new location.

Browsers available on DESY computers

The preferred browser to use on DESY central computers is NCSA Mosaic for X (Mosaic) if you have access to an X-terminal or a workstation. The most notable advantage, of course, is the point-and-click interface that X11/Motif provides over terminal-based clients. However what sets Mosaic apart from other X11-based clients is the graphics and sound support. Initially all WWW clients were only able to recognize hypertext and plain text files. When Mosaic was introduced, it included recognition of other types of files for images, sound, and movies. These files were transferred and displayed through the use of various Unix public domain tools. Thus Mosaic was the first client to support *hypermedia*.

The NCSA officially supports Mosaic on SGI-Irix, IBM-Aix, SunOS, DEC-Ultrix, and DEC Alpha-OSF/1. Therefore the only supported version running on the central computers is the SGI version. However, with the use of private (but unsupported) ports, Mosaic is also available on the HP cluster, Apollo, and VXDESY. Attempts will be made to keep the unsupported system binaries as up-to-date as possible, but there is no guarantee that the executables for these systems will be completely bug free. There may also be delays on the availability of the latest versions on these systems. Currently the SGI and VXDESY have Mosaic 2.0, while the HP and Apollos have Xmosaic 1.1. The name was changed by the NCSA in the transition to version 2.0 because of major changes, including the use of the second generation common WWW library.
To start Mosaic on UNIX systems, type

prompt> `mosaic &`	*for version 2.0*
prompt> `xmosaic &`	*for version 1.1*

Due to the large number of monochrome X-terminals at DESY, Mosaic has been compiled on the central machines to appear in monochrome mode as default. If you have a color monitor, then add the -color flag, e.g.,

prompt> `mosaic -color &`

On VXDESY a symbol must be set first which can be placed in your login file

`MOSAIC:==$DESY$ROOT:[WWW]MOSAIC`

and then the command can be spawned or executed.

VXDSYA$ `MOSAIC`
or
VXDSYA$ `SPAWN/NOWAIT MOSAIC`

To obtain the color X resources, copy the file DESY$ROOT:[WWW]MOSAIC.DAT to your top login directory and restart Mosaic.

Mosaic features in-line graphics (i.e. the ability to place images within the view of a document), complete record of history, hotlists, and annotations. For more information and a short review of the browser's features, please consult the USG reference "Introduction to *NCSA Mosaic* "(USG/93/6).

For regular terminals, the line-mode browser is available on Unix and *lynx* is available on VXDESY. Both of these browsers can be accessed by the command

prompt> `www`

The lynx browser was chosen as the standard terminal-browser for VXDESY because support for the VMS version of the line-mode browser seems to have been dropped by CERN. So far lynx has had excellent support. Navigation is made simple by the use of arrow keys and all links are displayed in reverse video. Lynx is also available for IBM-Aix, DEC-Ultrix, and Sun.

WWW Services from DESY

On all central computers, the default home page has been set to the main WWW server on info.desy.de (currently X4U2). If you would prefer another home page, set the WWW_HOME variable (on Unix systems) or logical (on VMS) to the URL of your choice. The DESY home page provides links to User Support Group material, to the experiment specific servers of Hermes, ZEUS and H1, to software support from R2 (SALAD), the DESY FTP server, and a large collection of links and documents on GNU, C++, and other subjects that are maintained courtesy of Marcus Speh. A high energy physics page provides links to servers running at various HEP laboratories including CERN, SLAC, FNAL, and SSC. CERN also provides online documentation on GEANT, HBOOK, etc. Another page is available on miscellaneous topics.

The main server is supplemented by a second server, www2.desy.de, that runs on a VAXstation. Here documents obtainable in the VMS world are made available to the public. The server also has been modified to enable search queries. Queries can be made of the DESY phonebook, datebook, and stock catalog; of user registration information through Oracle; and of the DESY VMS help facility.

Other services in WWW and beyond

To access the rest of the world, one would like an index of all the possible servers and their contents. However, this is practically impossible as new servers appear all the time, while some servers become obsolete. There is an attempt to keep a master list on the server running at CERN (info.cern.ch).

As stated previously, the common WWW library includes code to access sources through other protocols. Thus WWW clients can also act as clients to gopher, another information retrieval system developed at the University of Minnesota. It provides simple menus to access data or other menus. Search queries are also possible. The number of gopher servers approaches 1000 running mainly in universities and academic laboratories.

Another information retrieval system is the Wide Area Information Server (WAIS) developed at Thinking Machine Corporation. This system relies mainly on search queries rather than menus. A number of WAIS-WWW gateways are available to search through the WAIS world.

Making your own information available

To make your own information available you can install a server on your own machine, prepare the links and documents, and then announce yourself! If you prefer this solution, recommended servers to install include

- The CERN server from info.cern.ch
- The NCSA server from ftp.ncsa.uiuc.edu
- The Plexus server from austin.bsdi.com

The CERN server is easy to install and configure to serve files that are already prepared. Binaries are available for many Unix platforms and for VMS. The NCSA server is also quite advanced and has additional functionality due to compliance to HTML+ (see the next section). All three servers now allow shell scripts (or perl scripts in the case of Plexus) that can be used to create search engines of for other functions that are not simply available as files. Note that the Plexus server requires that you have perl installed.

You can also use the info.desy.de server to make your information public. Your files are placed on disk on the X4U2, access rights are given, and links are provided from the DESY pages. Shell scripts are also welcome.

It you have a large source of information in a particular formatting language, it is possible to make use of translator software available in the public domain that convert a particular format to HTML. There are translators for LaTeX, GNU texinfo, and VAX Document, for example.

A final note is that although most of the latest software assume a Unix platform, there are WWW servers and clients available for the PC, Macintosh, VMS, and VM. In particular the NCSA has produced PC and Mac versions of the Mosaic browser (currently at version 1.0).

Future plans for the Web

The World Wide Web is continually evolving. New developments appear almost monthly. The second generation WWW common library recently released can handle different formats as MIME types. The advances that were made by Mosaic (recognition of different media file types) have now been incorporated into the common library. The latest advance in WWW is access authorization. It is now possible to restrict access to a server to only certain internet addresses or users. This has just been released and is not yet widely implemented, but this new security feature will probably be used soon.

There are plans to update HTML to HTML+ which will increase functionality by providing new tags to accommodate the use of fill-out forms for complex search queries or to order or log items and many other new features. The current version of Mosaic and the NCSA server have fill-out form support and new servers are rapidly appearing that incorporate these features.

To keep up with the latest developments, there is a news group, comp.infosystems.www, for general users and a mailing list for programmers, www-talk (listserv@info.cern.ch). The latter, however, is mainly for serious developers.

Status of DESY User's GNU Project

by Marcus Speh, II. Theorie

As announced in the *DESY Computing Newsletter No.4*, some users of the DESY Unix clusters have decided to improve the support of "GNU" products from the "Free Software Foundation".

Since April 1993 well above 80 GNU products, among them the popular `emacs` editor, the `gcc` compiler for `C` and `C++`, GNU `find`, `gnuplot`, and various GNU utilities, are regularly updated and documented as HyperText using the re-activated World Wide Web (WWW) server at DESY. A complete overview of the supported GNU products is given in the table on the next page. The bulk of installations were done by L. Lönnblad (now CERN) for HP, M. Speh (Universität Hamburg) and W. Wander (HERMES) for SGI, D. Michelsen (Universität Hamburg) for the Apollo, A. Köhler (DESY-IfH Zeuthen) for Convex, L. Sprute for VAX, and J. Kurzhöfer (H1) for AIX. Paging and editing tools like Emacs and `less` are centrally maintained by R2 members. An own service group, `gnustaff`, has been created.

In addition to the WWW pages, installation notes which allow for simple re-installation of software have been created specific to various machine architectures. Since most of the GNU software comes with simple, self-configuring scripts, maintaining the present level of GNU software support requires only a moderate knowledge of Unix. Professional advice can be given by regular members of the Computer Center if needed.

The project can only survive if members of the DESY user community volunteer to do the little work needed to update the present installation. So far, the project has proven wrong all people who think one can only realize useful things in computing with a committee and a huge staff of people wasting lots of time trying to coordinate their work and get things straight with the administration.

If you are interested in sharing your preferred GNU tools with others, please contribute. Send e-mail to `marcus@x4u.desy.de` to be included in the `gnustaff` group.

Table: Current version numbers for the most important software of the GNU project (ca. 40 left out), as of September 11, 1993.

GNU Product	HP	SGI	Convex	VAX	Apollo	AIX
archie	3.0	3.0				1.3
atavachron	1.1	1.1				
auc-tex	7.3	6.1				7.3
autoconf	1.5	1.4			1.3	
binutils	2.2.1	2.2.1			2.1	2.2.1
bison	1.21	1.21		1.20	1.19	1.2
cpio	2.2	2.2				
cvs	1.3	1.3				
diffutils	2.3	2.4			2.2	
emacs	19.19	19.19	19.12	18.59	18.57	19.19
f2c	4.28	4.28				
fileutils	3.6	3.6	3.4		3.5	3.6
find	3.8	3.8	3.7			3.7
flex	2.3.8	2.3.8				2.3.8
ftnchek	2.5	2.5				
gas	1.36				1.38	
gawk	2.15.2	2.14		2.14		2.15.2
gcc	2.4.5	2.4.5	2.3.3	2.2.2	2.3.3	2.4.5
gdb	4.10	4.10				4.10
ghostscript	2.5.2	2.5.2	2.5.2			2.61
ghostview	1.5	1.4.1	1.4.1	1.4.1		1.5
gnuplot	3.4	3.2		3.2		
grep	2.0	1.6				
groff	1.08	1.08				
gzip	1.2.4	1.2.1	1.2.3	1.05	1.0.5	1.2.4
indent	1.8	1.8				
ispell	4.0	4.0				4.0
lemacs	19.4	19.4			19.4	
less	1.77	1.77	1.70		1.77	
libg++	2.4	2.4	2.3			2.4
m4	1.0.3	1.0.3				
make	3.68	3.67	3.63		3.64	3.68
patch	2.1	2.1				2.1
perl	4.0.36	4.0.36	4.0.35			
rcs	5.6	5.6.0.1				
sed	2.03	1.18				2.03
shar	3.52	3.52				
shellutils	1.8	1.8			1.8	
tar	1.11.2	1.11.2	1.11.2		1.11.2	1.11.2
texi2roff	2.0	2.0				
texinfo	3.1	3.1				3.1
textutils	1.6	1.6			1.4	
time	1.4	1.5				
xfig	2.1.pl7					
xnetlib	3.4	3.4				

DESY C++ User Interest Group Formed

by Marcus Speh, II. Theorie

C++ is rapidly becoming the most popular object-oriented programming language, and there are several projects within High Energy Physics [HEP] using it. General questions about C++ HEP applications are discussed in an international HEPnet group, `hepnet.lang.c++`.

In August '93, the "DESY C++ User Interest Group" [DESY++] was formed as a group of physicists and computer scientists working with C++ at DESY. The group uses the local newsgroup `desy.c++` (`DESY.CPP` on VAX) for announcements and discussion.

The agenda of this interest group includes:

- Compiler questions: problems in porting between/performance of/understanding the various C++ compilers at DESY.
- C++ and OOP applications at DESY: at the outset, group members posted descriptions of their work and background. A concise overview of research activities employing C++ has been written up.
- Book reviews and articles.
- Programming/learning C++ : no question is too small for this group. Information on courses in the Hamburg area (such as J. Milewski's upcoming course for Hamburg University), and suitable tutorial material (now available for PC, VAX, and UNIX).
- Lectures/Talks: as a first instance, the video lectures for the GISMO C++ toolkit for Monte Carlo analysis and simulation in HEP experiments were shown at DESY in October (and will be shown again in November).
- Informal meetings: these shall take place irregularly and on demand only. Minutes of the first meeting can be retrieved from `ftp.desy.de` in the file `/pub/c++/news/930902.txt`.

DESY is also hosting a wealth of documents on C++ and especially HEP applications of C++, which are accessible via the World Wide Web. If you have further questions or suggestions, please contact me `<marcus@x4u.desy.de>`.

How to Send a FAX Using Email from any Computer at DESY

by Jan Hendrik Peters

We have installed a FAX modem on the central VAX which is directly connected to the email system. This allows users to send faxes from any computer at DESY using the local email system. Currently two types of messages can be sent: normal text and PostScript files. These text or PostScript files can either be incorporated into the message or sent as attachments.

A fax address looks almost like a standard email address: *fax-address*@text-fax.desy.de, where *fax-address* is either the fax-number of the recipient, or a more complete set of information including fax-number, the persons name, his/her group within the organisation, and so on. The more information you give, the faster it will be delivered within the receiving company or institution. Here are some examples:

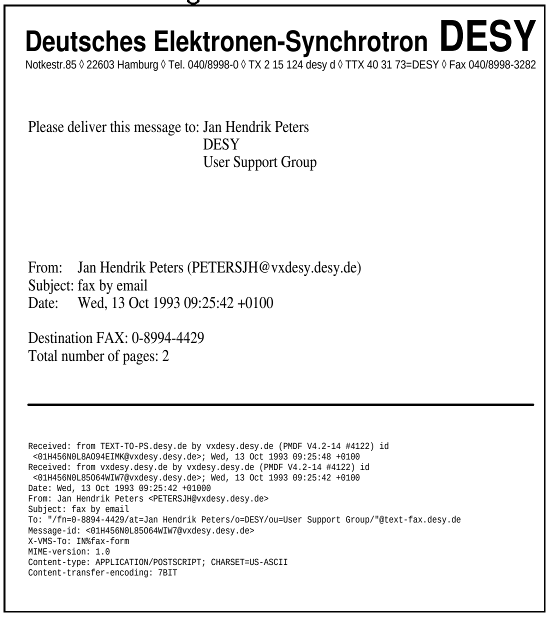
1. 4429@text-fax.desy.de
2. "/FN=0-001-303-449-1027/AT=M. Katherine Sills
   /TN=(303) 449 2673/O=HTI, Inc
   /OU=4700 Sterling Drive
   /OU=Boulder, Co 80301
   /OU=USA/"@text-fax.desy.de

3. "/fn=0-8994-4429/at=Jan Hendrik Peters /o=DESY/ou=User Support Group/" @text-fax.desy.de

where the tags used in the second and third example have the following meaning:

at	attention (usually a person's name)
o	organisation or company
ou	organisational unit or department
fn	phone number of the destination fax device dial 0-fax-number for external fax addresses
tn	recipient's telephone number
sfn	sender's fax number
stn	sender's telephone number

The fax that is sent consists of a DESY header page displaying the standard DESY letter head, the information you supplied in the fax-address, your email address, the subject field and date of your message, and the complete mail header. The second and following pages display the message which you included in your mail message.



Deutsches Elektronen-Synchrotron DESY

Notkestr.85 ◊ 22603 Hamburg ◊ Tel. 040/8998-0 ◊ TX 2 15 124 desy d ◊ TTX 40 31 73=DESY ◊ Fax 040/8998-3282

Please deliver this message to: Jan Hendrik Peters
DESY
User Support Group

From: Jan Hendrik Peters (PETERSJH@vxdesy.desy.de)
Subject: fax by email
Date: Wed, 13 Oct 1993 09:25:42 +0100

Destination FAX: 0-8994-4429
Total number of pages: 2

```
Received: from TEXT-TO-PS.desy.de by vxdesy.desy.de (PMDF V4.2-14 #4122) id
 <01H456N0L8AO94EIMK@vxdesy.desy.de>; Wed, 13 Oct 1993 09:25:48 +0100
Received: from vxdesy.desy.de by vxdesy.desy.de (PMDF V4.2-14 #4122) id
 <01H456N0L85O64WIW7@vxdesy.desy.de>; Wed, 13 Oct 1993 09:25:42 +0100
Date: Wed, 13 Oct 1993 09:25:42 +01000
From: Jan Hendrik Peters <PETERSJH@vxdesy.desy.de>
Subject: fax by email
To: "/fn=0-8894-4429/at=Jan Hendrik Peters/o=DESY/ou=User Support Group/"@text-fax.desy.de
Message-id: <01H456N0L85O64WIW7@vxdesy.desy.de>
X-VMS-To: IN%fax-form
MIME-version: 1.0
Content-type: APPLICATION/POSTSCRIPT; CHARSET=US-ASCII
Content-transfer-encoding: 7BIT
```

On successful delivery you will receive a receipt from the fax server which contains the destination fax number and the time needed for the transfer of the document. If the fax number cannot be reached, the fax server will try to deliver the message for 3 days and will afterwards return the message header to you.

Sending Faxes from the Central VAX

On the VAX there are two ways of sending faxes. One way uses the address syntax described above, the other uses a pop-up form which just needs to be filled out with the necessary information. The information given is passed in the correct format to the underlying mail program.

From VMS MAIL the fax form is opened up as follows:

$ mail
MAIL> send
To: IN%fax-form

The form will then pop up and you need to give the required information (fax number of the receiving device preceded by 0 for an external destination) and optional information like name and department of the recipient. Complete addresses can be stored under a nickname (alias) in your personal PMDF database. Press ctrl-z to leave the form and continue as if you were sending a regular mail.

To send a fax from VMS without using the fax-form you need to specify an additional pair of quotation marks around the address, e.g.:
MAIL> send
To: IN%"'/fn=0 (089) 12 34 56
/at=Herr A. Beispiel/'@text-fax.desy.de"
(Note, that you have to use different quotation marks for each level)

Using the new PMDF MAIL interface you no longer have to specify the IN% or use the additional set of quotation marks. The above example would look like this:

$ pmdf mail
EMAIL> send
To: "/fn=0 (089) 12 34 56
/at=Herr A. Beispiel/"@text-fax.desy.de

Sending Faxes from the IBM

On the IBM call the SENDMAIL program and specify the fax-address in the To: field of the main panel. Remember to specify the leading 0 for external connections:

To: "/fn=0(089)123456/at=A. Beispiel/"
@text-fax.desy.de

Fax addresses can be stored in the nick name files like normal mail addresses:
:nick. fax-nick
:email. "/fn=0089-123456/at=A. Beispiel/"
@text-fax.desy.de

Sending Faxes from Unix

On Unix systems use any of the email interfaces to send a fax just as if you were sending a regular mail. Either specify the fax address (don't forget the leading 0 for external destinations) as a parameter of the mail command or use the prompt or template given by the mail user interface. When using the mail prompt or template, give the fax address as shown below:

To : "/fn=0 001 333 111 5678
/at=someone in the US/"@text-fax.desy.de

If you specify the fax address as a parameter of the mail command, be sure to enclose the complete address in single quotation marks, e.g.:

[host] ~ $ elm '"/fn=0 001 333 111 5678
/at=someone in the US/"
@text-fax.desy.de'
[host] ~ $ pine '"/fn=0 001 333 111 5678
/at=someone in the US/"
@text-fax.desy.de'

Fax addresses can also be stored in the alias or address book files of the mail user interfaces.

Mailing from Unix Part I : pine

by Katherine Wipf

In the past few months, many of DESY's IBM and VAX users have been gaining experience with Unix. It hasn't always been easy, with many new concepts to get used to and very little documentation to explain them. One of the most difficult things to adjust to, apart from learning a new editor, is a new mail system. Both NEWLIB and VMS users are accustomed to typing the command `MAIL` in order to enter a user-friendly mail interface in which the next step is fairly easy to find. If you type `mail` on a Unix system, you also enter a mail interface (provided you have some mail waiting to be read), but the next step is not at all obvious. Fortunately there are several user-friendly mail programs (for example *pine, elm, mmh*) available on DESY's central Unix machines, and I strongly recommend that you learn one of these and NEVER EVER use the Unix `mail` command. In this article I would like to introduce you to the *pine* mail program.

Because *pine* is designed for an alphanumeric terminal, it will run just as well on a Falco or VT-terminal as it does on an X-terminal. It is very straight-forward and easy to use, and although it is designed primarily for novice users, it offers all the basic functions NEWLIB and VMS users are accustomed to. When you start up *pine* with the command `pine`, you find yourself in a menu (similar to MAIL on the IBM) from which you can select an action (i.e. read mail, send mail, update address book, etc.). It is also possible (as on the IBM) to bypass the menu and go directly to reading (with the command `pine -i`) or sending mail (with the command `pine` *e-mail_address*). A very nice feature of *pine* is that the relevant commands are always listed at the bottom of the screen and an on-line help is always available, even when you are reading or writing messages with *pine's* integrated pager or editor.

In one respect *pine* goes beyond what most mail programs offer: it is one of the first mailers to have MIME capabilities. MIME (Multipurpose Internet Mail Extension) is a new standard for sending multimedia e-mail (i.e. including binary data in Internet mail messages), and though it is not widely supported yet, its use is growing.

Sending Mail

In order to send mail messages from *pine* you need to select *C* (for *compose message*) from the main menu. You will then enter a new screen with the following prompts at the top:

```
To      :
Cc      :
Attchmnt:
Subject :
```

The `To:` and `Cc:` fields, as you might expect, are for entering the addresses of the people who should receive your mail or copies of it. You can type more than one address, separated by commas, and the ⟨RETURN⟩ key will bring you from one field to the next. ⟨CTRL⟩G will provide help if you are uncertain of the address format. ⟨CTRL⟩T offers rather a nice feature which allows you to inspect your *pine* address book and use the *M* command to select an address for the `To:` or `Cc:` fields.

The `Attchmnt:` field is part of *pine's* MIME support and allows you to attache files, images, other messages, sounds, or even videos to your message. If the recipient has a MIME-based mailer, he should be able to read these attachments. Even if his mailer does not support MIME, he should be able to read the text of your message and may be able to extract the attached files, although this is generally difficult in the case of binary files. To attach files, type the file names, separated by commas, into the `Attchmnt:` field. If you have forgotten the name or can't remember which directory the file is in, you will find that ⟨CTRL⟩T is very useful. It produces a display of your directory tree in which you can move around with the arrow keys and select a file with the *S* command. The attached files will not actually be shown in your message, but you can always tell that a file has been properly attached when its name and size appear in the `Attchmnt:` field (this occurs once you move the cursor out of the `Attchmnt:` field). If you wish you can also add a description of the file between the double quotation marks ("") which appear after the file size.

You should fill in the `Subject:` field so that the mail recipient knows what your message is about. For users who subscribe to several mailing lists and receive lots of mail, the message subject is very important for sorting mail and deciding what needs to be read immediately and what can wait until later. Once the `Subject:` field is completed, press ⟨RETURN⟩ to start editing your message.

For composing messages, *pine* offers a very simple editor which is a heavily modified version of micro emacs. The important commands are displayed at the bottom of the screen (the ^ symbol stand for the ⟨CTRL⟩ key) and ⟨CTRL⟩G provides a complete list of the available commands. Note that the ⟨CTRL⟩R command (for reading in a file) distinguishes itself from the `Attchmnt:` field in that it incorporates the file contents into your message text rather than attaching them as a separate part. Once the message is complete you can send it off with ⟨CTRL⟩X (*pine* will automatically save a copy of the mail in a folder called `sent-mail`). It is also possible to cancel the message with ⟨CTRL⟩C or, if you would like to finish the message later, you can use ⟨CTRL⟩O to postpone sending. Each time you start composing a message *pine* will check if you have a postponed message and ask if you would like to continue it. You can answer no and start a new mail message without deleting the postponed message.

Reading Mail

pine allows you to sort and store your mail in so-called folders (similar to the folders in VMS mail or the log files on the IBM). All incoming mail is first stored in a folder named INBOX and the main menu *I* command (for *folder index*) will show you an index of your new mail. For each message you are shown one line containing the date it was sent, the sender, and the subject. The number in parentheses in front of the subject is the number of characters contained in the message. A "+" symbol at the beginning of the line indicates that the message was sent directly to you (i.e. you didn't get it as part of a `cc:` or mailing list). If the message is new (unread) you will see an "N" in front of the message number.

In order to read a message you should move the highlight onto the message and press the ⟨RETURN⟩ key or use the *V* (*ViewMsg*) command in order to display the contents. There are a number of ways of moving the highlight: with the arrow keys, with the *N* (*NextMsg*) and *P* (*PrevMsg*) commands, with the *J* (*Jump*) command (which will prompt you for the number of the message you would like to highlight), or with the *W* (*WhereIs*) command (which will prompt you for a string to search for in the sender/subject of the message). Once you are reading a message you can use the commands:

- *D* - to delete the message
- *U* - to undelete it
- *R* - to send a reply
- *F* - to forward the message to someone else
- *Y* - to print the message on your default printer (use the *S* command from the main menu to set the printer)
- *S* - to save the message in a folder
- *E* - to save the message in a unix file (called exporting)
- *V* - to view a file attached to the message

If you send a reply to a message and then return to the INBOX index with the *I* command, you will see that the "N" (for a new, unread message) has been replaced by an "A" (indicating that the message has been answered).

Working with Folders

Creating and Filling Folders

Every message you receive with *pine* must be in a folder. A few folders are automatically created by *pine*, such as the INBOX and `sent-mail` folders. At the beginning of each month *pine* starts a new `sent-mail` folder and asks you if you would like to delete the old one or move the messages to a folder named `sent-mail-`*month-year*. In addition to these *pine*-created folders, you can create as many of your own folders as you like, using one of the following two methods:

1. Select *L* (for *folder list*) from the main menu and then use *A* (*Add*) and type in the folder name at the prompt.
2. Save a message and type in a new name when *pine* prompts you for a folder.

Any valid Unix file name can be used as a folder name (although you should avoid special characters like the "/"). Folders are filled by saving messages to them. When saving a message, you can type in the name of an existing or new folder or you can use ⟨CTRL⟩T in order to get a list of your folders from which you can select with the *S* command.

Viewing Folders

The main menu command *L* (for *folder list*) will bring you to a screen where you can either produce a list of all your folders or of your subscribed newsgroups (provided you have subscribed to newsgroups using a standard newsreader like *tin* or *xrn*). When you first enter this screen you will only see place-holders for these lists, which you can expand by moving the highlight onto the appropriate place-holder (bearing the text `[ Select Here to See Expanded List ]`) and pressing the ⟨RETURN⟩ key or using the *V*(*[Select]*) command.

Once you have a list of all your folders, you can use the commands:

D - to delete a folder
A - to add a new folder (*see above*)
R - to rename a folder
Y - to print the current screen

You can also select a specific folder to view in the same way you select a message to read (except that the *J*(*Jump*) command is not available). You will then be shown a folder index in the same format as the index of the `INBOX` folder. All the commands available for new mail (such as *R* for reply, *E* for export, etc.) can also be used on old mail saved in a folder. You can move messages from the current folder to a different folder by using the *S* (*Save*) command and then deleting the message with *D*.

If you expand the place-holder under the heading ``News-collection <Subscribed-Groups>'') you will see a list of the newsgroups you usually view with your favorite newsreader. *pine* treats the newsgroups as folders, so you can read the news in any group just as you would read the messages in a folder. Note that *pine* only allows read access to the groups - you cannot post to a group or send follow-ups from *pine*. If you try to use the *R* (*Reply*) command on a news item, you will find that the reply is sent to the author of the article.

Creating and Using a Personal Address Book

pine's personal address book is a very convenient way to store frequently used addresses so that you don't have to remember the complicated syntax. The concept is similar to that of the nicknames files used on the IBM, but the creation and maintenance are quite a bit simpler. Selecting *A* (for *Address Book*) from the main menu will show you your address book (which will be empty the first time you do this). Each entry in the address book consists of three parts: a nickname (usually short and easy to remember), a full name (the person's real name and other optional information), and an e-mail address (in the same format you would use in the `To:` field of the composer). It is possible to have entries, called distribution or mailing lists, in which the nickname refers to a whole list of addresses rather than just a single one. The address book is sorted alphabetically by the full name (generally listed last name first) with distribution lists sorted at the end.

The *A* command (for *Add*) allows you to add a new address to the book. You will be prompted for the full name (last name first), a nickname, and the full e-mail address. Instead of typing in the full name and address by hand, it is also possible to extract the e-mail address from a mail you have received. You can use the *T* command (for *TakeAddr*) either directly when you are reading the mail or when you are looking at the folder index and have the highlight on the appropriate mail. You will be prompted for a nickname and are given the chance to correct the full name or address if necessary. As soon as you have completed the entry, it will appear in the address book display. Distribution lists can be created with the *S* command (for *CreateList*). You will be prompted for a long name or description of the list (this name will appear at the beginning of the `To:` field when you send to the list), a nickname, and the e-mail addresses (you can type in full addresses or nicknames - even list nicknames are allowed).

Any of the entries in the address book can be modified by placing the highlight in the corresponding field and using the *E* command (for *Edit*). The text you wish to change will appear at the bottom of the screen and you can edit it using the delete and arrow keys. If you need to add addresses to a distribution list, you can do this with the *Z* command (for *AddToList*). You will be prompted for a single address to add to the list (repeat the procedure to add several addresses). You can delete any entry in the address book by placing the highlight on the corresponding nickname and using the *D* command (for *Delete*). If you only wish to delete one or two addresses from a distribution list (and not the entire list) you can do this by placing the highlight on the e-mail address before giving the delete command. You can make a hardcopy of your address book using the *Y* command (for *Print*).

Remote Login Without a DESY Computer Account

by Jan Hendrik Peters

Visitors quite often need to access their home computers while they are at DESY. Not all of the visitors, however, have a DESY computer account. In this paper we describe how you can access remote computers without having an account at DESY. There are several different methods of connecting to a remote computer, but all of them require the knowledge of the IP-address or IP-number of the remote host.

Connect to a Remote Host Using a VT Terminal

VT terminals can be found in almost all terminal pools at DESY, especially in the Computer Center and in the hallways of building 1. Most of the public VT terminals are connected to the DESY V.24 Switch, but some are directly connected to a terminal server. You need to be connected to a terminal server to set up a connection to a remote host. If the terminal is connected to the V.24 Switch, a prompt is displayed asking for a destination. Connection to the terminal server is established by choosing the destination LAT and pressing the RETURN key until one of the following terminal server prompts appears:

MIClocal>
Server>
Local>

- To connect to a Unix or VMS computer, type:

  MIClocal> **c ip-address-of-unix-or-vms-host**
  e.g.
  MIClocal> **c vxcern.cern.ch**

- To connect to an IBM system in full screen mode, type:

  MIClocal> c ibm

  and at the bottom of the next screen give your userid and the IP-address of the remote IBM system you want to connect to, e.g.:

  `login name (and host name if not DESY):`

  petersjh slacvm.slac.stanford.edu

  Precede your username with a colon (:) if you don't want to change the terminal settings.

  Press the *help* key and choose option *k* or press *ctrl-p* to obtain a listing of the PFkey assignments.

To leave the session, LOGOFF from the host or press ctrl-L to return to the terminal server prompt and type *disconnect*. The *break* key will disconnect you from the terminal server.

Connect to a Remote Host Using an X-Terminal

A connection to a remote host from an X-terminal without an account on the DESY unix systems is possible if the X-terminal is connected to a chooser program. If you see instead the login screen from the HP-Cluster or from the SGI Cluster, x4u, try to go back to the chooser panel by selecting the restart server button in the *options* menu of the HP, or by typing ctrl-c on the SGI.
Depending on the make of the X-terminal different choosers will appear.
On an NCD Terminal select the *Terminal Session* window and use the *Host:* input field to connect to the remote host.
On the Tektronix Chooser select the *telnet* option by clicking on the respective button, then click once on *localhost*. A screen with an error message will appear, press the *return* key and open a connection to the remote host.

- To connect to a Unix or VMS computer, type:

  TELNET> **open ip-address-of-unix-or-vms-host**
  e.g.:
  TELNET> open unixhub.slac.stanford.edu
  or on the NCD:
  Host: unixhub.slac.stanford.edu

- To connect to an IBM system in full screen mode, type:

  TELNET> open sun01a
  or on the NCD:
  Host: sun01a

  After the `login:` prompt from the terminal emulation machine, type the IP-address of the host followed by an exclamation mark (!), e.g.:

  `login:` cernvm.cern.ch!

  or simply:

  `login:` desy!

  to connect to the DESY full screen emulation program. At the bottom of the next screen, type your userid and optionally the IP-address of the IBM system you want to connect to, e.g.:

  `login name (and host name if not CERNVM.CERN.CH):` petersjh

  or in the second case

  `login name (and host name if not DESY):` petersjh cernvm.cern.ch

  Precede your username with a colon (:) if you don't want to change the terminal settings.

  Press the *help* key and choose option *k* or press *ctrl-p* to obtain a listing of the PFkey assignments.

To leave the session, LOGOFF from the host or press ctrl-] and type *quit* after the TELNET> prompt.

Including EPS graphics in TEX documents

by Peter K. Schilling

ENCAPSULATED POSTSCRIPT (EPS) graphics, which are generated by a variety of programs, may be included in TEX documents by using `dvips` together with some sets of macros to be used in your TEX source ("TEX source" means sources using Plain-TEX, LATEX, AMS-TEX, ...).

On the Apollo-cluster, the HP-cluster, the SGIs, and the IBM mainframe, the following packages are available: `epsfig` (by Sebastian Rahtz), `epsf` (by Tomas Rokicki), and `psfig` (by Trevor J. Darrell). A summary of the macros is given in the table at the end of this article. It is recommended to use the macros of the `epsfig` package.

Using the `epsfig` package with LATEX

Basic examples and hints

Here are some simple examples for LATEX users which show how to use `\epsfig` to include centered graphics in a `figure` environment:

By default, the width and height of the figure are taken from the "BoundingBox" of the figure.

```
\documentstyle[epsfig]{article}
\begin{document}
\begin{figure}
\begin{center}
\epsfig{file=abc.eps}
 % Include graphics at 'natural' size
\end{center}
\caption{Figure Caption}
\end{figure}
\end{document}
```

If width is specified, the figure is scaled so that it has the specified width. Its height changes proportionally. (Clearly: If height is specified, the figure is scaled so that it has the specified height. Its width changes proportionally. If both width and height are specified, the figure is scaled anamorphically.)

```
\documentstyle[epsfig]{article}
\begin{document}
\begin{figure}
\begin{center}
\epsfig{file=abc.eps,width=8cm}
 % Include graphics scaling width to 8 cm
\end{center}
\caption{Figure Caption}
\end{figure}
\end{document}
```

Hint 1: *All* length specifications must have *dimensions* (mm,cm,bp, . . .)!

Hint 2: There *should not* be any blanks *before* any of the keyword parameters in the \epsfig macro, i.e.
\epsfig{figure=abc.eps,␣width=5cm}
is wrong (␣ denotes a blank),
\epsfig{figure=abc.eps,width=5cm}
is correct.

More macros of the epsfig package

Besides \epsfig you may want to use the following macros:

\psdraft	draws an outline box but doesn't include the figure in the DVI file. Useful for previewing.
\psfull	includes the figure in the DVI file (default).
\psscalefirst	width= or height= specifies the size of the figure before rotation.
\psrotatefirst	(default) width= or height= specifies the size of the figure after rotation. Asymmetric figures will appear to shrink.

Some more examples

The included graphic is rotated first, then it is scaled to have a width of 5cm:

```
\epsfig{figure=abc.eps,width=5cm,angle=90}
```

The included graphic is first scaled to have a width of 5cm, then it is rotated:

```
\psscalefirst
\epsfig{figure=abc.eps,width=5cm,angle=90}
```

Using the epsfig package with other formats

Using the epsfig package with Plain TEX

Here are some simple examples for Plain TEX users which show how to use \epsfig to include graphics centered on a line (if the graphic is not to be centered, omit the \centerline{...}):

By default, the width and height of the figure are taken from the "BoundingBox" of the figure.

```
\input epsfig.sty
 % to be given once
 % for the first usage of \epsfig
\centerline{\epsfig{file=abc.eps}}
```

If width is specified, the figure is scaled so that it has the specified width. Its height changes proportionally. (Clearly: If height is specified, the figure is scaled so that it has the specified height. Its width changes proportionally. If both width and height are specified, the figure is scaled anamorphically.)

```
\input epsfig.sty
 % to be given once
 % for the first usage of \epsfig
\centerline{\epsfig{file=abc.eps,width=8cm}}
```

Using the `epsfig` *package with AMS-TEX*

Since AMS-TEX uses the character `@` as an "active" character, you have to use the following line as the first line in your AMS-TEX file:
`\catcode‘@=12\input epsfig.sty\catcode‘@=13`

Here are some simple examples for AMS-TEX users which show how to use `\epsfig` to include graphics centered on a line (if the graphic is not to be centered, omit the `\centerline{...}`):

By default, the width and height of the figure are taken from the "BoundingBox" of the figure.

```
\catcode‘@=12\input epsfig.sty\catcode‘@=13
 % only once at top of file
\centerline{\epsfig{file=abc.eps}}
```

If `width` is specified, the figure is scaled so that it has the specified width. Its height changes proportionally. (Clearly: If `height` is specified, the figure is scaled so that it has the specified height. Its width changes proportionally. If both `width` and `height` are specified, the figure is scaled anamorphically.)

```
\catcode‘@=12\input epsfig.sty\catcode‘@=13
 % only once at top of file
\centerline{\epsfig{file=abc.eps,width=8cm}}
```

Using the `epsfig` *package with* `cpzphysc` *and* `ppzphysc`

Since cpzphysc and ppzphysc use the character `@` as an "active" character, you have to use the following line as the first line in your cpzphysc and ppzphysc files:
`\catcode‘@=12\input epsfig.sty\catcode‘@=13`

Here are some simple examples for cpzphysc and ppzphysc users which show how to use `\epsfig` to include graphics centered on a line (if the graphic is not to be centered, omit the `\centerline{...}`) and within a figure:

By default, the width and height of the figure are taken from the "BoundingBox" of the figure.

```
\catcode‘@=12\input epsfig.sty\catcode‘@=13
 % only once at top of file
\centerline{\epsfig{file=abc.eps}}
```

If `width` is specified, the figure is scaled so that it has the specified width. Its height changes proportionally. (Clearly: If `height` is specified, the figure is scaled so that it has the specified height. Its width changes proportionally. If both `width` and `height` are specified, the figure is scaled anamorphically.)

```
\catcode‘@=12\input epsfig.sty\catcode‘@=13
 % only once at top of file
\centerline{\epsfig{file=abc.eps,width=8cm}}
```

Note that you have to specify a height of zero in the `\begfig` macro.

```
\catcode‘@=12\input epsfig.sty\catcode‘@=13
 % only once at top of file
\begfig 0cm
\epsfig{figure=abc.eps,height=8cm}
\figure{1.}{Figure caption}
\endfig
```

Using the `epsf` package

Since the `epsf` macros are contained in the `epsfig` package, they can be used after `epsfig` has been loaded.

Use `\epsfbox{`*filename*`}` wherever you want to insert a vbox for a figure. Alternatively, you can type `\epsfbox[0 0 30 50]{`*filename*`}` to supply your own "BoundingBox", thus using the bounding box you specify instead of reading in the values from the file.

The effect will be to typeset the figure as a TEX box at the point of your `\epsfbox` command. By default, the graphic will have its 'natural' width (namely the width of its "BoundingBox", as described in *filename*). The TEX box will have depth zero.

You can enlarge or reduce the figure by saying
`\epsfxsize=`*length<dimen>*
(or `\epsfysize=`*length<dimen>*)
`\epsfbox{`*filename*`}`
instead. Then the width of the TEX box will be `\epsfxsize` and its height will be scaled proportionally (or the height will be `\epsfysize` and its width will be scaled proportionally). The width (and

height) is restored to zero after each use.
If you want TEX to report the size of the figure (as a message on your terminal when it processes each figure), say `\epsfverbosetrue`.

Using the `psfig` package

The macros can be used after `epsfig` has been loaded.

The `\psfig` macro is like `\epsfig` with the only difference that the keyword parameter `angle` is not supported.

Specifying the EPS data to be included under MVS

The EPS data to be included is specified as <*name.suffix*> in the `epsfig` macro (as well as in `epsf` and `psfig`). This corresponds to a file `'userid.name.suffix'`.
Example:
`\epsfig{figure=graph1.eps}`
i.e. <*name*>=`graph1` and <*suffix*>=`eps`.
The macro would look for the file `'userid.graph1.eps'` in the user's catalog.

To avoid flooding the user catalog with a large number of sequential EPS files, and to allow for using already existing or new libraries of EPS members, `dvips` has been modified to allow for using partitioned libraries of EPS-members. It works like this:

If <*name.suffix*> is not found in the user's catalog, `dvips` (and TEX as well) looks for the member `name` in a partitioned dataset (library) allocated as file `suffix`.

Example:
Suppose you use `\epsfig{figure=graph1.eps}` and there is no file `'userid.graph1.eps'` in your catalog, but `graph1` is a member in your library `'userid.my.graphs'`, then `dvips` (and TEX) will find the data if you do an `alloc f(eps) da('userid.my.graphs')` before calling TEX and `dvips`.

EPS file and "BoundingBox"

An Encapsulated PostScript (EPS) file is a special form of a PostScript file which is built according to certain rules. The most important ones are:

1. The file contains only one "page" of graphics,

2. In its preamble the file contains a "BoundingBox" line, which describes the rectangle circumscribing the graphics by giving the coordinates of its lower left and upper right corners, e.g.
   `%%BoundingBox: 0 0 596 842`

   The values given refer to the `bbllx`, `bblly`, `bburx`, and `bbury` coordinates described in the attached table.

   The implicit dimension of the numbers given is "points" (LATEX: "bp"); there are 72 "points" to 1 inch.

epsfig	epsf	psfig
\psdraft	\epsfbox{*filename*}	\psdraft
\psfull	or e.g.	\psfull
\psscalefirst	\epsfbox[0 0 30 50]{*filename*}	
\psrotatefirst	\epsfxsize=*length<dimen>*	
\epsfig{%	\epsfysize=*length<dimen>*	\psfig{%
file=*filename*,%	\epsfbox{*filename*}	file=*filename*,%
figure=*filename*,%	\epsfverbosetrue	height=*height<dim>*,%
height=*height<dim>*,%		width=*width<dim>*,%
width=*width<dim>*,%		bbllx=*bbllx<dim>*,%
bbllx=*bbllx<dim>*,%		bblly=*bblly<dim>*,%
bblly=*bblly<dim>*,%		bburx=*bburx<dim>*,%
bburx=*bburx<dim>*,%		bbury=*bbury<dim>*,%
bbury=*bbury<dim>*,%		rheight=*rheight<dim>*,%
rheight=*rheight<dim>*,%		rwidth=*rwidth<dim>*,%
rwidth=*rwidth<dim>*,%		clip=,%
clip=,%		}
angle=*angle*,%		
}		

Description of the keyword parameters

file=*filename*	specifies the EPS data to be included
figure=*filename*,	synonym for file
height=*height<dim>*	width of the figure
width=*width<dim>*	height of the figure
bbllx=*bbllx<dim>*	lower left x-coordinate of the "BoundingBox"
bblly=*bblly<dim>*	lower left y-coordinate of the "BoundingBox"
bburx=*bburx<dim>*	upper right x-coordinate of the "BoundingBox"
bbury=*bbury<dim>*	upper right y-coordinate of the "BoundingBox"
rheight=*rheight<dim>*	the reserved height of the figure, i.e., how big TEX actually thinks the figure is; defaults to height
rwidth=*rwidth<dim>*	the reserved width of the figure, i.e., how big TEX actually thinks the figure is; defaults to width
clip=	ensures that no portion of the figure will appear outside its "BoundingBox". clip= is a switch and takes no value, but the '=' must be present.
angle=*angle*	angle of rotation (degrees); no dimension *<dim>* to be given!

Note 1: *<dim>* indicates that a dimension must be given, e.g width=10cm

Note 2: The four values (all of them must be given if this option is used) bbllx, bblly, bburx, and bbury control the PostScript "BoundingBox". If these four values are specified *before* the file option, the macro will not try to open the PostScript file.

Note 3: The parameters of the epsfig and psfig macros *should not* contain blanks!

Table 1: Summary of macros for EPS graphics inclusion

User Support Group Publications

Document №	Title	**Current Version**
USG/90/1	Printing at DESY	October 17, 1990
USG/90/2	IBM/MVS Extended Architecture	December 12, 1990
USG/93/1	Reference Card for the *vi* Editor on Falcos and X-terminals	June 28, 1993
USG/93/2	Pico - a very Simple Editor on Unix	June 21, 1993
USG/93/3	How to use the Previewer *xdvi* and *ghostview*	June 29, 1993
USG/93/4	The UCO Diskette Copying Facility	*to be revised*
USG/93/5	X-Terminal *Choosers*	June 24, 1993
USG/93/6	Introduction to *NCSA Mosaic*	November 26, 1993
USG/93/7	*Emacs* Mini-Reference Card	August 4, 1993
USG/93/8	Introduction to *Unix* at DESY	November 30, 1993
USG/93/9	How to Send a FAX Using Email from any Computer at DESY	October 21, 1993
USG/93/10	Remote Login Without a DESY Computer Account	October 21, 1993
USG/93/11	Compiling *FORTRAN* and *C* Programs on *UNIX* Machines	December 14, 1993

These publications are available in the self-service area of the computer center and via World Wide Web (see the article *WWW Information Services* in this newsletter).