DESY Computing Newsletter No. 6 September 1994

A Publication of the DESY User Support Group

Contents

Editorial

by Michael Behrens

If this newsletter issue were to have a theme, it would be best named: *Recent Changes and the Effect of On-going Changes in the DESY Computing Environment*. The key reasons for most of the changes we experience are not technical but financial. Modern – mostly, but not always RISC–based – computers are so much less expensive than their predecessors, that we, as everybody else, are forced to change our computing environment. This change is sometimes buried deep in the hardware, and therefore of limited interest to an average user, but sometimes it goes far beyond this. Both types of changes are described in this Newsletter.

One important change has already happened: the introduction of the Alpha machines in the VMS cluster. By the end of the year they will replace the former VAX machines. The details and implications of this change are described in an article by the VMS group.

The next step you will see on the hardware side is the replacement of the current IBM MVS system with a compatible, but smaller and less powerful system. Prerequisite for this change was moving the bulk of the computing load off our existing IBM to the central Unix systems. An article by Dietrich Mönkemeyer describes the MVS replacement and the rationale behind it.

While the VMS changes have relatively little impact on the users because the new systems offer even more performance than the older ones, the reduction on the MVS side is somewhat harder since users have to move off the IBM to a different platform.

One of the tools that should ease this transition is TITUS, described in another article. It helps to move your files from the IBM to a Unix system, changing most IBM dependencies on the fly and creating files easily usable on the Unix side. The article *Emacs for NEWLIB Users* should help you to start working on a Unix system afterwards.

Changes are also occurring on the network side. The first wide area network we became acquainted with at DESY was Bitnet, which grew in the context of the IBM mainframes, and is now shrinking and will soon disappear. Most of it's functions are offered via the Internet, and Internet has already taken over the best part of the networking. How this change will affect you is described in a short article.

Many people have accounts on various computers at DESY and elsewhere, e.g. at their home institute. How to work remotely under the X protocol is described in the article *Using X from Remote*.

You will also find an article about ORACLE. DESY is finally entering the relational database area to a larger extent than with the former DB2 system on the IBM, and this article should give you some first ideas of how to start using ORACLE and what to expect. ORACLE is available on (almost) all systems, and we expect a much heavier usage than we had before for DB2.

You will also find the usual selection of *Questions and Answers from the UCO* and the updated list of USG publications, as well as a new list containing unix tools recommended by the USG.

This newsletter and all the earlier issues are also available from www under the URL `http://info.desy.de/UCO/documentation.html`.

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

The new DESY Alpha Cluster

by The VXDESY System Group: Volker Heynen, Wolfgang Krechlok, Helmut Kreiser, Till Poser

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

In the last *DESY Computing Newsletter No. 5* the future of the DESY VMS Cluster was described. This new mixed architecture Cluster (4 new Alphas, 1 VAX 9000, 2 VAX 6400) has now been running since the beginning of this year with the following current status:

- The DESY VMS Cluster consists of 4 CPU servers, AXP 3000/500S, running OpenVMS AXP. Each server contains 256 MByte of main memory and has 20 Cern units of CPU power plus the old VAX-Cluster (1 VAX 9000 and 2 VAX 6400) with 14 Cern units.
- The servers are connected via FDDI.
- Each server has 5 GByte of internal disks used for system, library, paging and swapping space.
- The cluster has about 60 GByte space on user disks and 8 GByte of scratch space.
- The node–names are: AXDSY**a..d**. At the moment 3 machines are in production (AXDSY**a..c**) and one machine is reserved for test purposes (this machine will come into production in the near future).

During the first half of the year, all users and nearly all software were migrated to the new disks connected to the Alphas. As all disks are served cluster-wide (on VAX and Alpha), users see no difference when logged on to Alpha or VAX. To give all users the same environment on both platforms, all products must be migrated or reinstalled.

It is strongly recommended to use only the Alphas in the future. The old VAXen will be phased out at the end of this year. All users should migrate their user-written programs to the AXP-platform as soon as possible (see "Using the Alpha" later in this article).

The first period shows that the new concept brings much more power for the users. The whole cluster is stable. The migration is easy to do and nearly all important software tools were installed, either as new AXP products or as converted versions of VAX products:

- DEC Products (Compilers, Applications, Software Tools)
- Cernlibs and programs
- Ghostscript, Ghostview, and GNUplot (see HELP GNU)
- Mosaic
- Multinet
- Naglib
- Reduce
- the complete TeX environment (see HELP TEX)
- X–Window support for NCD and Tektronics X–terminals

Some products are not available for OpenVMS AXP at the moment:

- Parts of the Gnu project (C, C++,...)
- GTS-Gral GKS (being phased out, please use DEC GKS)
- Maple V

Using the Alpha

There is no difference in using the new Alphas compared with the old VAXes. The Alphas are running **OpenVMS AXP**. The Alpha-CPU is a full 64-bit machine with a 64 bit instruction set and address space.
DEC has converted their products to the Alpha line. These are installed on the Alpha Cluster.
For using user-written programs one has to consider the following points:

Change of the Floating Point Format

The floating point representation has been changed. While the AXPs still remain Big-Endian, the floating-point format has been changed to IEEE standard (Perhaps in some special cases the accuracy will be not as good as on the VAX).

Compiler

The compilers are much more sensitive and will perhaps show errors during compilation which the old VAX Computers haven't shown. The compilers have a switch to get an "Old-Style Vax Compiler" in case of problems. It is possible that this switch will be removed in the future. In any case it would be wise to correct errors and warnings whenever they occur.

Migration for User-Written Programs

VAX and Alpha systems are not binary compatible. For user-written programs this means that programs which should run on an Alpha system have to be either recompiled and relinked or must be translated.

DEC provides a tool for the translation of VAX executables to AXP executables if one doesn't want to recompile and relink. It is called VEST, the VAX Environment Software Translator (see HELP VEST) and is available on the Alphas.
Some more help on migration can be found in HELP ALPHA.

CERN Library and GKS Support

The CERN Library has been ported to the Alpha platform and it is in many ways identical to the VAX installation. However, there is **no** GTS-GRAL GKS support. Users who rely on "dumb" graphics terminals, such as FALCO terminals or DEC VT terminals with REGIS graphics, are urged to use DEC GKS, which has been installed on the Alphas. In due time the CERN library will support a platform independent PHIGS implementation.

X Terminals

Recently support for X–Terminals (NCD and Tektronics) was made available. Users can directly select the Alpha machines from the X–Terminal chooser.

Batch Queues

At the moment the following public queues, defined as logicals, are available to all Alpha users:

QUEUE NAME	CPU	limit	Priority
SYS$FAST	30	sec.	4
SYS$SHORT	5	min.	3
SYS$MEDIUM	30	min.	2
SYS$LONG	5	hours	1

This can change in the future depending on the usage of the Alpha cluster.

Support and more Help

All questions about the AXP systems and the migration will be attended to by the User Consulting Office (Tel: 2510, uco@desy.de).
More help for using Alphas and migrating from the VAX platform can be found with HELP ALPHA.
For suggestions, information about missing products, or problems during run-time, please send a mail to system@vxdesy.desy.de

The IBM Mainframe: Current Status and Future Plans

by Dietrich Mönkemeyer, R1

The Current Status

DESY currently runs an IBM 9021-720 as an MVS mainframe system. Roughly 2,800 users are registered on this system with circa 500 different people using the machine daily. The IBM is now more of a file server and file manager (also for the PC and unix world) than a CPU server. As a file server it offers the following services:

Data Storage for the Experiments: The ZEUS data for the current '94 run is still being stored directly on the IBM, while H1 only uses the IBM as a backup for their data taking which runs directly on the SGI machines. It is true of both experiments that all their important data is also cataloged and stored under MVS.

Cartridge Access: Up to 4,000 cartridges are mounted daily, with around 1,000 of these mounts serving to transfer data over Ultranet to the unix systems.

Cartridge Management: The MVS mainframe runs a very efficient management system for circa 100,000 3490-cartridges residing in tape archives or in the ACS cartridge robot.

Backup Services: The ADSM Backup System keeps 3 backup versions of circa 1.5 million PC and unix files. These backups are made twice per week and stored on 2 IBM disks and roughly 125 ACS cartridges.

Cataloging and Storage of MVS Files: About 400,000 files are cataloged in the MVS catalog and approximately 250,000 of these are under the control of the Hierarchical Storage Manager (HSM).

Downsizing Plans

The current MVS mainframe is relatively expensive to run, particularly in comparison with the very low-priced unix computers. For this reason DESY will replace the IBM with a cheaper model by the end of the year. When deciding how much to reduce the mainframe, a number of factors have to be considered, such as which applications it supports and how easy they are to port to a different platform.

The HERA experiments are the largest CPU consumers on the IBM. From the beginning, the programs and data formats of the experiments have wisely been conceived so that the data analysis can take place on unix systems just as well as on the MVS mainframe. This means that there are no porting problems for these users, in fact the greater part of their analysis is already running on the SGI computers.

On the other hand there are a few special applications on the MVS mainframe which are very difficult to port to unix. In particular this applies to the SAP administration program, a number of graphics and database applications, as well as special programs written in Fortran IV whose conversion to unix would require a great effort or whose source code is no longer available. The same is true of a number of "black box" applications in the form of Newlib clists or full screen packages.

It is also necessary to guarantee access to the 400,000 MVS files until the test connection between unix computers and the ACS goes into production **and** a direct access to files in HSM format is possible from unix platforms (or until all these files have been copied to unix with the help of the mainframe).

Since the best part of the CPU-intensive applications no longer need to run on the MVS mainframe, there are no fundamental problems in replacing it with a smaller model. A good start has been made with the reduction of the CPU quotas for the 5 largest user groups which took place at the end of May. With a considerably smaller machine it will still be possible to provide the file server functions, the support for SAP, and the necessary resources for running the few non-portable applications.

In consideration of these factors, R1 plans to replace the MVS mainframe by the 15th of November of this year with two small IBM computers with the new CMOS technology. These machines provide 40% of the current performance at less than 10% of the current operating cost.

The future of the MVS mainframe beyond the next 3 years is unclear at present. Due to its reliability and security, a long term use as a special server – for example for SAP or ADSM – is imaginable, especially since the operating costs of the new CMOS systems are comparable to other platforms such as VMS or UNIX. Moreover the new MVS operating system, which has been ordered together with the new machines, includes new open system capabilities enabling Unix file systems etc. under MVS.

There is no reason to panic. The MVS system may be reduced again to about 25% of the current CPU power during the next 2 or 3 years, but there are definitely no plans to remove the MVS system altogether. Once the large experiments, H1 and ZEUS, are no longer using the MVS system, 25% of the CPU power and disk space will be sufficient to cover the server functions mentioned above as well as providing resources for the non-portable applications.

Transferring IBM Source Files to a Unix System

by Harald Butenschön and Katherine Wipf

Computer users who used to work on the IBM mainframe and are now converting to unix know the problems inherent in transferring large libraries to unix computers. Ftp can be used for this task, but it is often unsatisfactory because files are transfered with upper case names, the .f and .c extensions for programs are missing, and COSY libraries (in NEWLIB's compressed format) cannot be transferred at all. For this reason Harald Butenschön developed titus (**t**ransfer of **I**BM source files **t**o a **U**nix **S**ystem) to solve all these problems.

The UNIX command `titus [options...]` transfers and converts IBM source datasets (i.e. libraries and sequential datasets) into a UNIX directory. Titus automatically:

- makes an ftp connection to the IBM machine
- makes an investigation of the format of the dataset to determine if it is transferable
- transfers of the dataset in the appropriate format
- converts members into files of a UNIX directory
- recognizes and converts NEWLIB MACROs to include-statements/files for the appropriate language

NEWLIB compressed (COSY) libraries are automatically recognized and handled correctly. Since a NEWLIB library carries the language nature within the members, titus is able to consistently transform the members into files which have the proper suffixes for the UNIX compilers. Macros are renamed in a proper way and the MACRO statements are replaced accordingly. The only thing you have to ensure is that all macros are members of the library: if not, use GET/PUT statements before using titus.

Unfortunately titus cannot support exotic member names (e.g. numbers as first character or the characters .<(+|&!)¬−/,%>_?:=˜‘ , i.e. all special characters except #$@). These members have to be renamed before the transfer, otherwise they are omitted.

At present titus has been installed on the SGI and the HP clusters and manual pages explaining the detailed syntax and usage are available on all these machines (see `man titus`).

Emacs Commands for NEWLIB Users

by Jan Hendrik Peters

This article describes commands for editing files with the emacs editor and compares them to the corresponding NEWLIB editor commands. This information is also available as a single page documentation (DESY-USG/94/4) in the self service area of the computer center and in the documentation section of mosaic (http://info.desy.de/UCO/documentation.html).
NEWLIB commands are written as uppercase words (CMD), NEWLIB linecommands as lowercase characters (c) and keys as (⟨KEY⟩). Emacs commands consists of key sequences with the control key C- (usually labeled as ⟨CTRL⟩ on the keyboard), the meta key M- (⟨COMPOSE⟩ on Tektronics, ⟨ALT⟩ on NCD), or the escape key ESC (⟨ESC⟩). Key combinations (e.g. C-x) have to be pressed simultaneously. Key sequences that start with kp- are the keys of the keypad on the right hand side of the keyboard.

description	NEWLIB	emacs
start editor leave editor	NEWLIB QUIT	emacs & C-x C-c
edit file save file reset file to last saved version compare files	CH *member* SAVE *member* RESET *member* COMPARE *member1 member2*	C-x C-f *file* C-x C-s M-x revert-buffer M-x diff *file1 file2*
toggle overwrite/insert mode toggle automatic fill mode	⟨INSERT⟩ –	⟨INSERT⟩ M-x auto-fill-mode
insert a new line into text delete a character delete last typed character delete next word delete till end of line	i ⟨SHIFT⟩-⟨DELETE⟩ ⟨DELETE⟩ – ⟨REMOVE⟩	⟨RETURN⟩ C-d or ⟨REMOVE⟩ ⟨DELETE⟩ M-d C-k
split a line join two line	SPLIT SPLITJOIN	⟨RETURN⟩ C-d or ⟨DELETE⟩
mark region of text with keys mark region of text with mouse kill a line kill text region move a line move text region copy text region copy lines from other file	– – d dd .. dd **1)** m **2)** a or b **1)** mm .. mm **2)** a or b **1)** cc .. cc **2)** a or b **1)** LIST *file* **2)** *copy text region*	*set mark with* C-@ *or* C-⟨SPACE⟩ *and move cursor to end of region* *mark with mouse-1* C-k **1)** *mark region* **2)** *2-click mouse-3 or* C-w **1)** C-k **2)** C-y **1)** *mark region and* C-w **2)** C-y **1)** *mark region and* ⟨ESC⟩-w **2)** C-y **1)** C-x C-r *file* **2)** *copy text region*
undo last change cancel partially typed command	– –	C-x u C-g
show key settings help tutorial	SHOW PF HELP GUIDE	⟨HELP⟩ C-h a C-h t
find all lines with string search string change string top ,bottom up, down beginning, end of line	FIND *string* POINT *string*, P CS *old new* TOP, BOTTOM UP, DOWN –	– kp-2 *string*, kp-3 ⟨PF4⟩ *old new* kp-7, kp-1 ⟨PrevScrn⟩, ⟨NextScrn⟩ kp-4, kp-6

BITNET Migration – What is it Replaced by?

by Jan Hendrik Peters

In Europe the BITNET/EARN network is shrinking rapidly. In August 1992, BITNET had 3435 nodes, in August 1993, 3190 nodes remained, and now there are only 2505 nodes left.

The BITNET connectivity will be cut at DESY as well. Starting October 1st, 1994, no more BITNET services will be available on the IBM. For the time being, the BITNET service will be continued as a pure email service on the central VAX. Until now BITNET has supplied many network services to the user community for which replacements must now be found. Most of the existing services are available on the Internet in some form or other. This little article describes what other utilities you can use, either on the IBM or on other computer systems, to replace the services you are using nowadays.

Replace all BITNET Addresses by Internet Addresses

Whenever sending mail you should only use Internet addresses. In your nickname or alias files you should change all addresses containing references to BITNET or EARN and replace them by the Internet address of the host. BITNET addresses have the form USERID@HOST or USERID@HOST.BITNET where USERID and HOST may have a maximum of 8 characters. At DESY we have two BITNET hosts: DESYVAX (the central VAX) and DHHDESY3 (the IBM mainframe). Internet addresses have the form **userid@host.domain** where **userid** and **host** may have any length. The domain name for computers at DESY is **desy.de**. If you have problems finding the correct Internet address of a BITNET host, please contact the UCO.

The nickname files for ZEUS and H1 on the IBM have already been changed to comply with this new rule. These two nickname files are exported to the central VAX on a regular basis. All H1 and ZEUS users and distribution lists are now reachable worldwide with their alias (usually the last name of the person) if you use the following email address:

- **h1+**alias**@desy.de** for H1 users
- **zeus+**alias**@desy.de** for ZEUS users

To change the way incoming mail is being sent, send a mail to all your communication partners to tell them your new email address. Tell them to use

- **userid@dsyibm.desy.de** instead of USERID@DHHDESY3.BITNET for the IBM
- **userid@vxdesy.desy.de** instead of USERID@DESYVAX.BITNET for the VAX
- **userid@desy.de** if you have an account for the central Novell server, an account for the central VAX, or a special entry on the VAX.

Sending/Receiving Mail Messages on the IBM

Even after October 1st, it will be possible to send and to receive mail on the IBM. The only difference is that you should not use any BITNET addresses, but should use Internet addresses when sending mail. In your status display on the IBM you will no longer have any jobname of the type useridRR but only TCPSMT, where all mails sent to you on one day will be collected. To view the mails use the command READMAIL.

In order to receive mail via Internet on the IBM, your userid needs to be registered under RACF. If your userid is not yet under RACF, contact the RACF administrator of your group (see HELP RACGRPS on the IBM). If you have an account on a Unix system or VMS system you should consider moving your mail applications completely to one of these platforms.

Subscription to Mailing Lists

If you are subscribed to a mailing list with a BITNET address, unsubscribe from the mailing list using the BITNET address of the mailing list server, and resubscribe with your new address by either

- sending the mail to the Internet address of the mailing list server or
- add the new mail address to the subscribe command, e.g.
  `subscribe list-name userid@host.desy.de`
  where `host` can be one of the unix machines, vxdesy or dsyibm.

Note: you should do this before October, 1st!

Sending Interactive Messages with TELL

Some people use interactive messages (TELL) on BITNET to communicate with other users on BITNET. An equivalent tool called **talk** exists on the central VAX and on all unix systems and allows you to communicate with other users over the Internet. This tool, however, is not available on the IBM, so you have to use either the central VAX or one of the unix systems.

Sending Interactive Commands with TELL

Interactive commands on BITNET are usually sent to LISTSERV servers or other servers on BITNET to request files or certain information from a specific host. Most BITNET servers not only accept commands via interactive messages, but also accept commands that are sent with email. There are usually two mechanisms available:

- specify the command in the mail body
- specify the command in the Subject: line

To find out what kind of command interface your server accepts, send a mail to the server with the command HELP in either the Subject: line or in the mail body. Use the Internet address of the server instead of the BITNET address.

File Transfer using TRANSMIT/RECEIVE

The file transfer utility TRANSMIT for sending files and RECEIVE for receiving files from BITNET will no longer be available. You need to use the Internet program ftp to transfer the files. The Internet address of the IBM is dsyibm.desy.de and the Internet address of the central VAX is vxdesy.desy.de.

PINBOARD Information Service

The PINBOARD services have supplied an easy to use information service on the IBM. Pinboard messages for the H1 collaboration have also been exported to other IBM systems on BITNET. This service will no longer be available. Use the netnews system instead of the pinboards. On the central VAX the **nnews** command and on unix system the **tin** or **xrn** command will interface to netnews here at DESY.

The existing pinboards for H1 can all be found in the netnews system with names desy.h1.news, desy.h1.phan, etc. It will no longer be possible to post messages to the pinboards from the IBM. Please, use one of the unix systems or the central VAX instead.

Using X from Remote

by Michael Behrens

Using a remote computer from DESY, or using a DESY computer from a remote site is quite simple if you restrict yourself to character-type applications. Then you simply use `telnet` or `rlogin`. Using X to or from a remote computer is slightly more complicated, but nevertheless often necessary or simply convenient.

Trusted Hosts and Accounts

It is convenient for a user to change from one computer to another without specifying his/her password again and again. It is also a security enhancement, since nobody can spy your password on the network simply because it will not be sent. In order to omit the password, the other host must trust the host you are logged in to.

By default the central DESY computer trust each other. If you want to work with another computer you may configure it for your account by editing a $HOME/.rhosts file. This file contains a list of trusted computers and login names. To prevent unauthorized access to your account, you must protect this file so that nobody except you can write to it. All r* commands like rlogin, remsh, rsh and xrsh support this file. A sample file might look like this:

```
x4u2
x4u2.desy.de
server1.rrz.uni-hamburg.de my_other_login
```

Line 1 means: Trust all x4u2 logins with my default login name. Line 2 means: Trust all x4u2.desy.de logins with my default login name even outside of DESY. Line 3 means: Trust all logins from computer server1.rrz.uni-hamburg.de but only with the login name my_other_login.

To get the right name of the local machine, `rlogin` to the remote host and type `who am i` (UNIX), `who am i -R` (HP) or `echo $REMOTEHOST` (SGI). To get the right name of the remote machine, `rlogin` back from the remote host to the local machine and give the command as explained above.

X Connections Locally

If you only use the centrally managed systems at DESY – that is the `x4u`, `dice2`, `hermes`, `zarah`, `HP-cluster` and `Apollo-cluster` – then building the X connection is extremely simple. No password is required in this case because you have the same login id on all systems and because the systems trust each other. The command is simply `xrsh hostname`, and an xterm with a proper header will be created on the other host.

More Complicated X Connections

If you have different login names on the two systems, you have to specify the login name for the other computer in the `xrsh` command and grant access through the `.rhosts` file. This should work if the other computer understands the Magic-Cookie-Authorization as of X11R4. Try

```
xrsh -l user1 remote.host [-auth default]
```

to connect to the computer *remote.host* as user *user1* provided you know the "correct" name of this computer and have created a proper `.rhosts` file in the remote home directory of the remote computer. If the computer is not on the DESY list of computers, you must specify the `-auth default` option to tell xrsh to distribute your X11 authorization.

If this fails, then either `xrsh` is not installed or you used the wrong name for the remote computer. It is also possible that the /usr/bin/X11 path is missing, or that the remote system does not use Magic–Cookie authorization.

Problem1: xrsh not installed

`xrsh` is a command developed at DESY with a command line interface close to the MIT version of xrsh. It is available on the central DESY systems and in some other places within HEP. If you do not have it on the system you are working on, you could get yourself a copy (it is a script which in many cases should run without modifications) and run it from your own home directory, or you could ask your system manager to install it systemwide if several people on this computer might need it. You simply use `ftp` to get it from your preferred DESY host. In case of problems please contact the UCO. See also the manual pages of xrsh and do_xauth.

Problem2: Finding the right hostname

The correct host name for X–sessions can be quite different from the one you use e.g. for `telnet` or `ftp`, and it is not always easy to find (actually it is the name of the interface, and a computer may have several of them). See the section 'Trusted Hosts and Accounts' at the beginning or ask the local support staff, but they may not know it either. My preferred way to find out is to `telnet` to the remote system and then to telnet back to the x4u from there. On the first login *after* this telnet session, the x4u ("real" name x4u2) will say ...`Last login:` ...and the name of the interface used during the last (telnet) login.

Problem3: /usr/X11/bin path is missing

This is sometimes the case on Aix and Sun machines. If you come across this problem,ask your system administrator to link the /usr/X11/bin path to the actual installation.

Problem4: Magic–Cookie does not work

If the Magic–Cookie scheme to authorize you on the other side does not work, you have to use host-based authorization as described below:

```
xrsh -l user1 remote.host -auth xhost
```

Host–based Authorization – the complex Case

If the `xrsh` tools are not available or do not work in your environment and you want to build an X connection to a DESY computer, then you have to use the following procedure:

- Issue the command `xhost` +*hostname* on your local station, where *hostname* is the name of the computer you want to connect to.
- Find out the name of your workstation or X–terminal, e.g. myworkstation.desy.de:0
- `telnet` to the DESY computer from your station
- Create an xterm from the telnet session , e.g. `xterm -display myworkstation.desy.de:0 &`

Then you should get an xterm and can start working.

If all else Fails

Unix systems sometimes show surprising differences, and you may have problems in using X even if you try all possibilities explained above. In this case you should ask the UCO for help and they will try to find a solution for you.

Network Considerations

You should always bear in mind that X applications over a network have a considerable demand on bandwidth in contrast to character-type applications. Running X over a slow line is frustrating and will also affect all other users of this line seriously. You should therefore use it mainly on high speed lines, e.g. between DESY and the Hamburg University and not over transatlantic lines or lines to institutions with a small bandwidth. If you have questions in this context, please ask the UCO for advice.

The ORACLE Relational Database Management System

by Ulrich Naujokat

Available ORACLE Products and Platforms

In the autumn of 93 the first ORACLE RDBMS installations were done. After some testing we are now moving to a production environment. The supply contract consists of the following options and tools:

- Parallel Server option
- Distributed option
- Procedural option
- SQL*PLUS
- ORACLE*Forms/Menu
- ORACLE*ReportWriter
- ORACLE*Graphics
- ORACLE*Book
- SQL*Net and protocol
- PRO*C
- PRO*Fortran

Under our license agreement users can work on the following platforms:

- VAX and Alpha under open VMS
- all UNIX machines
- Novell and PC-LANs
- MS-DOS, MS-Windows, Windows NT
- Macintosh
- IBM ES 9000

However, not all components are available for all platforms at DESY.

When and Why use ORACLE RDBMS?

As there are several benefits to using ORACLE, you should check whether it will be useful for your specific applications. As a general rule, Oracle is useful when you need to handle small or large quantities of related data that need to be shared across different platforms and which may be updated simultaneously by several people. Oracle offers line mode (Plus) and graphical user interfaces (Forms, ReportWriter, Graphics, Book) and also allows manipulation from standard programming languages (Fortran and C) or from other commercial programs on PCs (e.g. Excel, Access, Word, etc).

Here is a short list of the basic Oracle features:

- Large quantities of data can be read and updated.
- Complex data selections from one or more tables can simply be retrieved by using SQL*PLUS, application programs or 4GL ORACLE tools e.g. FORMS, REPORTS, GRAPHICS.
- Many commercial software products are based on ORACLE RDBMS.
- Users of well known PC tools, e.g. EXCEL, have the possibility of accessing external ORACLE tables in their well known environment.
- National language support enables developers to design applications with respect to the native language of the users.
- ORACLE database tables can be accessed by SQL commands which are embedded in standard programming language programs.
- ORACLE applications can invoke programs written in third-generation programming languages, i.e. you can call Fortran and C from within ORACLE applications.
- Applications are portable across platforms. No matter which platform you used for application development, the application will be adapted to your working environment.
- Data can be shared among many users regardless of whether they run their application on the same computer or not.
- Data can be shared with other applications.
- Access to tables or subsets of data can be restricted for users depending on their defined role.

- The level of data integrity and recovery in case of errors can be defined by the users requirements.

Principles of Running ORACLE at DESY

Since it has a client-server architecture, ORACLE can be located on a stand-alone computer or run in a distributed processing environment, depending on the requirements of the applications.
The DESY User Support Group recommends running ORACLE in an environment where client applications run on the user's local computer and the central server resides on a central host. Currently, the central server runs on x4u2.
Benefits of this concept are:

- Since client applications are not responsible for performing any data processing, the computer which runs the server can be changed or replaced without requiring any changes in the application software.
- Because only requests and results are shipped over the network, traffic is kept to a minimum.
- Applications can be designed and optimized for presentation with no dependence on the physical location of data but with respect to the characteristics of the operation system.
- Maintenance tasks, such as installing new releases of server software, debugging, or backing up and recovering data can be supported more easily a the central server.
- There is a single, consistent version of your data stored in the central database.

Responsibility of USG and of the Users

DESY's User Support Group (USG) registers all ORACLE users regardless of whether they deal with the central or a local server. Besides this the USG:

- orders and installs the software
- trains and supports users
- publishes information and primers
- provides manuals
- sets up the ORACLE network
- maintains the central server
- creates and tunes the central server database
- provides disk space

Application development is not a task of USG. Users are responsible for their client applications but may consult USG. In special cases, e.g. if applications must be migrated from other systems to ORACLE, more assistance will be given by USG. If applications use the ORACLE database as a 'black box' in a distributed processing environment or if a 'stand-alone' local server is used, the applications owner is responsible for maintenance on the local site.

Getting an ORACLE Account

To use ORACLE you must be registered. Registration forms may be obtained from the UCO or can be printed from within mosaic (see the entry on DESY User Support page). This form will also inform about the current procedure for connecting to ORACLE RDBMS. Currently there are three distinguishable user groups:

- The first group of users can log in as the public user SCOTT. Their password is TIGER as described in the ORACLE user guides or known from classes. Users who connect to ORACLE as user SCOTT are only allowed to select data from some public tables.
- The second user group will work on a testing environment where first experience can be made and where new users can decide whether ORACLE meets their expectations or not. All test users work in a common environment where they can create and manipulate their own data tables.
- Last but not least, there is a production environment with its applications and users. Here users are well separated from each other.

Users who are already registered for ORACLE will be informed by electronic mail or via the newsgroup desy.oracle if changes in connecting procedures occur.

Some Basics about Databases and Languages

These explanations shall protect users from confusion. Especially those users who have already worked with other DBMS or spreadsheets may use the same terms but with different meanings.

The Structure of an ORACLE Database

The *table* is the basic unit of data storage in a database. Tables hold all of the user-accessible data. From the point of view of an ordinary user, tables are the only objects he deals with. Each table is defined with a name and a set of named columns with specified widths and data types. Once a table is created, valid rows of data can be inserted into it. Rows can be queried, deleted, or updated.
Tablespaces are the next bigger logical unit a database is divided into. Tablespaces are created by database administrators and are used to group objects in order to simplify administrative operations.
The ORACLE database has both a physical and a logical structure. The physical structure is determined by the operating system files and consists of data files, log files, and control files. The physical storage can be managed without effecting the logical structure of the database. The logical structure of the database is determined by one or more tablespaces which consist of tables, views, indexes, stored procedures, etc .

Structured Query Language SQL

SQL, an English like non-procedural language, is an international standard and thus the common language for all relational databases. The basic commands are easy to learn. SQL is used for all types of database activities. Programs written in SQL on other systems can often be moved to ORACLE databases with very little modification.

Procedural Language PL/SQL

PL/SQL extends SQL. It allows you to logically group statements. PL/SQL procedures can be stored in centralized libraries for use in different applications. PL/SQL blocks may consist of any number of SQL statements combined with control statements like IF .... THEN ....ELSE, repetition statements like FOR .... LOOP or WHILE....LOOP, and unstructured commands like EXIT or GOTO .
Variables in PL/SQL blocks must be declared and values are assigned in two ways. The first way is by using the assigning operator := (colon followed by equal sign). The second way is to SELECT database values INTO variables.
PL/SQL is the common language for developing application in ORACLE's Cooperative Development Environment (CDE).

Embedded SQL

ORACLE provides precompilers for embedding SQL statements into host programs. Available precompilers at DESY are Pro*C and Pro*FORTRAN.
Prefacing SQL commands with the EXEC SQL clause is the only requirement for embedding them. The precompilers will translate all EXEC SQL statements to runtime library calls.

*SQL*Plus*

SQL*PLUS is a program for executing SQL commands and PL/SQL blocks in line mode. In most cases it is powerful enough to serve the needs of database users. New users of ORACLE RDBMS should use SQL*PLUS just to gain first experiences. A brief description of SQL*PLUS is available in the self service area of the computer center (DESY-USG/94/1).

*SQL*Loader, Import, Export*

The ORACLE utilities Import and Export are used to write tables to and read tables from host files. Exported files should only be processed by the Import utility.
The only purpose for using these utilities is either for interchanging data between different ORACLE RDBMS or for maintenance. If data from ORACLE tables should be processed by other programs, you

must use embedded SQL or write the data to host files by invoking SQL*Reports or SQL*Plus. Storing data from host files into ORACLE tables must be done by using the SQL*Loader utility. SQL*Loader allows you to process multiple files of different file types. Which data should be processed in what manner must be defined in a control file. A short description about the loader facility is under preparation.

*Oracle*Forms, Reports, Graphics, Book*

These utilities are graphical user interfaces (gui) to the database. Forms supports the creation of full screen panels, where data can be retrieved, changed and inserted. Graphics can produce charts, plots and histograms from your data. Reports is a tool for writing reports, i.e. formatted output, of your data, including graphical charts or plots from graphics. With Book, help files are accessed and your own information can be presented with hypertext capabilities. A more detailed description of the gui tools will follow in one of the next newsletters.

Questions and Answers from the UCO

by Katherine Wipf

The User Consulting Office answers quite a variety of questions, but usually only a few people hear the answers. This column tries to bring the answers to more frequently asked questions to a larger audience.

. . . about Unix

◦ *Question:* How do I specify a "nice" level for my background processes?

◦ *Answer:* You can set a "nice" level with the command `nice -`*`level command`* `&`. For example, if you wanted to run your program, *myprog*, on the HP cluster you could use the command:

```
nice -12 myprog &
```

You can inspect your running HP jobs with the command `i r` and the resulting table will show you your nice level. On the x4u machine you should use either nqs or the loadlever for your batch jobs (see the corresponding news and man pages).

◦ *Question:* I've heard that the NAG library is now available on the HP cluster and the x4u2. How should I link my program if I want to use the NAGlib and is there any documentation available?

◦ *Answer:* The NAG library is a comprehensive collection of routines for the solution of numerical and statistical problems. If you want to use routines from the NAGlib, you should compile your program as follows:

```
x4u2:  f77 prog.f -lnag      -or-
       f77 prog.f -L /usr/local/lib
       -lnag
HP:    fort77 prog.f -L /usr/local/lib
       -lnag
```

An online help facility is available on the x4u2 with the command `naghelp` and the complete documentation is available in the DESY library, the UCO, and several terminal pools. See `news naglib` on either machine for more information about online documentation.

◦ *Question:* I cannot log on to the dice cluster from my X-terminal. I can type in my userid and password and no error message appears, but I am thrown off the system before any windows are opened. The strange thing is that I am able to log on via `telnet`.

∘ *Answer:* There are several possible causes for this problem. Here are the most common ones:

- You have exceeded your disk quota. In this case you will see a message telling you how much you are overdrawn when you log in via `telnet`. Delete or compress files until you have enough disk space (the command `quota -v` will show you how much of your quota you are using).
- You made changes in your .profile or .login file. These files generally contain a case statement, and if terminal output is produced within the X11 part of the statement, your start-up file will crash and you will be thrown off the system. This happens because the X11 part of the .profile and .login files are executed before any terminals are available to receive the output. Make sure that all possible command output in the X11 section of the case statement is redirected to /dev/null (e.g. `do_xauth axdesy 1>/dev/null 2>&1`).

∘ *Question:* I would like the emacs editor to start a new line automatically when I reach the right hand margin. Is this possible?

∘ *Answer:* Once you are in emacs, give the command (`M-` stand for the meta-key):

`M-x auto-fill-mode`

If you would like emacs to start in auto-fill mode automatically, add the following line to your .emacs file:

```
(setq text-mode-hook '(lambda ()
                        (auto-fill-mode 1)))
```

∘ *Question:* I use mosaic on the x4u2 in order to read netnews groups, check on people's e-mail addresses, etc.. I find it tiresome to go through several screens before I reach the page I want, especially since I generally want the same page each time. Is there a short-cut?

∘ *Answer:* NCSA Mosaic offers a feature which allows you to save your favorite pages. First access the page you wish to save, then click on *Navigate* in the menu bar at the top of the screen, and choose *Add Current To Hotlist* from the pull-down menu. This will save your page. Whenever you wish to access the page, click on *Navigate* and choose *Hotlist...* from the pull-down menu. A window will appear, displaying the names of the pages you have saved. Simply double-click on the page you wish to see.

∘ *Question:* I have accounts on several DESY computers and would like to set up my e-mail so that it all arrives on the same machine. How can I set forwarding addresses on the VXDESY-cluster, the IBM, and unix machines?

∘ *Answer:* Each machine type has a different procedure for setting a forwarding address:

VXDESY: Call the MAIL facility with the command `$ MAIL` and then give the command
`MAIL> set forward in%"""`*`email_address`*`"""`
Note that three sets of quotation marks are necessary (since a single set of quotes would be interpreted as a character string delimiter). Alternately you can use the PMDF mail facility which you start with the command `$ PMDF MAIL`. To set a forward in PMDF MAIL you only need to type in `EMAIL> set forward` *`email_address`*. In both cases you should check that the address has been set correctly with the command `show forward`. The answer should be:
`Your mail is being forwarded to`
`IN%"`*`email_address`*`"`.

IBM: Use the MAIL clist to set your forwarding address. Once you have given the command `MAIL`, choose the action F. You will be shown a panel where you can enter a forwarding address, press the <enter> key to have the syntax checked, and then press PF2 to activate forwarding.

UNIX: To forward your unix mail, you need to create a file in your home directory with the name `.forward`. This file should contain one line with your desired forwarding address in it.

No matter which machine you set a forwarding address on, you should always send yourself a test mail to make sure forwarding is working correctly. If you do not do this, you run the risk of losing important mails because they cannot be delivered to an incorrect forwarding address. Please also remember not to use BITnet addresses as forwarding addresses since BITnet will be discontinued in October of this year.

◦ *Question:* My default login shell is ksh and I would like to change it to zsh - how do I do this?

◦ *Answer:* To change your default shell on the HP-cluster, you should give the command `chsh` *`userid`* `/bin/zsh`. You will then have the zsh as your default shell from your next login onward. For all the SGI machines (x4u, dice, hermes), you need to log yourself on to the Apollo cluster with the command `telnet apollo`. On the Apollo, give the command `chsh`. You will be told what your current shell is and prompted for a new shell. The information about your new shell is distributed from the Apollo to the SGI machines every half-hour.

◦ *Question:* From time to time I need to check which items are available in the DESY "Lager." I have seen that the information is available over World Wide Web using Mosaic as a browser. Unfortunately the price of the items is not given here, but I have been told that the prices are listed on a similar IBM facility. Do I need to get an IBM account to see the full information?

◦ *Answer:* No! The price information is also available with the NCSA Mosaic browser. The "Lager" display is broader than the default text width offered by Mosaic, and this is why the prices are not visible. Either increase the size of your Mosaic window or use the horizontal scroll bar (just above the buttons at the bottom of the screen) to shift the display to the right.

The User Support Group Recommends ...

Many users who have moved to the central unix platforms from the IBM or VXDESY feel somewhat overwhelmed by the large variety of tools and programs offered there. Where VXDESY and the IBM offered an editor (with one or two alternatives), the unix platforms let the user choose between n editors (where n is a number determined by the system administrator). The situation is similar when it comes to mail utilities or newsreaders. This makes it difficult for the user to decide which program to use or even to determine what the choices are. For this reason the User Support Group has put together a list of programs which we recommend. This list is by no means comprehensive and we do not claim to list "the best editor,newsreader,... of all times." We consider the programs we list to be usable and, more importantly, they are fully supported on the central unix platforms. At present the list is very short, but we expect to extend and complete it in the future.

The USG Recommendations		
Unix Shell		**zsh**
Editor	*easy to learn:*	**pico**
	powerful:	**emacs, vi**
Mail Program		**pine, elm**
Newsreader	*for vt-type terminals:*	**tin**
	for X-terminals:	**xrn**
WWW Browser	*for vt-type terminals:*	**lynx**
	for X-terminals:	**mosaic**

There are many, many more programs and tools available on the unix platform. To get an idea of what is available you can try searching the DESY sofware catalog which is currently being built and tested. The search utility is available on www with the URL `http://info.desy.de/cgi-bin/catalog.pl` or by choosing *software catalog* form the DESY home page. Note that the programs listed in the catalog are not necessarily fully supported on the central unix platforms.

User Support Group Publications

Document N⁰	Title	Current Version
USG/90/1	Printing at DESY *(to be revised)*	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	December 22, 1993
USG/93/5	X-Terminal *Choosers*	June 24, 1993
USG/93/6	Introduction to *NCSA Mosaic*	December 14, 1993
USG/93/7	*Emacs* Mini-Reference Card	August 4, 1993
USG/93/8	Introduction to *Unix* at DESY	August 5, 1994
USG/93/9	How to Send a FAX Using Email from any Computer at DESY	October 22, 1993
USG/93/10	Remote Login Without a DESY Computer Account	October 22, 1993
USG/93/11	Compiling *FORTRAN* and *C* Programs on *Unix* Machines	December 14, 1993
USG/94/1	Basic Introduction to SQL*PLUS	January 31, 1994
USG/94/2d	Das DESY-Rechenzentrum	June 16, 1994
USG/94/2e	The DESY Computer Center	June 15, 1994
USG/94/3	How to Include PostScript and Encapsulated PostScript into LaTeX	September 5, 1994
USG/94/4	Emacs Commands for NEWLIB Users	August 10, 1994
USG/94/5	How to use X-Terminals	August 18, 1994

These publications are available in the self-service area of the computer center and from WWW under the URL `http://info.desy.de/UCO/documentation.html`