Fermi National Accelerator Laboratory

FERMILAB-Conf-91/88

Fermi UNIX™ Environment

Judith Nicholls
Fermi National Accelerator Laboratory
P.O. Box 500
Batavia, Illinois 60510

March 1991

* Presented at Computing and High Energy Physics Conference, KEK, Tsukuba, Japan, March 10-15, 1991.

Fermi UNIX™ Environment

Judith Nicholls
Fermi National Accelerator Laboratory[1]

The introduction of UNIX at Fermilab involves multiple platforms and multiple vendors. Additionally, a single user may have to use more than one platform. This heterogeneity and multiplicity makes it necessary to define a Fermilab environment for UNIX so that as much as possible the systems "look and feel" the same. We describe our environment, including both the commercial products and the local tools used to support it. Other products designed for the UNIX environment are also described.

1. Motivation and Methodology

Based on many experiences in the High Energy Physics Community -- with our central VAX Clusters, HEPVM and our own VM system, DEC workstations, and, recently, our UNIX workstations -- it became clear that we would need to deal with this new player, UNIX, in a very organized way. On VMS and VM we had tried to provide coherence on the system, especially as it related to product support, and to provide and document the tools necessary for the work of the physicists, engineers, and others using the system. In the case of UNIX the job was clearly going to be much harder. Not only would there be hundreds of these systems, but, because of our goal of increased vendor-independence, they would not all be the same. UNIX systems would be used for both workstations and for farm systems. Clearly a user would need to move from UNIX system to UNIX system and these systems should appear the same to them. Some users would need to access a UNIX system from a VMS system; some would want to submit jobs to UNIX farms from VMS systems; users with UNIX workstations would want to access VMS systems.

The ideal computing system would have:

- Coherence on a single system (e.g., product support methodology)
- Consistency between systems (e.g., commands, key functions)
- Connectivity to existing systems (e.g., VMS)
- Consistency with existing environment (e.g., VMS)

We knew we would need some tools, in particular in the area of product support and distribution, but it was not clear how closely we could make these disparate

---

[1]Operated by Universities Research Association, Inc., under contract with the United States Department of Energy

systems "look the same" and how easy the communication between these systems and VMS could be made.

It should be added that our work was no doubt made easier by the fact that there was not a large installed base of UNIX users at Fermilab. We were able to choose that the Fermi UNIX Environment have some of the "look and feel" of VMS rather than the "look and feel" of UNIX. (Of course, any user is free to configure his environment in a more UNIX-like way.)

The first major project for what later became known as the Fermi UNIX Environment was the development of a UNIX product support system, **ups** [1]. In July 1990, we installed the UNIX Development System, targeted for user program development for the UNIX farm systems to come. In connection with this effort, we developed the first draft of the user's guide, *UNIX at Fermilab* [3], and a comprehensive one-day UNIX course. These experiences, plus the frequent UNIX Early Bird Meetings for the users of the Development system, brought to the fore many problems with UNIX in our environment and provided suggestions for addressing them.

We then began to hold cross-departmental meetings in the Computing Division specifically targeted at the "Fermi UNIX Environment." These meetings are primarily for status and discussion, but this group spawns (normally short-lived) working groups when the need arises. The Early Bird meetings, attended by users as well as representatives of all of the departments in the division, prod us on and provide us with course-corrections.

It should be noted that the Fermi UNIX Environment does put requirements on the platforms to be supported. A certain minimum commonality with the currently-supported systems (yet to be defined in detail) and a minimum set of extensions to Fortran are required.

2. Components of Fermi UNIX Environment

The Fermi UNIX Environment is still in the process of being defined and will no doubt encompass more than we describe here. However, at this time, it consists of the following parts:

- Default shell and window system
- Central uid/gid registration
- Definition of commonality in UNIX
- Fermilab Fortran Acquisition standard
- Connectivity
- Product support and distribution tools

- Tools and utilities to make life comfortable
- System management tools
- I/O packages
- Farm implements

The items in this list are really of two types: the first two relate to defining and documenting (and requiring) a commonality and the others relate to "added value."

2.1. Commonality

Providing commonality in UNIX has been approached in several ways: recommending a standard shell and graphical user interface, finding and documenting commonality in commands, and tailoring the environments to make them appear the same.

Fermilab has standardized on the C-shell and uses it in the Fermi UNIX Environment. For graphical user interface, we have standardized on the X11 window system. We are also providing central UNIX user id and group id (uid and gid) registration for the lab.

The definition of the Fermilab Fortran Acquisition Standard [2] assures the users that if they don't go beyond this superset of standard Fortran they will be able to use the UNIX platforms that we purchase.

We decided to provide a user's guide for UNIX [3], as we had done in the past for other platforms, even though excellent books on UNIX are widely available. There were a number of reasons for this. First, we would attempt to document a set of commands and options that worked on all the supported platforms, concentrating on the kinds of operations needed for program development and physics analysis. Second, we would describe how to go about program development and how to run jobs on UNIX, as well as how to do I/O. Third, we would include in this guide additions to the supported platforms, whether they be common UNIX utilities such as telnet or third-party add-ons such as EDT+ -- in other words we would document what we now call the Fermi UNIX Environment.

The tailoring of the environment to make the different systems appear the same was effected through standard cshrc and login files that work on all supported platforms to give the same effect [4]. They not only provide a common UNIX behavior, independent of a particular platform's defaults, they also provide a certain consistency with VMS. These files do such things as make certain products available (such as Berkeley mail and the setup command), define key functionality (such as the delete key and CTRL/C), set path and manpath, and set flags (for example, noclobber and ignoreeof).

Another very important issue for people using the UNIX workstations directly is keypad mapping for the EDT+ editor and the keypad mapping when using a

VMS system from the UNIX workstation. Using the keyboards of the UNIX workstations is currently documented in several technical notes [5].

To summarize, the recommendations for commonality are:

1. C-shell, X11 window system
2. Central uid/gid assignments
3. Fermilab Fortran Acquisition Standard
4. Standard cshrc and login files
5. VMS-style keypad mapping
6. UNIX User's Guide

2.2. Added Value

When considering an "added-value" product, we are keenly aware of budget considerations. We not only have to consider the cost of buying a product on hundreds of platforms at Fermilab, we have to consider whether the home institutions of our experiments will be able to afford them. We are tending to buy "site licenses" for many of these products, and in some cases the "site" includes the homes of employees as well as the home institutions of experiments. When we feel we can support a free or nearly-free product, we will include that as well even though it may overlap with a licensed product (e.g., EDT+ and EMACS).

Connectivity

Connectivity with existing platforms and between UNIX platforms is of vital concern. Fermilab has purchased site licenses for Multinet which provides TCP/IP and rcp protocol capability for VMS platforms. KI Research software provides DECnet capability for UNIX platforms. The result is a very rich set of commands. For login from one platform to another, **set host** or **telnet** or **rlogin**; to transfer files, **ftp** or **rcp**; and **rsh** to execute remote scripts.

Product Support and Distribution

Product support and distribution utilities, **ups** [1] and **upd** [6], have been written in-house.

The Computing Division will use a single product support and distribution methodology for UNIX products. **ups** (UNIX Product Support) is used by product maintainers to install versions of products, and users to list available products and make products available to them. Typically, a system is set up with a /usr/products directory which contains the products registered with **ups**. Some utilities are made available without setup by putting them in /usr/local/bin and system maintenance utilities are in /systools.

upd (UNIX Product Distribution) is used by product maintainers to make products available to users for distribution and by users to list the products available and to obtain copies of products.

We are currently setting up a single point for UNIX product distribution, independent of the product maintainer or the target machine type. Because of difficulty of putting large quantities of disk on our UNIX systems and the availability of disk elsewhere, we will be exporting disk from the VAX/VMS FNAL cluster to the UNIX distribution machine and also to several UNIX development platforms. Users desiring products will make their request to the UNIX distribution machine where the disks will appear to reside. To product developers, the disks will also appear to be on their machines for ease of installing and updating products.

Other Tools and Utilities

Licenses for a number of tools and utilities have been obtained to make life with UNIX more comfortable for Fermilab users. We have a site license for the edt-emulator EDT+ [7] which is available for a number of different UNIX platforms and is very popular with the users. We have also licensed a TPU-emulator (nu/TPU) [8] and a DCL-emulator (VCL) [9] on several platforms but evaluation of these is not complete. In the public domain area, we are providing EMACS on both UNIX and VMS as well as TeX and LaTeX on both platform types.

A number of utilities have been written and others will follow. **Info** is a simple news-like utility which can be used to keep users informed of new products, new releases, meeting announcements and minutes of meetings, etc., related to UNIX and the particular platform involved. **flpr** (Fermilab lpr) is used to print on printers connected to other systems, in particular VMS sytems. These are included in a package named **futil** [4], along with other utilities such as **less** (a public domain paging program) and the **log???** commands (which print out a specific field in the password file for a given user).

cedit[10] (Command EDITor) is a command recall and editing program which supports command editing in the manner of VMS.

funkern [4] is a selection of "kernel" utilities that other products in the Fermi UNIX Environment are allowed to rely on. **funkern** includes **funame** (Fermi uname) and **dropit** (deletes items in a path).

Software Development

It is not clear yet how the experiments will manage software development for UNIX machines. At the request of several experiments, EXPAND [11] has been enhanced to include support for UNIX and has been ported to the support UNIX platforms. This gives the experiments that use EXPAND and DEC's CMS the option to run both EXPAND and CMS on VMS or to use CMS on VMS followed by EXPAND on UNIX.

I/O

Clearly Fortran I/O is a very important topic in a high energy physics environment. Currently, we have a number of efforts underway. We are developing support for reading and writing ASCII labeled tapes, in particular VMS tapes (**rbio**) [12]. We have also provided a set of general tape I/O routines (**bufio**) [13] as well as a set of routines to read VAXONLINE data tapes (**fslib**) [14]. We plan to make extensive use of CERN's **zftp** [15].

Because of UNIX's lack of control on magnetic tape drives, we have developed a tape allocation system. [16]

System Management Tools

A set of local system management utilities that are maintained on all supported UNIX platforms have been developed [4]. **systools** is a set of UNIX system management tools, including user administration scripts, the **cmd** command which is used to grant root permissions for specific commands, and the **addlocal** command for adding or updating files in the /usr/local directory. The adduser script refers to the central uid/gid database for lab-wide consistency, and provides the user account with the default user .cshrc and .login which themselves refer to the fermi.cshrc and fermi.login, respectively.

Farm Implements

The ACP Cooperative Processes Software **cps** [17] is a package of software tools that makes it easy to split a job among a set of processes distributed over one or more computers, for example on a multi-processor farm. **cps_batch** [18] organizes and controls cps batch jobs and provides tools which allow users and system managers to interact with a production system.

3. Documentation and Training

As mentioned before, a UNIX User's Guide is in draft form at this time. The stock room stocks Digital Press' *UNIX for VMS Users* by Philip Bourne which is very helpful to our users. The recommended system management handbook for user system managers is *UNIX System Administration Handbook* by Nemeth, Snyder and Seebass, Prentice Hall. A comprehensive one-day course has been prepared for users and it is our intention to provide in-house overview of UNIX system management. The users are, unfortunately, mostly unwilling to attend a 3 to 5 day commercial course, so we attempt to provide a concentrated overview of the issues important to our users and system managers. An in-house course also allows us to include our local additions and exclude subjects that are not of interest to our users.

4. Issues

We are currently supporting four workstation types: Silicon Graphics, Sun, IBM RS/6000, and DECstation (ULTRIX). Every part of the Fermi UNIX Environment has to be ported to every platform type and made available to each platform type on the distribution platform. We have unfortunately found that we have dependencies not only with platform, but also with operating system releases on the individual platforms.

A distributed computing environment, with more and more interoperability, taxes our operational and monitoring capabilities. Embarrassingly, it is more and more the norm that a user has to call us to tell us that this or that is down. It is difficult also to know where to go for help because it could be some daemon process on any one of several systems, a router, some intermediary system, etc. Our simple and generally informal problem tracking systems are allowing failures to fall through the cracks more and more frequently.

We certainly don't have I/O on UNIX solved, in my opinion, although people are getting their work done, eventually. We are not comfortable with disk files with no record structure. Support for 8mm tapes has been a continuing saga and no end is yet in sight [19].

It is also difficult to forsee how to implement graphical user interfaces when many of our users still use systems from "dumb" terminals. The Fermilab budget does not allow an early solution to this problem. On the inexpensive end, we are currently weighing X terminals against inexpensive diskless workstations.

For that matter, is X a standard? What about products that will run on one platform but not another?

What do we do about free-ware? The users come to us much more frequently on UNIX systems with this great new thing that we really must support, and there really are good things out there. How do we decide what to support? How do we make it clear what we can and cannot do with our limited resources?

5. Acknowledgements

The Fermi UNIX Environment is the result of the work of many people including Farhad Abarghoui, Keith Chadwick, Chuck DeBaun, Matt Fausey, Randy Herber, Chip Kaliher, Paul Lebrun, Kevin Leininger, Bert Lindgren, John Marraffino, Uday Pabrai, Don Petravick, Ruth Pordes, Steve Rojak, Hemant Shah, Jonathan Streets, Roy Thatcher, Lourdu Udumula, Margaret Votava, Matt Wicks. We also want to recognize the "Early Birds" for their patient cooperation, constant prodding and excellent ideas.

References

[1] UNIX Product Support Philosphy & Overview, Fermilab Computing Division, DS0204.
UNIX Product Support Design & Implementation, Fermilab Computing Division, DS0204.
UPS Developers' Guide, Fermilab Computing Division, PN0426.

[2] FORTRAN Compiler Evaluaton of Fermilab Mandatory Requirements, Fermilab Computing Division, TN0057.

[3] UNIX at Fermilab, Fermilab Computing Division, GU0001.

[4] SEED Project System Administration Guide, Fermilab Computing Division, DCD003.

[5] VT100/VT200 Emulation on a SGI Workstation, Fermilab Computing Division, TSGI004.
VT100/VT200 Emulation on a SUN SPARCstation, Fermilab Computing Division, TSUN011.
VT100/VT200 Emulation on an IBM Workstation, Fermilab Computing Division, TIBM001.
IBM 3270 Emulation on a SUN SPARCstation, Fermilab Computing Division, TSUN016.

[6] UNIX Product Distribution User's Guide, Fermilab Computing Division, PN435.

[7] Boston Business Computing, Ltd.

[8] a/Soft Development Inc.

[9] Boston Business Computing, Ltd.

[10] CEDIT, Fermilab Computing Division, PU0116.

[11] EXPAND, Fermilab Computing Division, PU0093.

[12] rbio, Fermilab Computing Division, SU0033.
FORTRAN Raw Buffered I/O Interface: Internals and Setp, Fermilab Computing Division, IU0002.
FORTRAN Raw Buffered I/O Interface: Tape Interface Module, Fermilab Computing Division, IU0003.

[13] buflo, Fermilab Computing Division, SU0032.

[14] fslib, Fermilab Computing Division, PN289.

[15] Part of CERN's CSPACK.

[16] UNIX Tape Device Allocation System, Fermilab Computing Division, IU0001.

[17] ACP Cooperative Processes User's Manual, Fermilab Computing Division, GA0006.

[18] CPS Batch Users Guide and System Managers Guide, draft.

[19] 8mm Tape: Recommendations for Interchange Standards, Fermilab Computing Division, TN0058.