

A Brief History of Unix

This article will give people with no previous UNIX experience some sense of what UNIX is. This article will cover the history of UNIX and an introduction to UNIX.

HISTORY OF UNIX AND CAUSES FOR ITS POPULARITY

Most discussions of UNIX begin with the history of UNIX without explaining why the history of UNIX is important to understanding UNIX. The remainder of this document will describe some strengths and weaknesses of UNIX and attempt to explain why UNIX is becoming popular. All of UNIX's strengths and weaknesses can be directly related to the history of its development, hence a discussion of history is very useful.

UNIX was originally developed at Bell Laboratories as a private research project by a small group of people starting in 1969. This group had experience with a number of different operating systems research efforts in the 1970's. The goals of the group were to design an operating system to satisfy the following objectives:

- Simple and elegant
- Written in a high level language rather than assembly language
- Allow re-use of code

Typical vendor operating systems of the time were extremely large and all written in assembly language. UNIX had a relatively small amount of code written in assembly language (this is called the kernel) and the remaining code for the operating system was written in a high level language called C.

The group worked primarily in the high level language in developing the operating system. As this development continued, small changes were necessary in the kernel and the language to allow the operating system to be completed. Through this evolution the kernel and associated software were extended until a complete operating system was written on top of the kernel in the language C.

UNIX APPLICATION PROGRAMMING INTERFACE

Many proprietary operating systems have a simplified view of application behavior. The typical application reads some data from disk, tape or a terminal and does some processing. Output is produced onto disk, tape, terminal, or printer. The operating systems generally provide easy to use well-implemented facilities to support these types of facilities.

As applications become more sophisticated they need new features such as network access, multi-tasking, and interprocess communications. In traditional operating systems, these features are often hard to use, not well documented, and only callable from assembly language. When a program makes use of these features, the program may be much more complex and much more difficult to maintain.

In UNIX because the C language was written to be used to implement an operating system rather than a traditional "input-processing-output" application, use of these sophisticated features is quite easily done from the C language without writing any assembly language. In addition, the documentation for these sophisticated features is in the same format and location as the documentation for the normal application calls.

When UNIX was distributed, users could write applications in C and easily make use of all of the operating system facilities. This allowed application developers to quickly develop much more sophisticated applications using these facilities.

The pattern of development in UNIX when adding new features such as networking is to provide an application program interface from the C language to access the new features.

In general UNIX system developers and application developers program in the same language using the same application programming interface. In typical proprietary operating systems, the operating systems programmers are programming in assembly language and have access to a many capabilities which are not available to the application developer.

UNIX NETWORKING

In 1984, the University of California at Berkeley released version 4.2BSD which included a complete implementation of the TCP/IP networking protocols. Systems based on this and later BSD releases provided a multi-vendor networking capability based on Ethernet networking.

The networking support included, remote login, file transfer, electronic mail, and other important features.

As UNIX was ported onto more and more different types of computer hardware the UNIX networking allowed many different types of systems to share and mutually use data. Networks consisting of many different systems could be used as a large distributed system.

When SUN Microsystems added NFS (Network File System), this ability to share and mutually use data was significantly enhanced.

UNIX POPULARITY

At this point, the reader might be asking, "This document is designed for first time UNIX users. Why all of this discussion about programming and system programming?" The answer is because the document is using the history of UNIX to explain why UNIX is so popular. The application portability and system programming issues have caused many hardware and software vendors to choose UNIX.

The effect of many vendors choosing UNIX is that there is a wide variety of UNIX systems available to users at attractive prices. There are three primary causes for UNIX's popularity (and none is user interface):

- Only a very small amount of code in UNIX is written in assembly language. This makes it relatively easy for a computer vendor to get UNIX running on their system. UNIX is nearly the unanimous choice of operating system for computer companies started since 1985. The user benefit which results from this is that UNIX runs on a wide variety of computer systems. Many traditional vendors have made UNIX available on their systems in addition to their proprietary operating systems.
- The application program interface allows many different types of applications to be easily implemented under UNIX without writing assembly language. These applications are relatively portable across multiple vendor hardware platforms. Third party software vendors can save costs by supporting a single UNIX version of their software rather than four completely different vendor specific versions requiring four times

the maintenance.

- Vendor-independent networking allows users to easily network multiple systems from many different vendors. These features of UNIX have contributed to its rise in popularity since the mid 1980's

USER INTERFACE

So far, there has been no mention of the user interface for UNIX. UNIX is a good operating system for experienced programmers. The operating system was designed and implemented by experienced programmers so everything which the experienced programmer needs is present but not much else. A perfect example of this is the on-line documentation called "man-pages" or manual pages. The material is completely reference oriented with very little tutorial information. Experienced programmers find the man pages very useful but the beginning user often finds them overwhelming.

In the last few years, there has been extensive work to improve the user interface to UNIX. The most dramatic effort has been the addition of windowing interfaces on top of UNIX such as X-windows, Suntools, NextStep, Motif, OpenLook, etc. These windowing interfaces do not change UNIX itself but are built on top of UNIX to provide a more intuitive interface to UNIX. Each of the different user interfaces has some advantages and some disadvantages. Currently intensive development effort is being done on all of these Graphical User Interfaces (GUIs).

Vendors providing UNIX also have done a work to improve the user interface of their particular versions of UNIX for users without windowing interfaces. Even with all of these efforts, UNIX is weak in the end-user interface area.

USER PORTABILITY

Even with a relatively poor user interface, UNIX has a following of non-programmer users. The primary reason for this is because UNIX runs on so many different computer systems ranging from small desktops to the largest computers in the world. Once a user has learned UNIX, the skills can be used on many different systems. This ability for a user to work on many different makes of computer systems without re-training is called "user portability".

Many users of other operating systems have converted to using UNIX because they felt that UNIX would be the "last" operating system they would have to learn.

OPEN SYSTEMS

There is a recent effort to define what is an "open system" in the international standards area. An open system is a system which allows application portability, system interoperability, and user portability between many different computer vendor hardware platforms.

UNIX is a good example of the advantages to the user having an "open system".

HISTORY SUMMARY

From a simple beginning as a personal research project to an important role in the operating systems on a wide range of computer systems from desktop micros to the largest mainframes, UNIX has and will

have a lot of impact. The strength of UNIX is its portability across multiple vendor hardware platforms, vendor independent networking, and the strength of its application programming interface.

These benefits are so strong that the relative weak end-user interface has not slowed the adoption of UNIX.

The end users are not the direct beneficiaries of the portability and the application program interface. However end-users have already seen the dramatic drop in the cost of computing when multiple vendors can provide the same operating system and software solutions.

End users are currently making the choice for inexpensive and flexible computing rather than best user interface in choosing UNIX.

Source: <http://www.go4expert.com/articles/brief-history-unix-t288/>