ACS-586 Computer System
XENIX Version 2.4c Development System Utilities
Release Notes
January 11, 1984

Installation Procedure

Before installation of the XENIX Development System Utilities, the XENIX Run-time system must be installed. Once this has been done, the XENIX Development System Utilities will be ready for installation. Use the following instructions:

1. Login as "root" <cr>

2. Enter: cd / <cr>

3. For each Utility diskette, follow these steps:

    3.1 load the diskette
    3.2 Enter: tar xv <cr>
    3.3 When you have loaded all of the Utility diskettes, Enter: install <cr>

4. Now load the appropriate "C Compiler" diskette(s). Please read the full set of release notes for an evaluation of the two C compilers shipped with this release.

    4.1 load the "C Compiler" diskette(s)
    4.2 Enter: tar xv <cr>
    4.3 Enter: install

5. This step is optional. If you wish to load the Micro-Soft FORTRAN compiler, follow these steps:

    5.1 load the "MS/FORTRAN" diskette
    5.2 Enter: cd /tmp <cr>
    5.3 Enter: tar xv <cr>
    5.4 Enter: install

6. This step is optional. If you wish to load the Source Code Control System (SCCS), follow these steps:

    6.1 load the "SCCS" diskette
    6.2 Enter: cd / <cr>
    6.3 Enter: tar xv <cr>
    6.4 Enter: install

7. Your XENIX Development System has now been fully installed.

Enhancements to Utilities

1. Included with this release of XENIX are two versions of the C compiler. One version is called a.out (small model) and is essentially the same compiler Altos has always shipped with the ACS 586. The a.out compiler has a limit of 64K bytes of code and 64K bytes of data/stack address space.

   The new version of the C compiler is called x.out (medium model) and supports 192K bytes of code and 64K bytes of data/stack. There are many issues involved with switching to the new compiler that are discussed in a special section of this release note devoted to the x.out C compiler. After reading the release note thoroughly, the user should then decide which C compiler best suites their needs.

   The user should be aware of the fact that the relocation formats for the two compilers are different and not compatible. Many of the programs included with the different compilers are also not compatible. To avoid confusion, Altos recommends that the user install either the a.out or x.out compiler alone on the disk.

2. The XENIX 2.4 release contains new versions of /lib/libc.a and /lib/crt0.o that are used with the a.out (small model) C compiler. This version of the libraries fixes a problem with stack growth in C programs. If a user program uses more than 4K bytes of stack space, the program should be re-compiled using the new versions of these libraries.

3. The XENIX Fortran 77 compiler is no longer available with the XENIX Development System Utilities. In its place is the Micro-Soft Fortran (MS/FORTRAN) compiler. MS/FORTRAN allows the user to have medium model (192K code, 64K data) FORTRAN programs.

4. Two programs have been added to the XENIX development system; /usr/lib/spellin and /usr/lib/spellout. These programs are used with spell(1) and are used to add or take out words from the on-line spell dictionary. See the spell(1) manual page for details.

5. There is a new version of the delta(1) program (part of the SCCS package) included with this release that fixes a stack growth problem. Documentation and manual pages for the SCCS package can be found on the SCCS diskette.

6. There are two versions of the vi(1) editor included with this release. The /bin/vi program is the new version of vi(1) that supports the cursor control keys on the Altos-2 terminal. The old version of vi(1) is called

/bin/vi2.13.

Errata

1. The uucp(1) utility can not transfer more than 5 files in the initial transmission. However, once the link between machines is established, the user can queue any number of files to transfer.

2. The manual entry for the locking(2) system call indicates that the parameter nbytes is an integer. The parameter nbytes should be a long. The correct calling convention is as follows:

```
locking(fdes, locktype, nbytes);

int fdes, locktype;
long nbytes;
```