第 17 部

オペレーティングシステム

第 1 章

はじめに

WIDE プロジェクトでは、1990 年から BSD Unix を生んだカリフォルニア大学バークレイ校の CSRG (Computer Systems Research Group) との共同研究を行ってきた。主たる目的は、プロジェクト内の様々な研究開発のプラットフォームとして BSD Unix を利用していくためである。

BSD Unix は、多くの大学や研究機関において、長い間研究開発のプラットフォームとして利用されきた。もちろん、それは BSD がオペレーティングシステムとして優れた機能を持っていたからであるが、それ以外に、すべてのユーザがシステム全体に関するすべてのソースを持ち、それを自由に改変し、情報交換を行うことが可能であったという点は忘れてはならない。多くのユーザの研究成果は、CSRG によって正式リリースに取り入れられ、BSD 自体を改良するために貢献するという再生産の機構がうまく働いていたのである。

しかし、4.2BSD のリリース後、BSD をサポートするワークステーションが市場に現れてきた。これらは BSD がサポートする VAX-11 よりも安価で高性能であったため、ワークステーションを開発の対象として使うユーザの数が圧倒的に多くなった。また、メーカが独自にサポートした機能にも優れたものがあり、もはや通常の BSD Unix には戻れなくなってしまった部分もある。たとえば Sun Microsystems の NFS (Network File System) などがそれである。しかし、メーカのサポートする Unix にはソースコードは当然付いていない。Unix ワークステーションの普及は、Unix 文化を広めるという意味では成果を上げたが、反面、すべてのユーザがソースコードを共有するという Unix の大きな特徴を失わせてしまったのである。

4.3BSD までの BSD Unix は、DEC 社の VAX-11 を対象にしていた。しかし、CSRG によって Tahoe アーキテクチャと呼ばれる CCI Power シリーズがサポートされ、4.3BSD-Tahoe という名前でリリースされたのを皮切りに、複数のアーキクテチャがサポートされるようになった。

WIDE プロジェクトでは開発ベースとして多くの国産 Unix ワークステーションを利用しているが、これらはすべてメーカサポートのオペレーティングシステム上で動作しているため、ソースコードを参照することはできない。場合によってはメーカの協力によってソースコードを提供してもらえることもあるが、それをベースとして研究の成果を一般に配布することは困難である。そのため、BSD Unix が正式にサポートする機種として国産の Unix ワークステーションを加えることは、これからの開発環境を整備す

る上で大きな意味を持っている。

また、国産の Unix ワークステーションのオペレーティングシステムのバージョンアップは、元となる BSD Unix の新バージョンがリリースされてから 1 年以上経過してからになることが多かった。正式サポートに組み入れることによって、リリースと同時に国内でも利用が可能になり、常に最先端の技術が利用できるという点でも効果が期待できる。

第 2 章

WIDE における BSD Unix のサポート

2.1 4.3BSD-Reno

作業開始当初は Sony 社の NWS-3800 シリーズ上に 4.3BSD-Reno リリースを移植する作業をおこなった。これは NEWS-OS 上のクロス環境で行い、Sony の協力を得て NEWS-OS のコードを流用することで NEWS-3800 シリーズで動作する 4.3BSD-Reno を作成した。

4.3BSD-Reno に関しては 90 年度 **WIDE** プロジェクト報告書 [189] に報告されている。

2.2 4.4BSD

4.3BSD-Reno の実装を基にして 4.4BSD の実現を行う予定であったが、これにはいくつかの問題点があった。

1. 4.4BSD では、仮想記憶システムが Mach のそれに全面的に入れ替えられた [190]。NEWS-OS は、古い 4.3BSD の仮想記憶システムを基に作られているので、仮想記憶に関するほとんどの部分は使用できなくなった。

2. NEWS-OS は MIPS 社のプログラムを基に作られている。そのため、それを 4.4BSD に組み入れるためには、Sony だけではなく MIPS 社の了解を得なければならない。その手続きが難航することが予想された。

このような問題があったため、同じ MIPS の R2000/R3000 の CPU を使用する DECstation (pmax と呼ばれる) のコードを基にして、Sony NEWS 用のコードを全面的に書き換えることにした。pmax のコードは、完全にライセンスフリーのものであり、配布するためにライセンス上の問題が生じる可能性はない。

問題は pmax が、同じ MIPS の CPU を使うとはいうものの、バイトオーダが Little Endian であり、それに対し Sony は Big Endian を使用しているということであった。そこで pmax のコードを基にして、それを Little Endian, Big Endian のどちらでも動作するものにして、できるだけ多くのファイルを pmax と NEWS の間で共有できるよう努力をした。その結果 4.4BSD では、NEWS のコードの機種に依存しない大部分のファイルは pmax のファイルへのシンボリックリンクとして実現することができた。

4.4BSD の作業は、I/O プロセッサを持つ NWS-3800 シリーズではなく、シングル CPU の NWS-3400 シリーズで行った。現在は NWS-3400 シリーズと、同じアーキテクチャを持つ NWS-3200 (ラップトップ機) の上で動作している。機種異存の部分や、デバイスドライバなどには、Sony から提供されたコードを利用している。

現在は 4.4BSD 上で vmunix や、コマンド類を再構築することが可能である。コンパイラには GNU cc (バージョン 2.3.3) を使用している。ただし、アセンブラとローダは NEWS-OS に付属のものを利用している。これはオブジェクトフォーマットに MIPS の ECOFF フォーマットを利用しているためである。ECOFF フォーマットを用いていることにより、BSD a.out フォーマットを用いる他のアーキテクチャとのコードの共有性がいくぶん落ちている。今後、GCC と GNU アセンブラ (GAS) の新バージョンへの移行を行うと共に、BSD a.out フォーマットへの対応を行うことで、この問題は解決される予定である。新しい GAS は、 ECOFF フォーマットもサポートするようになったため、ECOFF フォーマットを利用しながら NEWS-OS のコンパイラキットへの依存性をなくすことも可能である。

WIDE プロジェクトによる作業の成果は、1992 年 7 月にリリースされた 4.4BSD のプレリリースである 4.4BSD-Encumbered に含まれている。また、4.4BSD の最終リリースは、1993 年 5 月現在、最終フリーズを行おうとしている段階にある。これは 1993 年夏以前にはリリースされることになるだろう。

なお、4.4BSD-Encumbered および 4.4BSD では、日本の国産ワークステーションとして Sony NEWS だけではなく、OMRON の Luna シリーズもサポートされる。

第 3 章

4.4BSD の概要

ここでは、4.4BSD がどのような特徴と機能を持つかということについて説明する。4.4BSD の最終リリースはまだ行われていないため、1993 年 5 月現在の時点での情報である。4.4BSD に関する解説は文献 [191] からも得られる。

3.1 4.3BSD 以降の BSD Unix の歴史

BSD Unix に由来するオペレーティングシステムを持つ多くのワークステーションは、4.3BSD をそのベースとしている。その後も、いくつかの BSD Unix がリリースされているが、市販のシステムにはあまり反映されていないため、その詳細について知る機会は意外に少ない。まず、4.3BSD 以降に、どのようなリリースがされたかを簡単に紹介する。

1986 4.3BSD

1988 4.3BSD-Tahoe それまでの VAX に加えて Tahoe アーキテクチャ [1] をサポートしたリリース。オペレーティングシステムの大部分は 4.3BSD から変化していないが、ネットワークコードに関しては改良が施されている。

1988 BSD Networking Software 4.3BSD-Tahoe から、ライセンスフリーの部分のみを抜き出したものが BSD Networking Software という名前でリリースされた。

1990 4.3BSD-Reno NFS や OSI などの新機能の追加、HP や 386 などの新アーキテクチャのサポートなど、大幅な変更を施されたバージョンである [189][192]。機能的には 4.4BSD と呼んでもいいほどの大きな変更が加えられているが、4.4BSD で実現すべきいくつかの大きな機能が含まれていないため 4.3BSD-Reno という名前で呼ばれた。以下のような特徴がある。

1. DEC VAX 8600/8650, VAX 8200/8250, MicroVAX II, III サポート
2. HP, i386 サポート
3. ISO/OSI ネットワークプロトコル

[1] CCI Power 6/32, 6/32SX の開発コード名。

4. VFS 仮想ファイルシステム
5. NFS ネットワークファイルシステム
6. MFS メモリファイルシステム
7. FFFS (Fat Fast File System)
8. Kerberos 認証システム
9. カーネル内汎用メモリアロケータ
10. tsleep
11. IEEE 1003.1 system interface
12. TCP/IP
    - Header Prediction
    - CSLIP
    - 階層的経路制御
13. NEW QUOTA

1991 BSD Networking Software II (NET/2) 1988 年の BSD Networking Software と同様に、ライセンスフリーの部分だけを抜き出したリリースであるが、仮想記憶システムを完全に Mach オペレーティングシステムのものに置き換えるなど、大幅な機能変更が施されている。

AT&T のライセンスに抵触するために、カーネル内のコードに一部省略された部分がある未完成なリリースであったが、ライセンスに縛られずに自由に配布できるということから、CSRG 以外で NET/2 を改造して使用可能なオペレーティングシステムとしてサポートする活動が始まった。この中には、次のようなものが含まれる。

- Berkeley Software Design, Inc. の BSD/386
- William Jolitz による 386BSD
- Mach の BSD サーバである BNR2SS
- NetBSD

1992 4.4BSD-Encumbered 4.4BSD のアルファリリースである。当初の計画では、少なく ともカーネルの部分は AT&T (USL) のライセンスからフリーに なる予定であったが、4.3BSD などと同様に 32V 以上のライセンスを必要とする。

リリース時のアナウンスでは、4.4BSD-Encumbered からフリー部分を取り出したものを 4.4BSD-Lite という名前で NET/2 と同様の扱いでリリースする予定であったが、これは未だ果たされていない。

1993 4.4BSD 4.4BSD の最終リリースは、1993 年の夏頃になりそうな見通しである。4.4BSD-Encumbered に比べて、大きな機能変更はほとんどない。CSRG は、4.4BSD のリリースの後解散することを宣言しているので、これが最後の BSD Unix になる可能性は非常に高い。

4.4BSD-Encumbered および 4.4BSD では、以下のような機能が追加されている。

- NEWS, Luna, DECstation, SPARC サポート
- LFS (Log-Structured File System)
- Stackable Filesystem
- 64 ビットファイルサイズ (off_t)
- 32 ビットデバイスタイプ (dev_t)
- IEEE 1003.2 shell and application utility interface
- 新 config 機構

これらについては、この後の部分で詳しく説明する。

3.2　サポートアーキテクチャ

4.3BSD の対象だった VAX と Tahoe は、新しい仮想記憶システムが採用された時点でサポートが打ち切られ、代わって HP 9000/300 シリーズが CSRG の主力開発機種となった。NET/2 では、これ以外に i386 が加わっている。

4.4BSD では、さらにいくつかの機種が加わり、次のようなアーキテクチャをサポートしている。

- HP 9000/300 68000-based workstations
  - 68030 と 68040 の CPU をサポート
- Intel 386/486-based machines (ISA/AT or EISA bus のみ)
  - コードは入っているが、実際には動作しない
- Sony NEWS MIPS-based workstations
  - MIPS R3000 (big-endian), シングルプロセッサマシン
- Omron LUNA 68000-based workstations
  - Luna-I と Luna-II がサポートされる予定
- DECstation 3100/5000 MIPS-based workstation

  - MIPS R2000/R3000 (little-endian)

- SPARCstation I & II SPARC-based workstations

3.3 4.4BSD に含まれるコマンド類

4.4BSD に含まれる、主なコマンド類の一覧を挙げる。4.4BSD では、/usr/src の下のサブディレクトリの下には、各コマンドが 1 つのディレクトリとして収められるようになっている。したがって、各ディレクトリの下のリストによって、そこにどのようなコマンドが置かれているかを知ることができる。

3.3.1 /sbin

```
Makefile        fastboot/       mount_lofs/     nfsiod/         shutdown/
Makefile.inc    fsck/           mount_nfs/      nologin/        slattach/
XNSrouted/      fsdb/           mount_null/     ping/           startslip/
badsect/        icheck/         mount_portal/   quotacheck/     swapon/
clri/           ifconfig/       mount_umap/     reboot/         tunefs/
disklabel/      init/           mountd/         restore/        umount/
dmesg/          mknod/          ncheck/         route/
dump/           mount/          newfs/          routed/
dumpfs/         mount_fdesc/    newlfs/         savecore/
dumplfs/        mount_kernfs/   nfsd/           scsiformat/
```

3.3.2 usr.sbin

```
Makefile        config/         kgmon/          pwd_mkdb/       sysctl/
Makefile.inc    config.new/     kvm_mkdb/       quot/           syslogd/
ac/             cron/           lpr/            quotaon/        timed/
accton/         dev_mkdb/       mkproto/        repquota/       traceroute/
amd/            diskpart/       mtree/          rmt/            trpt/
arp/            edquota/        named/          rwhod/          trsp/
bad144/         eeprom/         nvi.recover/    sa/             update/
chown/          inetd/          portmap/        sendmail/       vipw/
chroot/         iostat/         pstat/          sliplogin/
```

3.3.3 bin

```
Makefile        date/           hostname/       pax/            rmdir/
Makefile.inc    dd/             kill/           ps/             sh/
cat/            df/             ln/             pwd/            sleep/
chmod/          echo/           ls/             rcp/            stty/
cp/             ed/             mkdir/          rm/             sync/
csh/            expr/           mv/             rmail/          test/
```

3.3.4 usr.bin

```
Makefile        expand/         logname/        rev/            tsort/
Makefile.inc    f77/            look/           rlogin/         tty/
apply/          false/          lorder/         rs/             ul/
ar/             file/           m4/             rsh/            uname/
at/             find/           mail/           ruptime/        unexpand/
banner/         finger/         make/           rwho/           unifdef/
basename/       fmt/            man/            sccs/           uniq/
bc/             fold/           mesg/           script/         units/
bdes/           fpr/            mkdep/          sed/            unvis/
biff/           from/           mkfifo/         shar/           users/
cal/            fsplit/         mklocale/       showmount/      uucp/
calendar/       fstat/          mkstr/          size/           uudecode/
cap_mkdb/       ftp/            more/           soelim/         uuencode/
checknr/        gcore/          msgs/           sort/           vacation/
chflags/        gprof/          mt/             spell/          vgrind/
chpass/         graph/          netstat/        spline/         vis/
cksum/          grep/           nfsstat/        split/          vmstat/
cmp/            head/           nice/           strings/        vmstat.sparc/
col/            hexdump/        nm/             strip/          w/
colcrt/         id/             nohup/          struct/         wall/
colrm/          indent/         nvi/            su/             wc/
column/         join/           pagesize/       systat/         what/
comm/           jot/            pascal/         tail/           whereis/
compress/       kdump/          passwd/         talk/           who/
cpp/            ktrace/         paste/          tcopy/          whois/
ctags/          lam/            patch/          tee/            window/
cut/            last/           plot/           telnet/         write/
dc/             lastcomm/       pr/             tftp/           xargs/
deroff/         ld/             printenv/       time/           xinstall/
diction/        learn/          printf/         tip/            xsend/
diff/           leave/          ptx/            tn3270/         xstr/
dirname/        lex/            quota/          touch/          yacc/
du/             locate/         ranlib/         tput/           yes/
env/            lock/           ratfor/         tr/
error/          logger/         rdist/          true/
ex/             login/          renice/         tset/
```

3.3.5 lib

```
lib:
Makefile        libcurses/      libmp/          libtelnet/
csu/            libedit/        libplot/        libterm/
libc/           libkvm/         libresolv/      libutil/
libcompat/      libm/           librpc/         liby/
```

```
lib/libc:
Makefile     gmon/       luna68k@    quad/       stdlib/     vax/
compat-43/   hp300/      mips/       regex/      string/
db/          i386/       net/        sparc/      sys/
gen/         locale/     obj@        stdio/      tahoe/
```

3.3.6 libexec

```
Makefile         fingerd/        kpasswdd/        rbootd/          talkd/
Makefile.inc     ftpd/           lfs_cleanerd/    rexecd/          telnetd/
bugfiler/        getNAME/        mail.local/      rlogind/         tftpd/
comsat/          getty/          makekey/         rshd/            uucpd/
```

3.4 バージョンによるコマンドの比較

4.3BSD-Reno, NET/2, 4.4BSD-Encumbered, 4.4BSD に含まれるコマンドを比較し、コマンドやシステムがそれぞれのリリースの中で、どのソースディレクトリに含まれるかを一覧で示す。

この表から、捨て去られたコマンドや、新しくリリースに組み入れらたコマンドの様子を見ることができる。4.3BSD-Reno の項目だけに pgrm というディレクトリが存在するが、これは、Reno では言語処理系を prgm というディレクトリに集めていたためである。4.4BSD では、このディレクトリは無くなって元の構造に戻ったため prgm に属するコマンドは無くなっている。

このリストの中には、すべてのリリースを通じて共通のコマンドは含まれていない。また、NET/2 以外の 3 つのリリースで old ディレクトリにあるコマンドも省いた。4.4BSD-Encumbered あるいは 4.4BSD で初めて導入されたコマンドはイタリック体で表示されている。

Command	Reno	NET/2	Alpha	4.4BSD
ac	usr.sbin		usr.sbin	usr.sbin
adb	bin		bin	old
adventure	games		games	games
apply	usr.bin		usr.bin	usr.bin
ar	pgrm	usr.bin	usr.bin	usr.bin
arff	usr.sbin		usr.sbin	old
as.hpux			old	old
as.tahoe	pgrm		old	old
as.vax	pgrm		old	old
at	usr.bin		usr.bin	usr.bin
awk	usr.bin		old,usr.bin	old
bc	usr.bin		usr.bin	usr.bin
bdes			usr.bin	usr.bin
boggle	games		games	games
cap_mkdb				usr.bin
cassette	sys			
cc	pgrm			
chflags			usr.bin	usr.bin
ching	games		games	games
cksum		usr.bin	usr.bin	usr.bin
config.new				usr.sbin
consolerl	sys			
cpio	usr.bin	usr.bin	usr.bin	old
cpp	pgrm		old	old,usr.bin
cron	usr.sbin		usr.sbin	usr.sbin
crypt			old	old
ctags	pgrm	usr.bin	usr.bin	usr.bin
dbconv			old	old
dbx	pgrm		old	old
dc	usr.bin		usr.bin	usr.bin
dd	bin		bin	bin
delivermail	libexec			
deroff	usr.bin		usr.bin	usr.bin
des	kerberosIV		kerberosIV	kerberosIV
dev			sys	sys
dev_mkdb		usr.sbin	usr.sbin	usr.sbin
diction	usr.bin		usr.bin	usr.bin
diff	usr.bin		usr.bin	usr.bin
dlmpcc	usr.sbin	usr.sbin	usr.sbin	old
dumplfs			sbin	sbin
dungeon	games		games	
ed	bin		bin	bin,old
eeprom				usr.sbin
enpload	sbin	sbin	sbin	old
eqn	usr.bin		old	old
error	pgrm	usr.bin	usr.bin	usr.bin
etc.i386		etc	etc	etc
etc.luna68k				etc
etc.pmax			etc	etc
ex	usr.bin		usr.bin	usr.bin
expr	bin		bin	bin
f77	pgrm		usr.bin	usr.bin

Command	Reno	NET/2	Alpha	4.4BSD
file	usr.bin		usr.bin	usr.bin
flcopy	usr.sbin	usr.sbin	usr.sbin	old
floppy	sys			
fpr	pgrm	usr.bin	usr.bin	usr.bin
fsdb	sbin		sbin	sbin
fsplit	pgrm	usr.bin	usr.bin	usr.bin
g++		usr.bin	usr.bin	
gas		usr.bin	usr.bin	
gcc	libexec	usr.bin	usr.bin	
gcore	pgrm	usr.bin	usr.bin	usr.bin
gdb		usr.bin	usr.bin	
gettable	usr.sbin	usr.sbin	usr.sbin	old
gprof	pgrm	usr.bin	usr.bin	usr.bin
graph	usr.bin		usr.bin	usr.bin
grep	usr.bin		usr.bin	usr.bin
groff		usr.bin	usr.bin	
groups	usr.bin	usr.bin	old	old
halt	sbin	sbin		
htable	usr.sbin	usr.sbin	usr.sbin	old
hunt	games			
icheck	sbin		sbin	sbin
id		usr.bin	usr.bin	usr.bin
implog	usr.sbin	usr.sbin	usr.sbin	old
implogd	usr.sbin	usr.sbin	usr.sbin	old
indent	pgrm	usr.bin	usr.bin	usr.bin
init	sbin		old,sbin	old,sbin
join	usr.bin		usr.bin	usr.bin
jot				usr.bin
kdb	kerberosIV,sys	kerberosIV	kerberosIV	kerberosIV
kdump				usr.bin
kerberos	etc,kerberosIV	kerberosIV	kerberosIV	kerberosIV
lam				usr.bin
ld	pgrm	usr.bin	old,usr.bin	old,usr.bin
learn	usr.bin		usr.bin	usr.bin
lex	old,pgrm	usr.bin	old,usr.bin	old,usr.bin
lfs_cleanerd				libexec
libF77	lib			
libI77	lib			
libU77	lib			
libcompat	lib		lib	lib
libdbm	lib,old		lib,old	old
libedit			lib	lib
libg++		lib	lib	
libkern			sys	sys
libkvm			lib	lib
libm	lib,old	lib	lib,old	lib,old
libmp	lib		lib	lib
libndbm				old
libpc	lib			
libplot	lib		lib	lib
libtelnet		lib	lib	lib
lint	pgrm			

Command	Reno	NET/2	Alpha	4.4BSD
lisp	pgrm	usr.bin	usr.bin	old
ln	usr.bin	bin	bin	bin
logname		usr.bin	usr.bin	usr.bin
look	usr.bin		usr.bin	usr.bin
lorder	pgrm	usr.bin	usr.bin	usr.bin
m4	pgrm	usr.bin	usr.bin	usr.bin
machine	usr.bin	usr.bin	usr.bin	old
mail.local		libexec	libexec	libexec
make	old,usr.bin	usr.bin	old,usr.bin	old,usr.bin
man	old,usr.bin	usr.bin	old,usr.bin	usr.bin
mdec	sys			
miscfs			sys	sys
mkdep	pgrm	usr.bin	usr.bin	usr.bin
mkhosts	usr.sbin		old	old
mklocale				usr.bin
mkpasswd	usr.sbin		old	old
mkproto	usr.sbin		usr.sbin	usr.sbin
mkstr	pgrm	usr.bin	usr.bin	usr.bin
more	usr.bin	usr.bin	old,usr.bin	old,usr.bin
mount_fdesc			sbin	sbin
mount_kernfs			sbin	sbin
mount_lofs			sbin	sbin
mount_nfs			sbin	sbin
mount_null			sbin	sbin
mount_portal			sbin	sbin
mount_umap			sbin	sbin
ms	share		old	old
ncheck	sbin		sbin	sbin
netccitt		sys	sys	sys
netrmp	sys	sys		
newlfs			sbin	sbin
nm	pgrm	usr.bin	usr.bin	usr.bin
nologin			sbin	sbin
np100				old
nvi				usr.bin
nvi.recover				usr.sbin
pascal	pgrm	usr.bin	usr.bin	usr.bin
patch				usr.bin
pax				bin
pcc	libexec		old	old
phantasia	games		games	games
pig				games
plot	usr.bin		usr.bin	usr.bin
portmap		usr.sbin	usr.sbin	usr.sbin
pr	usr.bin		usr.bin	old,usr.bin
pstat	usr.sbin		usr.sbin	usr.sbin
ptx	usr.bin		usr.bin	usr.bin
pwd_mkdb		usr.sbin	usr.sbin	usr.sbin
quiz	games		games	games
quot	usr.sbin		usr.sbin	usr.sbin
ranlib	pgrm	usr.bin	usr.bin	usr.bin
ratfor	pgrm		usr.bin	usr.bin

Command	Reno	NET/2	Alpha	4.4BSD
rbootd			libexec	libexec
refer	usr.bin		old	old
rev	old		usr.bin	usr.bin
roff	usr.bin		old	old
rogue	games,old	games	games,old	games,old
rs				usr.bin
rxformat	usr.sbin	usr.sbin	usr.sbin	old
sa	usr.sbin		usr.sbin	usr.sbin
scripts				sys
scsiformat			sbin	sbin
sed	usr.bin		usr.bin	old,usr.bin
sh	bin	bin	bin,old	bin,old
size	pgrm	usr.bin	usr.bin	usr.bin
sort	usr.bin		usr.bin	old,usr.bin
spell	usr.bin		usr.bin	usr.bin
spline	usr.bin		usr.bin	usr.bin
startslip			sbin	sbin
strings	pgrm	usr.bin	usr.bin	usr.bin
strip	pgrm	usr.bin	usr.bin	usr.bin
struct	pgrm		usr.bin	usr.bin
sum	usr.bin		old	old
symorder	pgrm	usr.bin	old	old
sysctl				usr.sbin
sysline	usr.bin		old	old
systat	usr.bin		usr.bin	usr.bin
tail	usr.bin		usr.bin	usr.bin
talk	old,usr.bin	usr.bin	old,usr.bin	old,usr.bin
tar	bin		bin	old
tbl	usr.bin		old	old
test	bin		bin,old	bin,old
tetris				games
tmac	share	share	old,share	old,share
tsort	pgrm	usr.bin	usr.bin	usr.bin
uname				usr.bin
unifdef	pgrm	usr.bin	usr.bin	usr.bin
units	usr.bin		usr.bin	usr.bin
uucp	usr.bin		usr.bin	usr.bin
vfilters			old	old
vgrind	pgrm	usr.bin	usr.bin	usr.bin
vipw	sbin	usr.sbin	usr.sbin	usr.sbin
vm		sys	sys	sys
vmstat.sparc				usr.bin
warp	games		games	
whereis	usr.bin	usr.bin	old	old,usr.bin
which	usr.bin	usr.bin	old	old
whoami	usr.bin	usr.bin	old	old
xinstall	pgrm	usr.bin	usr.bin	usr.bin
xneko			games	games
xroach				games
xsend			usr.bin	usr.bin
xstr	pgrm	usr.bin	usr.bin	usr.bin
yacc	old,pgrm	usr.bin	old,usr.bin	old,usr.bin

3.5　64 ビットファイルサイズ

従来の Unix では、ファイル内のオフセットを表すためのデータ構造 (off_t) には、32 ビット符号付き整数が用いられてきた。しかし、昨今のハードディスク技術の進歩により、32 ビット符号付き整数で表すことができる最大値以上の容量を持つハードディスクも現れてきた。そのため、4.4BSD では off_t として 64 ビット符号付き整数 (quad_t) を使用するように変更された。

現在のコードでは off_t は実際には long long 型であり、これをサポートするコンパイラでしか 4.4BSD のシステムを構築することはできない。ただし、long long 型が直接使用されているわけではなく、64 ビット整数を表す型としては quoad_t が使用されている。コンパイラによる違いは、quoad_t を指定する部分で吸収できる。

ファイル内のオフセットを指定するためのデータ構造が変化したことにより、いくつかのシステムコールがその影響を受ける。たとえば、lseek システムコールは次のような呼びだし形式を持つ。

```
off_t lseek(int fdes, off_t off, int sbase)
```

この呼びだし形式は変わらないが、off_t が表すデータの大きさが変わるので、不用意に long 型などを用いているソースコードは、再コンパイルにより動作しなくなってしまう。ただし、バイナリ互換性は保たれているので、4.3BSD のシステムで作成されたバイナリコードは、そのまま 4.4BSD 上で動作することが出来る。

また、以下のようなデータ構造が影響を受ける。

- struct flock (l_start, l_len)
- struct file (f_offset)
- struct stat (st_size)
- struct uio (uio_offset)

64 ビット整数が導入されたことにより、その影響受けないはずの変数にも影響が現れる。それは、gcc 2.0 では 64 ビット変数は 64 ビットのデータ境界に置かれるようになっているからである。構造体のメンバの中に 64 ビット変数があると、その変数が 64 ビット境界に置くように配置される。これにより大きさがあらかじめ決まっている構造体のサイズが思ったものと異なってしまうからである。多くの構造体のメンバの再配置が必要とされた。

3.6　仮想記憶システム

4.3BSD の仮想記憶システムは 1979 年頃に設計されたものである。当時、メモリのコストはハードディスクの約 1000 倍であり、仮想記憶機構の主な目的は、限られたメモリ

を使っていかに広い仮想空間を提供できるかということにあった。メモリのコストが下がり、当時と比べものにならないほどのメモリを搭載することが一般的になってくると、ハードウェアの技術レベルと仮想記憶システムの間のギャップが表面化してきた。このため、BSD Unix 仮想記憶システムの見直しの必要性が以前から指摘されてきた。

4.4BSD の仮想記憶システムを実現するうえで、解決すべき課題であると考えられていたのは次のような点である。

- 豊富にあるメモリを効率よく使用する
- スワップ空間より大きなメモリをサポート
- 高い移植性
- 広大で、離散的な空間をサポート
- バッファキャッシュとメモリプールの統合

実装はユタ大学によって行われた。CMU の Mach オペレーティングシステムの仮想記憶管理システムを全面的に採用し、それに BSD Unix の仮想記憶インタフェースの皮を被せるような形で実現されている。

従来の Unix では、プログラムの仮想空間は text, data, bss, stack の 4 つの領域から成り立っていた。しかし、新しい仮想記憶システムにより任意の数の領域を持つことができるようになった。これを使って、離散的なアドレス空間や、プロセス間での共有メモリを実現することができる。

3.7 ファイルシステム

4.3BSD では、ハードディスクを利用した通常のローカルファイルシステムしかサポートされていなかったが、NET/2 で NFS (Network File System) をサポートするために VFS (Virtual File System) のインタフェースが取り入れられ、複数の種類のファイルシステムをサポートすることが可能になった。

4.4BSD では、さらにスタッカブルファイルシステム (stackable file system) が採用され、ファイルシステムを階層的に積み重ね、あるファイルシステムの機能を使って、別の種類のファイルシステムを実現することが可能になった。この機構により、新しいファイルシステムの開発が容易になり、いくつもの種類のファイルシステムがサポートされている。

3.7.1 MFS

MFS (Memory based file system) は 4.3BSD-Reno からサポートされているファイルシステムである。単なるメモリディスクとは異なり、仮想空間上にファイルシステムを実

現する [193]。このため、メモリの量に制限されずにファイルシステムの大きさを設定することが可能で、メモリが不足してくると、アクセスされていない部分はスワップデバイスに書き出される。MFS はテンポラリディレクトリ (/tmp) に対して利用されることが多く、短時間しか存在しないテンポラリファイルはメモリ上作成されすぐに消去されるのに対し、長時間存在するファイルはスワップデバイスに待避され、メモリを圧迫しないという特長を持つ。

MFS はソースコードで 560 行程度と非常に単純な機構で実現されている。MFS の作成は newfs コマンドによって行い、この newfs コマンドは実行を終了せずデーモンとして実際の入出力の処理を行う。入出力を行うプロセスは、デーモンとして待機している newfs を wakeup することで、デーモンに対して処理を依頼する。

現在の MFS は、メモリとのデータ転送と比較して約 45%程度の速度を実現している。メモリに比べ約半分の速度しか実現されていないのは、ユーザ空間からカーネルバッファと、カーネルバッファから newfs が管理する空間への 2 回のメモリ間コピーが必要とされていることによる。将来的には、1 度のコピーで実現することを目標としている。

3.7.2 NQNFS

NFS (Network File System) もまた、4.3BSD-Reno からサポートされた。Sun Microsystems 社の NFS と互換性があるが、独自の拡張機能も持っているため NQNFS (Not Quite NFS) と呼ばれる [194]。実装はカナダ Guelph 大学の Rick Macklem が行い、NFS の仕様を元にして独自に実装された。RPC/XDR の部分については Sun の提供するコードを利用しているが、カーネル内では RPC/XDR のメカニズムは使用せず、直接データ構造を操作する構造になっている。

4.4BSD では、4.3BSD-Reno や NET/2 にはない Lease というキャッシュ管理の機構が実装されている。Lease というのは、サーバがクライアントに対して Lease と呼ばれる短時間だけ有効なアクセス権を発行するメカニズムである [195]。あるクライアントに Lease を与えている最中に、別のクライアントからそのファイルに対するアクセスがあった場合、サーバはただちにその要求に答えるのではなく、Lease を持っているクライアント (Lease Holder) に対して、キャッシュを書き出す要求を出し、その応答を受け取ってから、2 番目のクライアントに対してデータを送る。Lease Holder が応答しなければ、Lease の有効期限が終り、その Lease は効力を無くしてしまうので、その後で 2 番目のクライアントの要求に答えることが出来る。このように、Lease はステートレスな NFS のプロトコル上でも有効に動作するキャッシュ管理のメカニズムである。

3.7.3 Stackable Filesystem

Stackabel Filesystem は、UCLA の Ficus プロジェクトの中の活動として、John S. Heidemann によって実現された機能である [196][197]。この目的は次のような点である。

- システムから独立したファイルシステム

- 再利用性の向上
- 開発の容易性
- 拡張性の向上

/sys/miscfs の下には、次のようなサブディレクトリが存在する。

deadfs umount されたファイルが属する

fdesc プロセスのファイルディスクリプタをアクセスする

fifofs FIFO を実現する

kernfs カーネルに関する情報

lofs ループバックファイルシステム

nullfs Stackable Filesystem のスケルトン

portal Portal Filesystem

specfs スペシャルファイル

umapfs uid のマッピングを行う Stackable Filesystem のサンプル

3.8 LFS

LFS (Log-Structured File System) は、UCB Sprite プロジェクトの成果であり、Mendel Rosenblum, John K. Ousterhout 等によって実装された。

LFS は、現在のファイルシステムが持つ次のような問題点を解決するために考案されたファイルシステムである。

- ディスクのスループットは向上してもアクセス速度は変わらない
- read の効率はキャッシュにより解決されるので write が問題となる
- FFS では小さなファイルに書き込みを行う時には、ディスクのバンド幅の 5% 以下しか利用されなていない

LFS は、以上のような問題点を解決するために、ディスクアクセスのためのヘッドのシークを極力排除するという方針をとっている。そのため、ディスクに書き出すべきデータは連続したブロックに更新情報として書き込まれていく。これが Log-Structured と呼ばれる所以である。Unix では、小さいファイルの作成と消去を繰り返す処理が非常に多い。この処理は inode データの更新を伴うため、物理的に離れた場所にあるデータを更新しなければならず、ディスクヘッドのオーバーヘッドが大きなコストを持ってくる。LFS は主に、小さなファイルの作成とデータの書き込みに焦点を当てたファイルシステムである。

また、ディスクの中で最近更新したデータが局所化されているので、システムがクラッシュした際の復旧の点からも有利である。定期的にチェックポイント情報を更新し、クラッシュした際には、最後のチェックポイントより後に更新されたデータだけを検査すればよい。従来のファイルシステムでは、ファイルシステム全体を走査して整合性の確認をしなければならなかったの対して、検査するデータ量が非常に少なくなっている。

3.9　残された課題

4.4BSD の最終リリースにも実現することのできなかった大きな機能がいくつか残っている。

3.9.1　BStreams

AT&T ベル研究所の 8th Edition Unix の Streams の考えと、BSD の socket を統合して BStreams と呼ぶ機構を実装する予定であったが、4.4BSD には未サポートのままリリースされる。

3.9.2　ネットワークコードの高速化

LBL (Lawrence Berkeley Laboratory) の Van Jaconson によって、TCP/IP のネットワークコードの大幅な改良が試みられている。実装上の階層をできる限り排除することと、ネットワークハードウェアからの割り込み処理がなるべく起こらないようにするというのが基本的なアプローチである。大幅な構造の改造を伴う変更であり、現時点では 4.4BSD には取り入れられない。

3.9.3　バッファキャッシュと仮想記憶の統合

現在はファイルシステムのバッファキャッシュと仮想記憶システムでは、別々のメモリプールを管理している。この 2 つを統合して、互いのメモリを単一のインタフェースで管理するという計画が以前から存在したが、これも 4.4BSD では実現されていない。