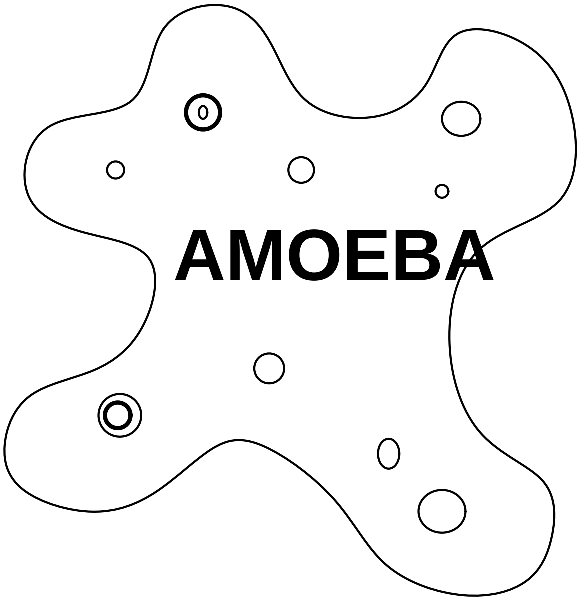
The Amoeba Reference Manual

Release Information For Amoeba 5.3



Amoeba is a registered trademark of the Vrije Universiteit in some countries.

AMOEBA is a trademark of the Vrije Universiteit.
Sun-3, Sun-4, NFS, SPARCclassic, SPARCstation, MicroSPARC, SunOS and Solaris® are trademarks of Sun Microsystems, Inc.
SPARC is a registered trademark of SPARC International, Inc.
UNIX is a trademark of Unix International.
Irix and Personal Iris are trademarks of Silicon Graphics Corporation.
DEC, VAX, MicroVAX, Ultrix and DEQNA are registered trademarks of Digital Equipment Corporation.
Intel 386, Pentium, Multibus and Intel are trademarks of Intel Corporation.
Ethernet is a registered trademark of Xerox Corporation.
IBM and PC AT are registered trademarks of International Business Machines Corporation.
X Window System is a trademark of the Massachusetts Institute of Technology.
WD, EtherCard PLUS and EtherCard PLUS16 are trademarks of Standard Microsystems Corporation.
Logitech is a trademark of Logitech, Inc.
Novell is a trademark of Novell, Inc.
3Com and Etherlink II are registered trademarks of 3Com Corporation.
MS-DOS is a trademark of Microsoft Corporation.

Amoeba includes software developed by the University of California, Berkeley and its contributors.

Contents

1 Copyright and Licence Information

The files distributed with the Amoeba Distributed Operating System are all subject to copyright. All files are subject to the following copyright unless some other copyright notice is included in the file.

© copyright 1989,1990,1991,1992,1993,1994,1996 Vrije Universiteit and Stichting Mathematisch Centrum

Permission to use, sell, duplicate or disclose this software must be obtained in writing from:

Dr. Andrew S. Tanenbaum
Faculteit Wiskunde en Informatica
Vrije Universiteit
De Boelelaan 1081a
1081 HV Amsterdam

Files which are copyright the Vrije Universiteit that come from ACK (The Amsterdam Compiler Kit) are subject to the following copyright.

© copyright 1987,1990,1991,1992 by the Vrije Universiteit, Amsterdam, The Netherlands.

Permission to use, sell, duplicate or disclose this software must be obtained in writing. Requests for such permissions may be sent to

Dr. Andrew S. Tanenbaum
Faculteit Wiskunde en Informatica
Vrije Universiteit
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands

Any other copyright restrictions are as described in individual files.

2 Change Notes

Below is the information about upgrading between various releases of Amoeba, and the changes between the the past few releases of Amoeba. Following this are possible changes for future releases.

Warning

> Both SunOS 4 and SunOS 5 (Solaris 2) are supported in Amoeba 5.3. This is the last release of Amoeba that will support the UNIX FLIP driver in SunOS 4. The compilers and other tools of SunOS 4 will also cease to be supported in future releases.
>
> This is the last release in which the Sun 3 Amoeba kernels will be supported. The Amakefiles may continue to be available in future releases but they probably will not have been tested.

Upgrade Information

Upgrading to Amoeba 5.3 from versions lower than 5.2 may not work and is not recommended. It is possible to upgrade from Amoeba 5.2 to Amoeba 5.3. The upgrade is not simple and a special package has been provided with the distribution. This package contains a script to update the directory graph structure and a `README` file with the instructions about how to perform the upgrade. Failure to follow the instructions will almost certainly result in the loss of data.

It is necessary to install new versions of the *boot*(A) and *soap*(A) servers.

Programs that ran under Amoeba 5.2 or earlier might not run under Amoeba 5.3 so it is better to recompile them.

The version of X Windows source required for Amoeba 5.3 is X11R6, patch level 13.

Below is a list of important changes between releases that might affect you. Other minor bug fixes are not listed.

2.1. Major Changes from Amoeba 5.2 to Amoeba 5.3

The following changes have been made in Amoeba 5.3:

1. The directory layout has changed in important ways to enhance system security. It is possible to convert the old directory graph to the new layout. The script *amupgrade* has been provided for this purpose.
2. Most of the include files have been modified for ANSI C and POSIX name-space pollution conformance. Library function names now begin with an underscore in almost every case.
3. The include file *src/h/disk.h* is obselete. It will be removed in the next release. Its contents has been moved to *src/h/server/disk/disk.h*. Clients of the disk server should include *src/h/module/disk.h* to obtain client stub prototypes and data structures.
4. The include file *src/h/proc.h* is obselete. It will be removed in the next release. Its contents has been moved to *src/h/server/process/proc.h*. Clients of the process server should include *src/h/module/proc.h* to obtain client stub prototypes and data structures.
5. The include file *src/h/tape.h* is obselete. It will be removed in the next release. Its contents has been moved to *src/h/server/tape/tape.h*. Clients of the tape server should include *src/h/module/tape.h* to obtain client stub prototypes and data structures.
6. Under the direction of the *run*(A) server, the process server in each kernel regularly changes the capability it accepts for executing new processes. This means that just because somebody was able to execute a process on a particular host at time T, they will not be able to do so at time T+Δ unless the system is still willing to allow access to that host.
7. It is now possible to provide SPARCstations with secret numbers in the NVRAM in the IDPROM, from which an unpredictable host capability can be made. This, in combination with the changes to the process server, allows for security of kernel directories and hosts.
8. The program *rip2icmp*(A) has been replaced with the program *irdpd*(A).
9. The *grp_create* (see *grp*(L)) function no longer includes the header size as part of the buffer size.
10. The program *chkbuild*(A) has been provided for helping to analyze the output of *build*(A).
11. Orca has been unbundled from Amoeba. A new more efficient version of Orca should be available for Amoeba in the second quarter of 1996.
12. The command-line interfaces to the IP server administrative programs like *add_route*(A) have been unified. See the manual pages for details.
13. The *soap*(A) server has been modified. It now keeps a version number for each directory. This is used by *starch*(U) for making incremental backups. The new *soap* server recognizes old soap directories and upgrades them automatically. The upgrade process however can take several days.
14. *Starch*(U) has been modified. It is now able to make incremental dumps, in addition to full dumps. See the manual page for details.
15. A FLIP driver for SunOS 5.X (Solaris 2.X) is now available.

16. The TCPIP server is much more robust. Telnet and FTP clients and servers are provided.

17. Many customers had problems booting Amoeba on PCI-bus machines and sometimes on AT-bus machines. The main symptom was that the kernel gave a ‘‘panic: interrupt queue overflow’’ while trying to do the disk server initialization. This problem has been resolved. It was due to the SCSI driver falsely believing it had found a SCSI card.

18. The i80386 kernel Ethernet ‘‘configuration warning’’ messages now only appear if the kernel is booted with the `-netdebug:1` flag.

19. The scheduling time-slice has been reduced from 100ms to 10ms.

20. The source code for a fault-tolerant, self-replicating version of the Bullet Server, known as the Group Bullet Server, has been provided with the sources. The code has not been thoroughly tested but is included for those seeking examples of how to write such a server. The source can be found in *src/kernel/server/gbullet*.

21 The source code for a fault-tolerant, self-replicating version of the Soap Server, known as the Group Soap Server, has been provided with the sources. The code has not been thoroughly tested but is included for those seeking examples of how to write such a server. The source can be found in *src/admin/gsoapsvr*.

2.2. Major Changes from Amoeba 5.1 to Amoeba 5.2

1. The System Administration Guide now contains a section on how to upgrade an existing Amoeba system to the new distribution. The Guide has also been revised in the light of comments from customers and newly supported hardware.
2. The SPARC® port for the Sun4c and Sun4m (V8 SPARC) is available. It is a beta-test version (caveat emptor!). At present there is no audio driver and as yet no support for the *reboot*(A) command. The floppy driver for the Sun4m will appear shortly.
3. The Amoeba i80386 X server has been integrated into the standard XFree distribution. A number of X utilities have been added to the binary distribution and several additional graphics and mouse drivers are now supported.
4. The POSIX conformance has been significantly improved. See *posix*(L) for details of conformance.
5. Various bugs in the FLIP group communication have been fixed.
6. The Novell NE2000 and NE1000 Ethernet cards and the 3Com503 Etherlink II Ethernet card are now supported for the i80386 architecture.
7. The speed of bootstrapping from floppy disk on the i80386 has been dramatically improved.
8. *Fdisk*(A) was wasting a cylinder per disk partition. This has been repaired.
9. The numbering of physical disk devices has been altered to be more logical. Formerly they were in reverse natural order.
10. The GNU C compiler is available for native Amoeba for all supported architectures.
11. A suite of test programs has been added to the distribution. The programs are described in the *System Administration* guide. The binaries can be found in the directory subgraph */public/module/test*.
12. The group error codes in the left column below have been removed from Amoeba. In their place the group communication returns the error code listed in the right column. Any code that tests for the obsolete error conditions should be changed. The function *err_why*(L) now supports the remaining BC_ error codes.

Obsolete Error	Replaced By
BC_OK	STD_OK
BC_NOMEM	STD_NOMEM
BC_ILLARG	STD_ARGBAD
BC_EXIST	STD_EXISTS

13. In addition to the RPC and Group communication interfaces there is now a system-call interface to the raw FLIP layer of Amoeba. This makes it possible to implement network protocols on top of FLIP in user space. This should provide a valuable teaching tool for computer networking classes. See *rawflip*(L) for details of how to use this interface.
14. System security has been improved so that Amoeba is more suitable for use in a teaching environment. It is now possible to set the permissions on host capabilities to values other than `FF:FF:FF`. A value of `FF:2:4` is recommended. The new value should be set for all processor capabilities including those in the processor pools.

Unprivileged users should then no longer be able to access kernel memory or virtual disks. Nor should they be able to kill processes they do not own using *stun*(U). The session server (see *session*(U)) has been modified to implement the directory */dev/proc* internally (i.e., it is no longer in the Soap server) and it maintains there the capabilities for all processes created during that session. If a user wishes to stun a process then they can access the capabilities for their own processes via the directory */dev/proc*. System administrators should read the new session server manual page carefully.

15. The program *aminstall*(A) is now available under Amoeba to install the system binaries compiled under Amoeba.

16. In previous versions of Amoeba processes that were dying sometimes remained hanging in ‘‘stunned’’ state. This should no longer occur.

17. The *format*(U) command has been added to format floppy disks. The command *eject*(U) is used to extract floppy disks from the Sun 4 floppy drives.

18. The GNU debugger has been ported to Amoeba, including support for multi-threaded programs. It is available with the third-party software.

19. AIL has been modified to provide better support for arrays. It produces, as a result, slightly different functions. The old functionality can be obtained with the

20 Support has been added to the i80386 for the Adaptec AHA-154X SCSI cards. This makes it possible to load the distribution from tape. **–v 3.5** option.

2.3. Changes from Amoeba 5.0 to Amoeba 5.1

1. Support for the GNU C compiler and assembler for compiling Amoeba under UNIX has been added. This consists of a new Amake toolset and various changes to the Amoeba sources to eliminate compiler warnings and to support the GNU assembler syntax. The additional sources needed for the GNU compiler and loader to make Amoeba binaries are available in the third-party software.

   Note that no binaries will be distributed that have been made with GNU compilers. It is up to customers to compile Amoeba with the GNU toolset if they wish to take advantage of the extra performance that can be gained from using the GNU compiler.

2. Support for profiling of user programs has been added to *ack*(U). The program must be compiled with the **−p** option and may be linked with standard libraries or profiling libraries, depending on the degree of profiling required. See *profiling*(L) and the **−p** option of *ack*(U) for details.

3. The X11R5.am tree structure has been modified to contain the directory *mit* and its siblings. When the configuration is made it is then able to build the *contributed* X programs as well.

   NOTE: The Amoeba version of X11R5 assumes patch-level 17 or higher for the original X sources.

4. The Xfree 386 X server accelerator has been incorporated into Amoeba. Several bugs in the Amoeba version of the i80386 X server, including the occasional disappearance of the mouse cursor, have been resolved as a consequence.

5. Hardware floating-point support has been added to the i80386 port. If floating-point hardware is present then user programs can be compiled without the software floating-point option.

6. *Amake* has been modified so that it is no longer dependent upon other programs such as *mkproto* for its compilation. It uses the *stdio* library instead of various ACK modules. Therefore it should be much simpler to port it to other operating systems and architectures.

7. A security bug has been removed from *login*(A). The bug will not be described here since it may compromise security at sites not yet upgraded to Amoeba 5.1.

8 The program *be*(U) has been removed from the distribution. Those desirous of an Emacs-like editor can use *jove*(U).

2.4. Changes from Amoeba 4.0 to Amoeba 5.0

1. The FLIP network protocol has replaced the Amoeba 4.0 network protocol. FLIP not only provides RPC, it also provides reliable broadcast for group communication. See *rpc*(L) and *grp*(L) for full details of how to use the remote procedure call and group communication.

   Advantages of FLIP:

   - FLIP has a higher throughput than the old protocol.
   - It provides internetworking.
   - Transactions size may be up to 2^{31} are possible, although the current RPC layer does not yet support the larger transactions.
   - It provides reliable broadcast.

   Disadvantages of FLIP:

   - Small RPCs are slightly slower.
   - FLIP kernels use lots of buffers and require more memory than the old Amoeba kernels.

2. The semantics of *ports* have changed. The F-boxes described in many papers have been implemented in software in the FLIP layer. There is now a difference between *get-ports* and *put-ports*. The routine *priv2pub*(L) is used to convert a *get-port* to a *put-port*. *Getreq* and *rpc_getreq* must be called with the *get-port* in the header parameter and *trans* and *rpc_trans* must be called with a *put-port* in the header. See the relevant manual pages and the section on writing clients and servers in the *Programming Guide* for more details.

3. There are two new standard routines: *std_getparams* and *std_setparams* for setting the values of server parameters. Parameters can be set per server or per object, depending upon the implementation of the server.

4. The ANSI C and POSIX conformance have been improved. Total POSIX conformance is not yet possible.

5. The performance and functionality of the TCP/IP server has been improved. There are also more support utilities and library routines.

6. The boot server (see *boot*(A)) now has the ‘‘after’’ clause to specify dependencies. The most common use for this is to make the server wait until it has started the name server before attempting to start all the programs that depend on it.

7. The load balancing server (see *run*(A)) now supports heterogeneous processor pools. That is, pools containing processors of more than one type of architecture. It first calculates the intersection of architectures in the pool and the architectures for which there is a binary for the desired program. It then selects the most appropriate machine from that subset according to the algorithm provided. The parameters for the algorithm can be set on a per-pool basis by the system administrator.

8. The Swiss Army Knife (SAK) Server has been included. It allows RPCs to be delayed until after some event. This is used to provide *cron*(U) and *at*(U) utilities, although there are many other possibilities. See *sak*(A) for more information.

9 An *f77*(U) front end has been added to ACK. It is implemented using the program

f2c(U). This converts Fortran 77 to C.

10 TEX has been ported to Amoeba. It requires large amounts of memory to process even moderately sized documents but is otherwise fully functional.

11 The mail transport facility MMDF II and its associated utilities have been ported to Amoeba. It is available to those who qualify under its redistribution restrictions.

12 The X11R5 Window system has been ported to Amoeba.

13 Major structural improvements have been made to the name server (see *soap*(A)).

14 The command *aman*(U) provides on-line manuals and keyword searching thereof.

15 The command *mkbootpartn*(A) has been provided for constructing bootable floppies and hard disk partitions.

16 Each manual now has an index. The index is generated automatically based on keywords embedded in the text. It is simple to add extra index entries.

17 The command *chm*(U) takes its parameters in a different order and format so that it can accept multiple object names.

18 The command *mkhost*(A) has been modified slightly to match the restoration of the basic kernel directory security. This means that it is necessary to remake all the host capabilities!

19 The command *disklabel*(A) has been modified so that it can run on any host and correctly label disks attached to any other Amoeba host in the system. This replaces the old *disklabel.sun* and *disklabel.ibm_at* programs.

20 A bug in the i80386 disk labeling has been detected and removed. Sites with a previously installed version of Amoeba should either rerun *disklabel*(A) and adjust their disk partitioning or reinstall Amoeba from the coldstart.

21 The command *reboot*(A) has been modified to accept multiple host names to reboot and it also allows special arguments to be passed to the kernel, such as the **–noreboot:1** argument for the i80386 kernels.

22 Several bugs in the UNIX FLIP driver have been fixed. In particular, that the size of the reply buffer for *trans* was ignored, sometimes resulting in overwriting the client's memory.

23 The bug in the *boot*(A) server whereby it failed to restart a dead service, even though it had detected that the service needed booting, has been fixed.

2.5. Expected Changes in Future Releases

The following changes are expected in future releases of Amoeba:

1. Bigger capabilities - this implies that you must use the routines for manipulating capabilities and not directly access their contents! It also requires extreme caution with unsigned 16-bit types such as `bufsize` and `command`. They will become signed and 32-bit in future releases. Macros are provided for converting them to `errstat`.
2. Service timeouts on RPCs are being considered. *Getreq* will be able to listen to a set of ports instead of just one port.
3. New process Management - we hope that the high level interface will stay the same but that the use of system calls will decrease. In addition, stacks should grow automatically. At present the stack size is fixed at link time.
3. Process migration should become possible at any point in a process' execution. At present it is only possible when a process is not doing a system call. N.B. Although technically possible, process migration is **not** implemented.
4. Paging or swapping may be added. At present, processes which do not fit completely in memory will not run.

In addition to these major changes there will be internal improvements to various servers to overcome known deficiencies.

3 Bibliography

This bibliography contains a list of all of the published papers about Amoeba In addition to this there is a small collection of unpublished technical reports, some of which have been submitted for publication.

1. *The Amoeba Distributed Operating System: Selected papers 1984 − 1987,* CWI tract 41, Amsterdam (1987).
2. Baalbergen, E.H., ‘‘Parallel and Distributed Compilations in Loosely-Coupled Systems: A Case Study,’’ *Proc. Workshop on Large Grain Parallelism*, Providence, RI (Oct 1986).
3. Baalbergen, E.H., ‘‘Design and Implementation of Parallel Make,’’ *Computing Systems* **1**(2), pp. 135-158 (Spring 1988).
4. Baalbergen, E.H., Verstoep, C., and Tanenbaum, A.S., ‘‘On the Design of the Amoeba Configuration Manager,’’ *ACM SIGSOFT Software Engineering Notes* **17**, ACM, Proc. 2nd Int. Workshop on Software Configuration Management (ACM) (Nov 1989).
5. Bacon, J.M., Horn, C., Langsford, A., Mullender, S.J., and Zimmer, W., ‘‘MANDIS: Architectural Basis for Management,’’ *Proc. of the EUTECO 88 Conf.*, Vienna, Austria, pp. 795-809, North-Holland (Apr 1988).
6. Bal, H.E. and Renesse, R. van, ‘‘A Summary of Parallel Alpha-Beta Search Results,’’ *ICCA Journal* **9**, pp. 146-149 (Sept 1986).
7. Bal, H.E. and Renesse, R. van, ‘‘Parallel Alpha-Beta Search,’’ *Proc. 4th NGI-SION Symposium Stimulerende Informatica*, Utrecht, pp. 379-385 (Apr 1986).
8. Bal, H.E., Renesse, R. van, and Tanenbaum, A.S., ‘‘Implementing Distributed Algorithms Using Remote Procedure Calls,’’ *Proc. of the 1987 National Computer Conf.*, Chicago, Ill, pp. 499-506 (June 1987).
9. Bal, H.E. and Tanenbaum, A.S., ‘‘Distributed Programming with Shared Data,’’ *Proc. IEEE CS 1988 Int. Conf. on Computer Languages*, Miami, FL, pp. 82-91 (Oct 1988).
10. Bal, H.E., Steiner, J.G., and Tanenbaum, A.S., ‘‘Programming Languages for Distributed Computing Systems,’’ *ACM Computing Surveys* **21**(3), pp. 261-322 (Sept 1989).
11. Bal, H.E., Kaashoek, M.F., and Tanenbaum, A.S., ‘‘A Distributed Implementation of the Shared Data-Object Model,’’ *Proc. 1st USENIX/SERC Workshop on Experiences with Building Distributed and Multiprocessor Systems*, pp. 1-19, IEEE (Oct 1989).
12. Bal, H.E., Kaashoek, M.F., Tanenbaum, A.S., and Jansen, A.J., ‘‘Replication Techniques for Speeding Up Parallel Applications,’’ IR-202, Dept Mathematics and Computer Science, Vrije Universiteit, Amsterdam (Oct 1989). (to appear in Concurrency—Practice and Experience)
13. Bal, H.E., Kaashoek, M.F., and Tanenbaum, A.S., ‘‘Experience with Distributed Programming in Orca,’’ *Proc. Int. Conf. on Comp. Languages ’90*, pp. 79-89, IEEE (March 1990).
14. Bal, H.E., Tanenbaum, A.S., and Kaashoek, M.F., ‘‘Orca: A Language for Distributed Programming,’’ *ACM SIGPLAN Notices* **25**(5), pp. 17-24 (May 1990).

15. Bal, H.E., *Programming Distributed Systems*, Silicon Press, Summit, NJ (1990).

16. Bal, H.E., *Programming Distributed Systems,* Prentice Hall International, Hemel Hempstead, England (1991).

17. Bal, H.E. and Tanenbaum, A.S., ‘‘Distributed Programming with Shared Data,’’ *Computer Languages* **16**, pp. 129-146 (Feb 1991).

18. Bal, H.E., ‘‘A Comparative Study of Five Parallel Programming Languages,’’ *EurOpen Spring 1991 Conference on Open Distributed Systems*, Tromso, Norway, pp. 209-228 (May 1991).

19. Bal, H.E. and Kaashoek, M.F., ‘‘Data Distribution in Orca through Compiler Optimization,’’ IR-254, Vrije Universiteit, Dept. of Math. and Comp. Sci., Amsterdam (Sept 1991).

20. Bal, H.E., Kaashoek, M.F., and Tanenbaum, A.S., ‘‘Parallel Programming on Amoeba Using Efficient Distributed Shared Memory,’’ pp. 89-105 in *Scientific Computing on Supercomputers III*, Plenum Press (1992). (Proc. of 6th Int’l Conf. on the Use of Supercomputers in Theoretical Science, Jan 1991)

21. Bal, H.E., Kaashoek, M.F., and Tanenbaum, A.S., ‘‘Orca: A language for Parallel Programming of Distributed Systems,’’ *IEEE Transactions on Software Engineering* **18**(3), pp. 190-205 (Mar 1992).

22. Bal, H.E., Kaashoek, M.F., Tanenbaum, A.S., and Jansen, J., ‘‘Replication Techniques for Speeding up Parallel Applications on Distributed Systems,’’ *Concurrency Practice & Experience* **4**(5), pp. 337-355 (Aug 1992).

23. Bal, H.E. and Kaashoek, M.F., ‘‘Object Distribution in Orca using Compile-Time and Run-Time Techniques,’’ *Systems, Languages and Applications*, Washington, D.C. (26 Sep - 1 Oct 1993).

24. Bhoedjang, R., Rühl, T., Hofman, R., Langendoen, K., Bal, H., and Kaashoek, F., ‘‘Panda: A Portable Platform to Support Parallel Programming Languages,’’ *Proc. of the 4th Symposium on Experiences with Distributed and Multiprocessor Systems*, San Diego, pp. 213-226 (Sep 1993).

25. " Doorn, Leendert van.as [A " and Tanenbaum, Andrew S., ‘‘Using Active Messages to Support Shared Objects,’’ *Proc. Sixth ACM SIGOPS European Workshop, ACM*, Wadern, Germany, pp. 112-116 (Sep 1994).

26. Douglis, F., Kaashoek, M.F., Tanenbaum, A.S., and Ousterhout, J.K., ‘‘A Comparison of Two Distributed Systems: Amoeba and Sprite,’’ *Computing Systems*, pp. 353-384 (Fall 1991).

27. Hall, J., Renesse, R. van, and Staveren, J.M. van, ‘‘Gateways and Management in an Internet Environment,’’ *Proc. of the IFIP TC6 WG6.4A Int. Workshop on LAN Management*, Berlin (West), Hahn-Meitner-Institute (July 1987).

28. Holden, D.B., Anshus, O.J., Fallmyr, T., Hall, J., Renesse, R. van, Staveren, J.M. van, and Skogseth, G., ‘‘An Approach to Monitoring in Distributed Systems,’’ *Proc. of the EUTECO 88 Conf.*, Vienna, Austria, pp. 811-821, North-Holland (Apr 1988).

29. Johansen, D. and Anshus, O.J., ‘‘A Distributed Diary Application,’’ *Proc. of the IFIP TC 6 First Iberian Conf. on Data Communications*, Lisbon, Portugal, North-Holland (May 1987).

30. Kaashoek, M.F., Tanenbaum, A.S., Flynn Hummel, S., and Bal, H.E., ‘‘An Efficient Reliable Broadcast Protocol,’’ *Operating Systems Review* **23**, pp. 5-19, ACM (Oct 1989).

31. Kaashoek, M.F., Bal, H.E., and Tanenbaum, A.S., ‘‘Experience with the Distributed Data Structure Paradigm in Linda,’’ *Proc. 1st USENIX/SERC Workshop on Experiences with Building Distributed and Multiprocessor Systems*, pp. 175-191, IEEE (Oct 1989).

32. Kaashoek, M.F. and Tanenbaum, A.S., ‘‘Fault Tolerance Using Group Communication,’’ *Proc. of 4th ACM SIGOPS European Workshop*, Bologna, Italy (Sept 1990).

33. Kaashoek, M.F. and Tanenbaum, A.S., ‘‘Group Communication in the Amoeba Distributed Operating System,’’ *Proc. 11th Int. Conf. on Distributed Computer Systems*, Arlington, TX, IEEE Computer Society (May 1991).

34. Kaashoek, M.F., Michiels, R., Bal, H.E., and Tanenbaum, A.S., ‘‘Transparent Fault-tolerance in Parallel Orca Programs,’’ *Symposium on Experiences with Distributed and Multiprocessor Systems III*, Newport Beach, pp. 297-312 (Mar 1992).

35. Kaashoek, M.F., ‘‘Group Communication in Distributed Computer Systems,’’ Ph.D. Thesis, Vrije Universiteit, Amsterdam (1992).

36. Kaashoek, M.F., Tanenbaum, A.S., and Verstoep, K., ‘‘An Experimental Comparison of Remote Procedure Call and Group Communication,’’ *Proc. Fifth ACM SIGOPS European Workshop* , Le Mont Saint-Michel, France (Sept. 1992).

37. Kaashoek, M.F., Tanenbaum, A.S., and Verstoep, K., ‘‘Group Communication in Amoeba and its Applications,’’ *Proc. EurOpen*, Utrecht, The Netherlands (Nov. 1992).

38. Kaashoek, M.F., Renesse, R. van, Staveren, H. van, and Tanenbaum, A.S., ‘‘FLIP: an Internetwork Protocol for Supporting Distributed Systems,’’ *ACM Trans. Comp. Syst.* **11**(1), pp. 73-106 (Feb 1993).

39. Kaashoek, M.F., Tanenbaum, A.S., and Verstoep, K., ‘‘Using Group Communication to Implement a Fault-Tolerant Directory Service,’’ *Proc. 13th International Conference on Distributed Computing Systems*, Pittsburgh, PA, pp. 130-139 (May 1993).

40. Kaashoek, M.F., Tanenbaum, A.S., and Verstoep, K., ‘‘Group Communication in Amoeba and its Applications,’’ *Distributed Systems Engineering Journal* (Jul/Aug 1993).

41. Levelt, W.G., Kaashoek, M.F., Bal, H.E., and Tanenbaum, A.S., ‘‘A Comparison of Two Paradigms for Distributed Shared Memory,’’ *Software−Practice and Experience* **22**(11), pp. 985-1010 (Nov 1992).

42. Moergestel, L.J. van, Bal, H.E., Kaashoek, M.F., Renesse, R. van, Sharp, G.J., Staveren, H. van, and Tanenbaum, A.S., ‘‘Amoeba on a Multiprocessor,’’ IR-206, Dept Mathematics and Computer Science, Vrije Universiteit, Amsterdam (Dec 1989). (to appear in in Force File)

43. Mullender, S.J. and Tanenbaum, A.S., ‘‘Protection and Resource Control in Distributed Operating Systems,’’ *Computer Networks* **8**(5-6), pp. 421-432 (Oct 1984).

44. Mullender, S.J. and Vitányi, P.M.B., ‘‘Distributed Match-Making for Processes in Computer Networks,’’ *Proc. of the 4th ACM Conf. on Principles of Distributed Computing*, Minaki, Canada (Aug 1985).

45. Mullender, S.J., ‘‘Principles of Distributed Operating System Design,’’ PhD. Thesis, SMC, Amsterdam (Oct 1985).

46. Mullender, S.J. and Tanenbaum, A.S., ‘‘A Distributed File Service Based on Optimistic Concurrency Control,’’ *Proc. of the 10th ACM Symposium on Operating System Principles*, Orcas Island, Washington, USA, pp. 51-62 (Dec 1985).

47. Mullender, S.J. and Renesse, R. van, ‘‘Accommodating Heterogeneity in the Amoeba Distributed System,’’ *Proc. of SOSP Heterogeneity Workshop*, Orcas Island, Washington, USA (Dec 1985).

48. Mullender, S.J., ‘‘Making Amoeba Work,’’ *Proc. of 2nd SIGOPS Workshop Making Distributed Systems Work*, Amsterdam (Sept 1986).

49. Mullender, S.J., ‘‘The Amoeba Distributed Operating System II,’’ *CWI Newsletter*, Amsterdam(12), Centre for Mathematics and Computer Science (Sept 1986).

50. Mullender, S.J. and Tanenbaum, A.S., ‘‘The Design of a Capability-Based Distributed Operating System,’’ *The Computer Journal* **29**(4), pp. 289-300 (Mar 1986).

51. Mullender, S.J., ‘‘Process Management in Distributed Operating Systems,’’ in *Proc. of Workshop ‘Experiences with Distributed Systems’ in Computer Science Lecture Notes*, Springer Verlag, Berlin (Sept 1987).

52. Mullender, S.J., ‘‘Amoeba as a Workstation Operating System,’’ *Proc. of the Workshop on Workstation Operating Systems*, Cambridge, Ma. (Nov 1987).

53. Mullender, S.J., ‘‘Distributed Operating Systems,’’ *Computer Standards & Interfaces* **6**(1), pp. 37-44 (1987).

54. Mullender, S.J., ‘‘Distributed Operating Systems: State-of-the-Art and Future Directions,’’ *Proc. of the EUTECO 88 Conf.*, Vienna, Austria, pp. 57-66, North-Holland (Apr 1988).

55. Mullender, S.J. and Vitányi, P.M.B., ‘‘Distributed Match-Making,’’ *Algorithmica, 2nd special issue on distributed algorithms* (1988).

56. Mullender, S.J., Rossum, G. van, Tanenbaum, A.S., Renesse, R. van, and Staveren, H. van, ‘‘Amoeba: A Distributed Operating System for the 1990s,’’ *Computer* **23**(5), pp. 44-53 (May 1990).

57. Renesse, R. van, ‘‘From UNIX to a Usable Distributed Operating System,’’ *Proc. of the EUUG Autumn ’86 Conf.*, Manchester, UK, pp. 15-21 (Sept 1986).

58. Renesse, R. van, Tanenbaum, A.S., Staveren, J.M. van, and Hall, J., ‘‘Connecting RPC-Based Distributed Systems Using Wide-Area Networks,’’ *Proc. of the 7th Int. Conf. on Distributed Computing Systems*, Berlin (West), pp. 28-34 (Sept 1987).

59. Renesse, R. van, Tanenbaum, A.S., and Sharp, G.J., ‘‘The Workstation: Computing Resource or Just a Terminal?,’’ *Proc. of the Workshop on Workstation Operating Systems*, Cambridge, MA (Nov 1987).

60. Renesse, R. van, Staveren, J.M. van, Hall, J., Turnbull, M., Janssen, A.A., Jansen, A.J., Mullender, S.J., Holden, D.B., Bastable, A., Fallmyr, T., Johansen, D., Mullender, K.S., and Zimmer, W., ‘‘MANDIS/Amoeba: A Widely Dispersed Object-Oriented Operating System,’’ *Proc. of the EUTECO 88 Conf.*, Vienna, Austria, pp. 823-831, North-Holland (Apr 1988).

61. Renesse, R. van and Tanenbaum, A.S., ‘‘Voting with Ghosts,’’ *Proc. of the 8th Int.*

Conf. on Distributed Computing Systems, San Jose, CA, pp. 456-461 (June 1988).

62. Renesse, R. van, Staveren, J.M. van, and Tanenbaum, A.S., ‘‘The Performance of the World’s Fastest Distributed Operating System,’’ *ACM Operating Systems Review* **22**(4), pp. 25-34 (Oct 1988).

63. Renesse, R. van, Staveren, J.M. van, and Tanenbaum, A.S., ‘‘The Performance of the Amoeba Distributed Operating System,’’ *Software—Practice and Experience* **19**(3), pp. 223-234 (Mar 1989).

64. Renesse, R. van, Tanenbaum, A.S., and Wilschut, A., ‘‘The Design of a High-Performance File Server,’’ *Proc. of the 9th Int. Conf. on Distributed Computing Systems*, Newport Beach, CA, pp. 22-27 (June 1989).

65. Renesse, R. van and Tanenbaum, A.S., ‘‘Short Overview of Amoeba,’’ *Proc. of the USENIX workshop on Micro-Kernels and Other Kernel Architectu*, Seattle, Washington, pp. 1-10 (Apr 27-28, 1992).

66. Romein, J.W. and Bal, H.E., ‘‘Parallel N-Body Simulation on a Large-Scale Homogeneous Distributed System,’’ pp. 473-484 in *EURO-PAR’95 Parallel Processing, Lecture Notes in Computer Science, 966*, ed. S. Haridi, K. Ali & P. Magnusson, Springer-Verlag, Stockholm, Sweden (Aug 1995).

67. Rossum, G. van, ‘‘AIL – A Class-Oriented Stub Generator for Amoeba,’’ *Proc. of the Workshop on Experience with Distributed Systems*, Springer Verlag (1989).

68. Sharp, G.J., ‘‘The Design of a Window System for Amoeba,’’ IR-142, Vrije Universiteit, Dept. of Math. and Comp. Sci., Amsterdam (Dec 1987).

69. Steketee, C.F., Zhu, W.P., and Moseley, P.A., ‘‘Implementation of Process Migration in Amoeba,’’ *Proc. 14th International Conference on Distributed Computing Systems*, Poznan, Poland, pp. 194-201, IEEE (1994).

70. Steketee, C.F., Socko, P., and Kiepuszewski, B., ‘‘Experiences with the Implementation of a Process Migration Mechanism for Amoeba,’’ *Australian Computer Science Communications* (1996).

71. Tanenbaum, A.S., Kaashoek, M.F., and Bal, H.E., ‘‘Using Broadcasting to Implement Distributed Shared Memory Efficiently,’’ pp. 387-408 in *Readings in Distributed Computing Systems*, ed. T.L. Casavant and M. Singhal, IEEE Press (1994).

72. Tanenbaum, A.S. and Renesse, R. van, ‘‘Distributed Operating Systems,’’ *ACM Computing Surveys* **17**(4), pp. 419-470 (Dec 1985).

73. Tanenbaum, A.S. and Renesse, R. van, ‘‘Making Distributed Systems Palatable,’’ *Proc. of 2nd SIGOPS Workshop Making Distributed Systems Work*, Amsterdam (Sept 1986).

74. Tanenbaum, A.S. and Renesse, R. van, ‘‘Research Issues in Distributed Operating Systems,’’ pp. 35-46 in *Computing in High-Energy Physics*, ed. L.O. Hertzberger and W. Hoogland, North-Holland, Amsterdam (1986).

75. Tanenbaum, A.S., Mullender, S.J., and Renesse, R. van, ‘‘Using Sparse Capabilities in a Distributed Operating System,’’ *Proc. of the 6th Int. Conf. on Distributed Computing Systems*, Cambridge, MA, pp. 558-563 (May 1986).

76. Tanenbaum, A.S. and Renesse, R. van, ‘‘Reliability Issues in Distributed Operating Systems,’’ *Proc. of the 6th Symposium on Reliability in Distributed Software and Database Systems*, Kingsmill-Williamsburg, VA, pp. 3-11 (Mar 1987).

77. Tanenbaum, A.S. and Renesse, R. van, ‘‘A Critique of the Remote Procedure Call Paradigm,’’ *Proc. of the EUTECO 88 Conf.*, Vienna, Austria, pp. 775-783, North-Holland (Apr 1988).

78. Tanenbaum, A.S., Renesse, R. van, Staveren, H. van, Sharp, G.J., Mullender, S.J., Jansen, A.J., and Rossum, G. van, ‘‘The Amoeba Distributed Operating System,’’ *Proc. of ESTEC Workshop on Communication Networks and Distributed Operating Systems Within the Space Environment*, Noordwijk, The Netherlands, ESTEC (Oct 1989).

79. Tanenbaum, A.S., Renesse, R. van, Staveren, H. van, and Sharp, G.J., ‘‘Beyond UNIX - A True Distributed System for the 1990s,’’ *Proc. UKUUG Summer Conference*, pp. 251-260 (1990).

80. Tanenbaum, A.S., Renesse, R. van, Staveren, H. van, Sharp, G.J., Mullender, S.J., Jansen, A.J., and Rossum, G. van, ‘‘Experiences with the Amoeba Distributed Operating System,’’ *CACM* **33**(12) (Dec 1990).

81. Tanenbaum, A.S., Renesse, R. van, Staveren, H. van, Sharp, G.J., Mullender, S.J., Jansen, A.J., and Rossum, G. van, ‘‘Ervaring met het Gedistribueerde Bedrijfsysteem Amoeba,’’ *Informatie* **33**(2) (Feb 1991).

82. Tanenbaum, A.S., Kaashoek, M.F., Renesse, R. van, and Bal, H., ‘‘The Amoeba Distributed Operating System - A Status Report,’’ *Computer Communications* **14**, pp. 324-335 (Jul/Aug 1991).

83. Tanenbaum, A.S., Kaashoek, M.F., and Bal, H.E., ‘‘Parallel Programming Using Shared Objects and Broadcasting,’’ *IEEE Computer* **25**(8), pp. 10-19 (Aug 1992).

84. Tanenbaum, A.S., Bal, H.E., and Kaashoek, M.F., ‘‘Programming a Distributed System Using Shared Objects,’’ *Proc. 2nd Int’l Symposium on High-Performance Distributed Computing*, Spokane, WA, IEEE (Jul 1993).

85. Tanenbaum, A.S. and Kaashoek, M.F., ‘‘The Amoeba Microkernel,’’ *Distributed Open Systems, IEEE Press*, pp. 11-30 (1994).

86. Tanenbaum, A.S., ‘‘A Comparison of Three Microkernels,’’ *Journal of Supercomputing* **9**(1/2), pp. 7-22 (1995).

87. Verstoep, K. and Sharp, G.J., ‘‘The Amoeba Processor Reservation System - Process Creation and Processor Scheduling in Amoeba,’’ IM-11, Dept Mathematics and Computer Science, Vrije Universiteit, Amsterdam (Jul 1995).

88. Zhu, W.P., Steketee, C.F., and Muilwijk, B., ‘‘Load balancing and workstation autonomy on Amoeba,’’ *Australian Computer Science Communications* **17**(1), pp. 588-597 (Feb 1995).

89. Zhu, W.P. and Steketee, C.F., ‘‘An Experimental Study of Load Balancing on Amoeba,’’ *Proc. Aizu International Symposium on Parallel Algorithms / Architecture Synthesis*, Aizu-Wakamatsu, Japan (1995).

4 Bug Reports

Below is the list of reported bugs not resolved in this release of Amoeba. Their current status and level of importance is listed for each bug report.

For academic sites, if you discover a bug or wish to request an enhancement to the system which is not yet on the list, the contact person for your site should send electronic mail to *amoeba-support@cs.vu.nl*. Send this from an account where a response can be received, or provide in the message details of where the response should be sent. Furthermore, if the problem seems to be in the kernel, describe the hardware where the problem arises, even if it seems irrelevant to the problem.

```
------------------  Amoeba Modification Request #10 ------------

Date:         Thu Apr  5 14:24:30 MET DST 1990
Reporter:     siebren (Siebren van der Zee,M346,020-5964098)
Category:     Enhancement
Module:       /bin/sh
Severity:     Minor
Summary:      $? incorrect when child process interrupted
Handler:      condict
Status:       Open

Description:

Repeat-by:

$ cat     # This will block. Hit ^C to send it SIGINT and
     # your prompt comes back
$ echo $?
-1
$

This is not what happens on the unix machines I could check.
They set $? to 130. (Ultrix 2.?, 4.3 BSD, SunOs).
```

```
------------------  Amoeba Modification Request #37 ------------

Date:         Fri Jun 15 09:22:00 MET DST 1990
Reporter:     gregor@cs.vu.nl (Greg Sharp)
Category:     Bug
Module:       sun3 kernel
Severity:     Minor
Summary:      The sun3 does not reboot properly when booted from disk
Handler:      gregor
Status:       Open

Repeat By:

Boot Amoeba from disk on a sun3 and then type <home>-r

Description:

When Amoeba is booted from disk on a sun3 the <home>-r command
to return to the monitor sometimes generates error messages from
the monitor.  This is because the page tables that the monitor
expects to get back have been stomped on by Amoeba.


------------------  Amoeba Modification Request #38 ------------

Date:         Fri Jun 15 09:25:22 MET DST 1990
Reporter:     gregor@cs.vu.nl (Greg Sharp)
Category:     Enhancement
Module:       /bin/sh
Severity:     Important
Summary:      there is no unset command in sh
Handler:      condict
Status:       Open

Description:

The shell does not have an unset command.  This is irritating when
one needs to get rid of a shell string environment variable.
```

```
------------------  Amoeba Modification Request #70 ------------

Date:        Wed Aug  8 11:37:29 MET DST 1990
Reporter:    jack (& Jansen,M346,4098,160335)
Category:    Enhancement
Module:      kernel
Severity:    Minor
Summary:     User program output should sometimes show in printbuf
Handler:     <to be assigned>
Status:      Open

Description:

     Currently, output from user programs (i.e. anything written
     to <machine>/tty:00) does not show up in the printbuf. Usually,
     this is what you want, but there are exceptions. For instance,
     to examine the output the bootserver and the soapserver print
     on their respective console I have to physically walk over to
     the console. Needless to say, this is totally unacceptable:-).

     I suggest adding a device ’console:00’ (or just console) which
     acts exactly like tty:00, but also stores the output in the
     printbuf. This could then be used by the bootserver and friends,
     while allo or the shell still uses tty:00.

     We could implement the console device in the tty driver, by
     either using a bit in the rights field (right to write to the
     printbuffer) or a bit in the object number.


------------------  Amoeba Modification Request #93 ------------

Date:        Fri Oct 12 12:43:56 MET 1990
Reporter:    gregor@cs.vu.nl (Greg Sharp)
Category:    Quality/Performance
Module:      goodport
Severity:    Minor
Summary:     gp_trans returns RPC_BADPORT
Handler:     <to be assigned>
Status:      Open

Description:

The macro gp_trans in goodport.h returns RPC_BADPORT if it detects that
you are attempting to access a port that it knows is dead.  There should
be a separate error status that correctly reflects this error.  BADPORT
is reserved for when you call trans,getreq or putrep with a null port.
```

```
------------------  Amoeba Modification Request #105 ------------

Date:         Wed Oct 24 09:55:16 MET 1990
Reporter:     sater@cs.vu.nl (Hans van Staveren,kamer S414,telef. 5486219)
Category:     Quality/Performance
Module:       timer on Force30
Severity:     Minor
Summary:      timer on force30 is inaccurate
Handler:      <to be assigned>
Status:       Open

Description:

The timer of the Force30 board doesn’t run exactly at 10 Hz.
Tweekings have already been made to the timer code, but somebody should
look at it some time.


------------------  Amoeba Modification Request #125 ------------

Date:         Wed Nov 14 18:02:41 MET 1990
Reporter:     sater@cs.vu.nl (Hans van Staveren,kamer S414,telef. 5486219)
Category:     Bug
Module:       machdep/dev/force30/uart68562.c
Severity:     Important
Summary:      interrupts at wrong vector
Handler:      <to be assigned>
Status:       Open

Repeat By:

By inspection.

Description:

The driver uses one of those fancy info structures to read what its vector
should be. It even tells the hardware. Then it merrily continues doing a
setvec to vector 244. It looks like the original author of the code had good
intentions but could not make them work.

Either the driver should be fixed properly, or the suggestion that the vector
is programmable should be taken out.

Just a job for the same guy that has to figure out why the clock interrupts
at the wrong rate.
```

```
------------------  Amoeba Modification Request #126 ------------

Date:        Thu Nov 15 14:07:48 MET 1990
Reporter:    sater@cs.vu.nl (Hans van Staveren,kamer S414,telef. 5486219)
Category:    Quality/Performance
Module:      force30 kernels
Severity:    Important
Summary:     Force30 kernels do not use memory busrt fill
Handler:     <to be assigned>
Status:      Open

Description:

The Force30 hardware manual suggests that the memory burst fill feature
of the MC68030 will work on the board. Our kernel doesn’t use it.
I know why, it didn’t seem to work. Someone should look at this once.
You are supposed to get a reasonable speedup from this.


------------------  Amoeba Modification Request #146 ------------

Date:        Fri Feb  8 17:30:39 MET 1991
Reporter:    jack@cwi.nl (Jack Jansen, 592 4098)
Category:    Enhancement
Module:      STD_ module
Severity:    Minor
Summary:     a STD_VERSION command would be very useful...
Handler:     <to be assigned>
Status:      Open

Description:

     Something for the far future: I think a STD_VERSION transaction
     would be very useful. A server would reply to this with a standard
     version message like
        foo created March 19, 1990 in /usr/amoeba/.... by jack@schelvis
     I’ve just again run into a problem with an outdated server....
```

```
------------------  Amoeba Modification Request #149 ------------

Date:         Thu Mar  7 17:40:35 MET 1991
Reporter:     beugel@cs.vu.nl (B.J. Beugel, VU Amsterdam.)
Category:     Bug
Module:       sig_raise()
Severity:     Important
Summary:      Raising signal 0 kills the entire process.
Handler:      <to be assigned>
Status:       Open
```

Repeat By:

Call "sig_raise(0)" in some thread and the entire process exits.

Description:

The ’sig_raise()’ system call has no return value so errors are not reported.
According to the manual ’sig_raise()’ should only be called with signals
greater than 1. However raising negative signals (exceptions) with
Unfortunately, raising signal 0 with ’sig_raise()’ kills the process.
Maybe ’sig_raise()’ should report an error condition and check for
"illegal" signals. Just abortiong the process is a bit crude.

```
------------------  Amoeba Modification Request #156 ------------

Date:         Fri Apr 12 13:35:31 MDT 1991
Reporter:     jack@cwi.nl (Jack Jansen, 592 4098)
Category:     Enhancement
Module:       map_segment
Severity:     Minor
Summary:      map_segment doesn’t offer full functionality
Handler:      <to be assigned>
Status:       Open
```

Description:

It has just been brought to my attention that map_segment
doesn’t offer the full functionality of mapping a segment
on program execution: it is not possible to specify the
offset in the file from which the segment should be initialized.

Needless to say, now somebody wants that functionality.

There is a very minor backward compatible change that would

```
     allow this:
     - Change the systemcall map_segment to map_segoff with the
       extra offset parameter.
     - Add a map_segment wrapper to the library that calls
       map_segoff.


------------------  Amoeba Modification Request #167 ------------

Date:        Wed May 22 11:43:40 MET DST 1991
Reporter:    gregor@cs.vu.nl (Greg Sharp)
Category:    Quality/Performance
Module:      Xsvr
Severity:    Important
Summary:     no security for the server capability
Handler:     <to be assigned>
Status:      Open

Description:

When the X server starts up it attempts to write its capability in
/profile/cap/x11svr.  If this is the central directory of /super/cap
then it must be writable for everyone or the X server won't run.
However if this directory is world writable then anyone can tamper
with my X server capability and thus interfere with my workstation.

Several solutions are possible:
1. Everyone has there own profile/cap/x11svr directory where only
   they can write
2. The X server can only be started by the boot server which can
   have sufficient permission to write in /super/cap/x11svr.
No doubt there are more solutions.


------------------  Amoeba Modification Request #168 ------------

Date:        Fri May 31 11:29:18 MET DST 1991
Reporter:    gregor@cs.vu.nl (Greg Sharp)
Category:    Bug
Module:      ksh & sh
Severity:    Minor
Summary:     generates syntax errors when it shouldn't
Handler:     <to be assigned>
Status:      Open

Repeat By:
```

```
for i in 1 2 3 4 5 ; do /super/users/sater/home/dhrystone & ; done
```

(note that a separate line for each part of the loop DOES work.)
Things that don’t work in the ksh but do work in sh.

```
 cat /etc/hosts | wc &
 (cat /etc/hosts | wc ) &
```

Description:

The above command generates the message "syntax error" from the ksh and sh. This command line works fine under unix. I think they still give the wrong operator precedence to &. Subshells with pipes don’t seem to work at all under ksh.

------------------ Amoeba Modification Request #170 ------------

```
Date:        Wed Jun 12 10:23:46 MET DST 1991
Reporter:    versto@cs.vu.nl (5485302)
Category:    Bug
Module:      ail
Severity:    Important
Summary:     multiple arrays in one operation results in wrong server code
Handler:     <to be assigned>
Status:      Open
```

Repeat By:

Consider the following piece of ail code:

```
-----------
class test_class [13..13] {
        test_op(*,
                in char  buf1[len1: 20],
                in int   len1,
                in char  buf2[len2: 20],
                in int   len2
        );
};

generate test_class { server; };
------------
```

Description:

The server mainloop code generated for the operation test_op() is as follows:

```
---------------
            int len1; /* in redeclared (h_extra) */
            int len2; /* in redeclared (h_size) */
            /* in buf1 not declared */
            char buf2[20]; /* in */
            len1 = (short) _hdr.h_extra;
            len2 = (short) _hdr.h_size;
            _adv = _buf;
            if (_hdr.h_size>20) {
                _hdr.h_command = AIL_SVR_IN_BAD;
                goto _fout; /* Marshaling error */
            }
            (void) memcpy(buf2, _adv, _hdr.h_size); _adv += _hdr.h_size;
            _hdr.h_command = impl_test_op(&_hdr, _buf, len1, buf2, len2);
----------------
```

I.e. buf2 is copied from the transaction buffer as if it were right at the beginning of it.
Before copying takes place, the pointer "_adv" should be set to _buf+len1 instead of _buf.

------------------ Amoeba Modification Request #171 ------------

Date: Mon Jun 24 16:12:52 MDT 1991
Reporter: jack@cwi.nl (Jack Jansen, 592 4098)
Category: Bug
Module: /bin/sh and/or ajax exec
Severity: Important
Summary: the shell can hang when machines crash
Handler: <to be assigned>
Status: Open

Repeat By:

Execute a program on a machine that is down. For instance, have one machine of type X in the pool (run server handled), crash it, and execute a binary of that type before the run server notices the crash.

Type control-C while the shell hangs in the exec. The shell will return with an ’I/O error’ error. The next exec will hang forever.

Description:

The shell will longjump out of the exec() (or ajax_newproc, or whatever), but the routine that was trying to do the exec still holds a mutex. The next exec() will hang forever. The mutex in question is ajax_forkmu (I think), in libajax/startproc.c.

Note that this description might be wildly incorrect.

------------------ Amoeba Modification Request #174 ------------

```
Date:         Wed Jun 26 22:15:57 MET DST 1991
Reporter:     beugel@cs.vu.nl (B.J. Beugel, VU Amsterdam)
Category:     Bug
Module:       Korn shell
Severity:     Important
Summary:      The ’trap’ command doesn’t always trap as it should.
Handler:      <to be assigned>
Status:       Open
```

Repeat By:

Make two shell scripts. Name the first one "foo". It goes:

```
echo ’’
echo This is

trap ’echo

bar

echo
echo ’’
```

Name the second script "bar". It should have the lines:

```
echo ’’
echo This is

trap ’echo

sleep 100

echo
echo ’’
```

Execute "foo" in three different ways:

1. Type "foo" at the prompt for the Korn shell. A little while after the

message "This is "bar"" has appeared, hit Ctrl-C to generate a signal.

2. Type "ksh foo" at the shell prompt and hit Ctrl-C as mentioned at 1.

3. Type ". foo" at the shell prompt and hit Ctrl-C as mentioned at 1.

With 1, "bar" catches the signal correctly but as soon as "bar" is done,
"foo" is killed and doesn't get to finish its last few lines.
With 2, "bar" doesn't catch the signal and is consequently killed. I can't
finish its remaining lines. "foo", however does catch the signal and
gets to execute its last 'echo' commands.
The third alternative gives the same results as with 2.

Description:

If the execution of "bar" in "foo" is changed to ". bar" (instead of "bar")
the behaviour is again different. With 1, neither "foo" nor "bar" catch
the signal. With 2, "bar" catches the signal, "foo" doesn't but still
gets to finish its remaining lines. With 3 it's the same as with 2.
Change the execution of "bar" to "ksh bar" and the behaviour changes once
again. Respectively, "bar" catches the signal and "foo" is killed,
both "bar" and "foo" catch the signal and finish al the way and the same
thing happens with the third alternative.
In Unix both "foo" and "bar" catch the signals and finish the rest of their
scripts, no matter how "foo" is executed.

------------------ Amoeba Modification Request #183 ------------

```
Date:         Tue Aug  6 15:30:27 MET DST 1991
Reporter:     raymond@cs.vu.nl (Raymond Michiels, 020-5483500)
Category:     Bug
Module:       stdio
Severity:     Minor
Summary:      printf isn't line buffered
Handler:      versto
Status:       Open
```

Repeat By:

create two threads and let each one of them do a printf().

Description:

The output looks like:

```
     nseg = 12
     type = 33554704
     nsskeigp p=e d7.

     ttyyppee  ==  3333555545720342

     sskaivpipnegd .t
     ot y0p.e0 .=0 13.3.5.5
```

instead of something like:

```
     nseg = 12
     type = 33554704
     nseg = 7.
     skipped.

     type = 33554704
     type = 33555232

     skipped.
     saving to 0.0.01...

     type = 3355
```

------------------ Amoeba Modification Request #189 ------------

```
Date:        Tue Dec 10 15:05:56 MET 1991
Reporter:    sater@cs.vu.nl (Hans van Staveren,kamer S414,telef. 6219)
Category:    Bug
Module:      yap (and possibly others)
Severity:    Important
Summary:     generates infinite stream of broadcasts
Handler:     <to be assigned>
Status:      Open
```

Repeat By:

Start up yap under Amoeba, with output to a Unix window.
Kill the Unix window. Watch the broadcasts fly.

Description:

Yap assumes that when a read fails it could be a temporary condition.
Comments in the source say something about interrupts, ˆZ.
It just tries again. Under Unix this does not harm since it will be
killed by a SIGHUP anyhow, but under Amoeba it generates a stream

of broadcast locates to find the nonexisting server.

Either SIGHUP should be implemented or yap should be changed.

------------------ Amoeba Modification Request #195 ------------

Date: Tue Mar 31 16:43:28 MET DST 1992
Reporter: philip@cs.vu.nl (Philip Homburg,kamer R446,telef. 3546)
Category: Bug
Module: session server
Severity: Crucial before next release
Summary: crashme hangs session server.
Handler: <to be assigned>
Status: Open

Repeat By:

```
crashme-1.8 1000 1 1000 0:5:0 100
```

Description:

None of the functions related to the session server work.
Here is the part of ‘‘kstat -c nano’’ that contains the session server:

```
40:57:B6:F9:C1:C0:B:0 (pid 1) : 10 thread
27(0): tid 294 getreq ( ) timer: -1 retrial 10 loctime 5200 scnt 0 rcnt 29768
    ctid 295 cident 1 cadr: 40:57:B6:F9:C1:C0:B:0 rtime 6
    port: pub ?i????
24(1): tid 504 getrep ( doop ) timer: 6500 retrial 10 loctime 5000 scnt 0 rcnt 0
    ctid 59216 cident 0 cadr: CE:8D:7:2A:F8:68:EB:0 rtime 6
    port: pub ?A????
    cache: ??G?*? sid 1 saddr: 9A:93:67:F5:B2:E5:C9:0 rtime 100
23(2): tid 429 idle ( doop ) timer: -1 retrial 10 loctime 5200 scnt 16 rcnt 0
    ctid 2280 cident 4 cadr: 21:61:AE:D2:65:C3:3F:0 rtime 100
    port: pub ?A????
25(3): tid 0 idle ( doop sigtrans ) timer: -1 retrial 10 loctime 5200 scnt 16 rcnt 16
    ctid 19 cident 0 cadr: 35:AF:84:CD:C0:82:3:0 rtime 100
    port: pub ?A????
26(4): tid 0 getreq ( ) timer: -1 retrial -1 loctime 5200 scnt 0 rcnt 6000
    ctid 0 cident 0 cadr: 0:0:0:0:0:0:0:0 rtime 0
    port: pub ?A????
28(5): tid 441 idle ( ) timer: -1 retrial 10 loctime 5200 scnt 0 rcnt 36
29(6): tid 0 getreq ( ) timer: -1 retrial 0 loctime 5200 scnt 0 rcnt 0
    ctid 0 cident 0 cadr: 0:0:0:0:0:0:0:0 rtime 0
    port: pub h:????
30(7): tid 0 getreq ( ) timer: -1 retrial 0 loctime 5200 scnt 0 rcnt 0
    ctid 0 cident 0 cadr: 0:0:0:0:0:0:0:0 rtime 0
    port: pub h:????
31(8): tid 0 getreq ( ) timer: -1 retrial 0 loctime 5200 scnt 0 rcnt 29768
    ctid 0 cident 0 cadr: 0:0:0:0:0:0:0:0 rtime 0
    port: pub ?i????
32(9): tid 332 getreq ( ) timer: -1 retrial 10 loctime 5200 scnt 0 rcnt 6000
    ctid 2145 cident 2 cadr: 21:61:AE:D2:65:C3:3F:0 rtime 100
    port: pub ?A????
```

------------------ Amoeba Modification Request #198 ------------

Date:	Mon May 18 11:41:26 MET DST 1992
Reporter:	gregor@cs.vu.nl (Greg Sharp)
Category:	Bug
Module:	group communication
Severity:	Important
Summary:	known problems
Handler:	kaashoek
Status:	Open

Description:

The group communication is still young. There are some known bugs which should be taken into account:

1. The timers in the group code are not dynamically tuned. This does not usually lead to difficulties.

2. Recovery needs more extensive testing.

------------------ Amoeba Modification Request #203 ------------

Date:	Thu Nov 4 08:47:59 MET 1993
Reporter:	dholmes@ee.uts.edu.au (David Holmes)
Category:	Bug
Module:	ttn
Severity:	Minor
Summary:	ttn filches output
Handler:	<to be assigned>
Status:	Open

Repeat By:

There seems to be a strange glitch when you logout of a ttn session.
The next command entered at the amoeba% prompt gets a character

```
stolen from it. eg. dir -> dr: not found
                 1234 -> 234: not found
```

Description:

See above.

------------------ Amoeba Modification Request #205 ------------

Date:	Mon Dec 20 11:30:12 MET 1993
Reporter:	gregor@cs.vu.nl (Greg Sharp,kamer S414,telef. 6219)
Category:	Bug
Module:	/bin/sh
Severity:	Important
Summary:	commands in grave accents may not be parsed correctly
Handler:	<to be assigned>
Status:	Open

Repeat By:

```
D=xyz
X=‘echo ${D} | sed "s/))/1./"‘
echo $X
```

should print . but prints the value of D

Description:

When you have grave accents the shell should parse the input twice and thus do the substitution of $D later. However we had to fiddle with the shell to get commands in grave accents to execute and it probably means we only parse the command once.

```
------------------  Amoeba Modification Request #206 ------------

Date:        Tue Dec 21 17:26:17 MET 1993
Reporter:    gregor@cs.vu.nl (Greg Sharp,kamer S414,telef. 6219)
Category:    Bug
Module:      tty server
Severity:    Important
Summary:     VMIN and VTIME don’t work
Handler:     <to be assigned>
Status:      Open

Repeat By:

Set VMIN and VTIME to zero using tios_setattr.  Then read from the
tty.

Description:

You should get a non-blocking read but in fact it blocks.  The VMIN and
VTIME stuff isn’t properly implemented.
```

```
------------------  Amoeba Modification Request #209 ------------

Date:        Wed Mar  9 11:27:16 MET 1994
Reporter:    gregor@cs.vu.nl (Greg Sharp,kamer S414,telef. 6219)
Category:    Bug
Module:      AIL
Severity:    Important
Summary:     AIL ails
Handler:     <to be assigned>
Status:      Open

Repeat By:

Create a stub with as argument an array of 10000 longs.

Description:

The above produces a 40 kB array which cannot fit in a trans() buffer
but AIL tries it anyway.
```

```
------------------  Amoeba Modification Request #210 ------------

Date:        Wed Mar  9 11:30:16 MET 1994
Reporter:    gregor@cs.vu.nl (Greg Sharp,kamer S414,telef. 6219)
Category:    Bug
Module:      AIL
Severity:    Important
Summary:     AIL can’t handle floating-point
Handler:     <to be assigned>
Status:      Open

Repeat By:

Write a stub description with a floating-point argument.

Description:

AIL cannot manipulate floating-point objects.  It should perhaps support
IEEE FP in a machine independent way.


------------------  Amoeba Modification Request #211 ------------

Date:        Fri Oct  6 16:13:28 MET 1995
Reporter:    gregor@cs.vu.nl (Greg Sharp)
Category:    Bug
Module:      AIL
Severity:    Minor
Summary:     AIL filename truncation is inconsistent
Handler:     <to be assigned>
Status:      Open

Repeat By:

Get AIL to generate an include filename longer than 14 characters,
plus C code which includes the generated include file.  Then run AIL
with the -f13 option to truncate filenames to 13 characters.

Description:

AIL will correctly produce an include file whose name is 13 characters long.
However, the #include in the generated C source will use the untruncated
name and thus not find the include file.
```

```
------------------  Amoeba Modification Request #212 ------------

Date:        Fri Mar 15 14:15:45 MET 1996
Reporter:    gregor@cs.vu.nl (Greg Sharp)
Category:    Bug
Module:      386 Kernel
Severity:    Important
Summary:     The 386 kernel made with the GNU compiler is buggy
Handler:     <to be assigned>
Status:      Open
```

Repeat By:

Compile the Amoeba i80x86 kernel with the GNU compiler.

Description:

The Amoeba i80x86 kernel when compiled with the GNU compiler seems to make errors in memory mapping. The GNU compiler may be "over-optimising" some part of the kernel.

The workaround is to compile the kernel with the ACK compiler.