

Chapter 15

- 15.1 If the write end of the pipe is never closed, the reader never sees an end of file. The pager program blocks forever reading from its standard input.
- 15.2 The parent terminates right after writing the last line to the pipe. The read end of the pipe is automatically closed when the parent terminates. But the parent is probably running ahead of the child by one pipe buffer, since the child (the pager program) is waiting for us to look at a page of output. If we're running a shell, such as the Korn shell, with interactive command-line editing enabled, the shell probably changes the terminal mode when our parent terminates and the shell prints a prompt. This undoubtedly interferes with the pager program, which has also modified the terminal mode. (Most pager programs set the terminal to noncanonical mode when awaiting input to proceed to the next page.)
- 15.3 The `popen` function returns a file pointer because the shell is executed. But the shell can't execute the nonexistent command, so it prints

```
sh: line 1: ./a.out: No such file or directory
```

on the standard error and terminates with an exit status of 127. `pclose` returns the termination status of the command as it is returned by `waitpid`.

- 15.4 When the parent terminates, look at its termination status with the shell. For the Bourne shell, Bourne-again shell, and Korn shell, the command is `echo $?`. The number printed is 128 plus the signal number.
- 15.5 First add the declaration

```
FILE *fpin, *fpout;
```

Then use `fdopen` to associate the pipe descriptors with a standard I/O stream, and set the streams to be line buffered. Do this before the `while` loop that reads from standard input:

```
if ((fpin = fdopen(fd2[0], "r")) == NULL)
    err_sys("fdopen error");
if ((fpout = fdopen(fd1[1], "w")) == NULL)
    err_sys("fdopen error");
if (setvbuf(fpin, NULL, _IOLBF, 0) < 0)
    err_sys("setvbuf error");
if (setvbuf(fpout, NULL, _IOLBF, 0) < 0)
    err_sys("setvbuf error");
```

The write and read in the `while` loop are replaced with

```
if (fputs(line, fpout) == EOF)
    err_sys("fputs error to pipe");
if (fgets(line, MAXLINE, fpin) == NULL) {
    err_msg("child closed pipe");
    break;
}
```

- 15.6** The `system` function calls `wait`, and the first child to terminate is the child generated by `popen`. Since that's not the child that `system` created, it calls `wait` again and blocks until the `sleep` is done. Then `system` returns. When `pclose` calls `wait`, an error is returned, since there are no more children to wait for. Then `pclose` returns an error.
- 15.7** The `select` function indicates that the descriptor is readable. When we call `read` after all the data has been read, it returns 0 to indicate the end of file. But with `poll` (assuming a STREAMS-based pipe), the `POLLHUP` event is returned, and this event may be returned while there is still data to be read. Once we have read all the data, however, `read` returns 0 to indicate the end of file. After all the data has been read, the `POLLIN` event is not returned, even though we need to issue a `read` to receive the end-of-file notification (the return of 0).
- With an output descriptor that refers to a pipe that has been closed by the reader, `select` indicates that the descriptor is writable. But when we call `write`, the `SIGPIPE` signal is generated. If we either ignore this signal or return from its signal handler, `write` returns an error of `EPIPE`. With `poll`, however, if the pipe is STREAMS based, `poll` returns with a `POLLHUP` event for the descriptor.
- 15.8** Anything written by the child to standard error appears wherever the parent's standard error would appear. To send standard error back to the parent, include the shell redirection `2>&1` in the `cmdstring`.
- 15.9** The `popen` function forks a child, and the child executes the shell. The shell in turn calls `fork`, and the child of the shell executes the command string. When `cmdstring` terminates, the shell is waiting for this to happen. The shell then exits, which is what the `waitpid` in `pclose` is waiting for.
- 15.10** The trick is to open the FIFO twice: once for reading and once for writing. We never use the descriptor that is opened for writing, but leaving that descriptor open prevents an end of file from being generated when the number of clients goes from 1 to 0. Opening the FIFO twice requires some care, as a nonblocking open is required. We have to do a nonblocking, read-only open first, followed by a blocking open for write-only. (If we tried a nonblocking open for write-only first, it would return an error.) We then turn off nonblocking for the read descriptor. Figure C.18 shows the code for this.

```
#include "apue.h"
#include <fcntl.h>

#define FIFO    "temp.fifo"

int
main(void)
{
    int    fdread, fdwrite;

    unlink(FIFO);
    if (mkfifo(FIFO, FILE_MODE) < 0)
        err_sys("mkfifo error");
    if ((fdread = open(FIFO, O_RDONLY | O_NONBLOCK)) < 0)
```

```

    err_sys("open error for reading");
    if ((fdwrite = open(FIFO, O_WRONLY)) < 0)
        err_sys("open error for writing");
    clr_fl(fdread, O_NONBLOCK);
    exit(0);
}

```

Figure C.18 Opening a FIFO for reading and writing, without blocking

15.11 Randomly reading a message from an active queue would interfere with the client-server protocol, as either a client request or a server's response would be lost. To read the queue, all that is needed is for the process to know the identifier for the queue and for the queue to allow world-read access.

15.13 We never store actual addresses in a shared memory segment, since it's possible for the server and all the clients to attach the segment at different addresses. Instead, when a linked list is built in a shared memory segment, the list pointers should be stored as offsets to other objects in the shared memory segment. These offsets are formed by subtracting the start of the shared memory segment from the actual address of the object.

15.14 Figure C.19 shows the relevant events.

Parent i set to	Child i set to	Shared value set to	update returns	Comment
0	1	0		initialized by mmap child runs first, then is blocked parent runs
		1	0	then parent is blocked child resumes
		2	1	then child is blocked parent resumes
2	3	3	2	then parent is blocked
		4	3	then child is blocked parent resumes
4	5			

Figure C.19 Alternation between parent and child in Figure 15.33

Chapter 16

16.1 Figure C.20 shows a program that prints the system's byte order.

16.3 For each endpoint we will be listening on, we need to bind the proper address and record an entry in an `fd_set` structure corresponding to each file descriptor.

```

#include <stdio.h>
#include <stdlib.h>
#include <inttypes.h>

int
main(void)
{
    uint32_t i;
    unsigned char *cp;

    i = 0x04030201;
    cp = (unsigned char *)&i;
    if (*cp == 1)
        printf("little-endian\n");
    else if (*cp == 4)
        printf("big-endian\n");
    else
        printf("who knows?\n");
    exit(0);
}

```

Figure C.20 Determine byte order on system

We will use `select` to wait for connect requests to arrive on multiple endpoints. Recall from Section 16.4 that a passive endpoint will appear to be readable when a connect request arrives on it. When a connect request does arrive, we will accept the request and process it as before.

- 16.5 In the main procedure, we need to arrange to catch `SIGCHLD` by calling our signal function (Figure 10.18), which will use `sigaction` to install the handler specifying the restartable system call option. Next, we need to remove the call to `waitpid` from our `serve` function. After forking the child to service the request, the parent closes the new file descriptor and resumes listening for additional connect requests. Finally, we need a signal handler for `SIGCHLD`, as follows:

```

void
sigchld(int signo)
{
    while (waitpid((pid_t)-1, NULL, WNOHANG) > 0)
        ;
}

```

- 16.6 To enable asynchronous socket I/O, we need to establish socket ownership using the `F_SETOWN` `fcntl` command, and then enable asynchronous signaling using the `FIOASYNC` `ioctl` command. To disable asynchronous socket I/O, we simply need to disable asynchronous signaling. The reason we mix `fcntl` and `ioctl` commands is to find the methods that are most portable. The code is shown in Figure C.21.

```

#include "apue.h"
#include <errno.h>
#include <fcntl.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#if defined(BSD) || defined(MACOS) || defined(SOLARIS)
#include <sys/filio.h>
#endif

int
setasync(int sockfd)
{
    int n;

    if (fcntl(sockfd, F_SETOWN, getpid()) < 0)
        return(-1);
    n = 1;
    if (ioctl(sockfd, FIOASYNC, &n) < 0)
        return(-1);
    return(0);
}

int
clrasync(int sockfd)
{
    int n;

    n = 0;
    if (ioctl(sockfd, FIOASYNC, &n) < 0)
        return(-1);
    return(0);
}

```

Figure C.21 Enable and disable asynchronous socket I/O

Chapter 17

- 17.3 A *declaration* specifies the attributes (such as the data type) of a set of identifiers. If the declaration also causes storage to be allocated, it is called a *definition*.

In the `opend.h` header, we declare the three global variables with the `extern` storage class. These declarations do not cause storage to be allocated for the variables. In the `main.c` file, we define the three global variables. Sometimes, we'll also initialize a global variable when we define it, but we typically let the C default apply.

- 17.5 Both `select` and `poll` return the number of ready descriptors as the value of the function. The loop that goes through the `client` array can terminate when the number of ready descriptors have been processed.

Chapter 18

- 18.1 Note that you have to terminate the `reset` command with a line feed character, not a return, since the terminal is in noncanonical mode.
- 18.2 It builds a table for each of the 128 characters and sets the high-order bit (the parity bit) according to the user's specification. It then uses 8-bit I/O, handling the parity generation itself.
- 18.3 If you happen to be on a windowing terminal, you don't need to log in twice. You can do this experiment between two separate windows. Under Solaris, execute `stty -a` with standard input redirected from the terminal window running `vi`. This shows that `vi` sets `MIN` to 1 and `TIME` to 1. A call to `read` will wait for at least one character to be typed, but after that character is entered, `read` waits only one-tenth of a second for additional characters before returning.

Chapter 19

- 19.1 Both servers, `telnetd` and `rlogind`, run with superuser privileges, so their calls to `chown` and `chmod` succeed.
- 19.3 Execute `pty -n stty -a` to prevent the slave's `termios` structure and `winsize` structure from being initialized.
- 19.5 Unfortunately, the `F_SETFL` command of `fcntl` doesn't allow the read-write status to be changed.
- 19.6 There are three process groups: (1) the login shell, (2) the `pty` parent and child, and (3) the `cat` process. The first two process groups constitute a session with the login shell as the session leader. The second session contains only the `cat` process. The first process group (the login shell) is a background process group, and the other two are foreground process groups.
- 19.7 First, `cat` terminates when it receives the end of file from its line discipline. This causes the PTY slave to terminate, which causes the PTY master to terminate. This in turn generates an end of file for the `pty` parent that's reading from the PTY master. The parent sends `SIGTERM` to the child, so the child terminates next. (The child doesn't catch this signal.) Finally, the parent calls `exit(0)` at the end of the `main` function.

The relevant output from the program shown in Figure 8.29 is

```

cat      e = 270, chars = 274, stat = 0:
pty      e = 262, chars = 40, stat = 15: F    X
pty      e = 288, chars = 188, stat = 0:
```

- 19.8 This can be done with the shell's `echo` command and the `date(1)` command, all in a subshell:

```
#!/bin/sh
( echo "Script started on " `date`;
  pty "${SHELL:-/bin/sh}";
  echo "Script done on " `date` ) | tee typescript
```

- 19.9 The line discipline above the PTY slave has echo enabled, so whatever pty reads on its standard input and writes to the PTY master gets echoed by default. This echoing is done by the line discipline module above the slave even though the program (ttyname) never reads the data.

Chapter 20

- 20.1 Our conservative locking in `_db_dodelete` is to avoid race conditions with `db_nextrec`. If the call to `_db_writedat` were not protected with a write lock, it would be possible to erase the data record while `db_nextrec` was reading that data record: `db_nextrec` would read an index record, determine that it was not blank, and then read the data record, which could be erased by `_db_dodelete` between the calls to `_db_readidx` and `_db_readdat` in `db_nextrec`.
- 20.2 Assume that `db_nextrec` calls `_db_readidx`, which reads the key into the index buffer for the process. This process is then stopped by the kernel, and another process runs. This other process calls `db_delete`, and the record being read by the other process is deleted. Both its key and its data are rewritten in the two files as all blanks. The first process resumes and calls `_db_readdat` (from `db_nextrec`) and reads the all-blank data record. The read lock by `db_nextrec` allows it to do the read of the index record, followed by the read of the data record, as an atomic operation (with regard to other cooperating processes using the same database).
- 20.3 With mandatory locking, other readers and writers are affected. Other reads and writes are blocked by the kernel until the locks placed by `_db_writeidx` and `_db_writedat` are removed.
- 20.5 By writing the data record before the index record, we protect ourselves from generating a corrupt record if the process should be killed in between the two writes. If the process were to write the index record first, but be killed before writing the data record, then we'd have a valid index record that pointed to invalid data.

Chapter 21

- 21.5 Here are some hints. There are two places to check for queued jobs: the printer spooling daemon's queue and the network printer's internal queue. Take care to prevent one user from being able to cancel someone else's print job. Of course, the superuser should be able to cancel any job.

Index

The function subentries labeled “definition of” point to where the function prototype appears and, when applicable, to the source code for the function. Functions defined in the text that are used in later examples, such as the `set_fl` function in Figure 3.11, are included in this index. The definitions of external functions that are part of the larger examples (Chapters 17, 19, 20, and 21) are also included in this index, to help in going through these larger examples. Also, significant functions and constants that occur in any of the examples in the text, such as `select` and `poll`, are also included in this index. Trivial functions that occur in almost every example, such as `exit`, are not referenced when they occur in examples.

- `#!`, *see* interpreter files
- `.`, *see* current directory
- `..`, *see* parent directory
- 2.9BSD, 216
- 386BSD, xxvii, 34–35
- 4.1BSD, 487
- 4.2BSD, 18, 112, 119–120, 167, 428–429, 474, 481, 483, 487, 545
- 4.3BSD, xxvii, 33–34, 36, 183, 240, 248, 265, 442, 497, 699, 846, 888
 - Reno, xxvii, 34, 72
 - Tahoe, xxvii, 34, 888
- 4.4BSD, xxii, xxvii, 21, 34, 105, 112, 119, 139, 216, 462, 497, 545, 699, 710, 888
- a2ps program, 805
- abort function, 180, 218, 223, 253, 256, 289, 293–295, 340–342, 353, 407, 848, 870
 - definition of, 340–341
- absolute pathname, 5, 7, 43, 49, 126, 131, 242, 859
- accept function, 138, 306, 411, 563–564, 570, 572, 597, 599–600, 780
 - definition of, 563
- access function, 95–97, 113, 116, 306
 - definition of, 95
- accounting
 - login, 170–171
 - process, 250–256
- acct function, 250

- acct structure, 251, 254
acctcom program, 250
accton program, 250–251, 255
ACOMPAT constant, 251
ACORE constant, 251, 254–255
acstime_r function, 402
add_job function, 783, 790
add_option function, 794, 797
addressing, socket, 549–561
addrinfo structure, 555–559, 569, 571, 573, 576, 578, 779, 782, 796
add_worker function, 787, 791
adjustment on exit, semaphore, 532–533
Adobe Systems, 885
advisory record locking, 455
AES (Application Environment Specification), 32
AEXPND constant, 251
AF_INET constant, 547, 551–552, 554, 557, 559–560
AF_INET6 constant, 551–552, 557
AF_IPX constant, 546
AF_LOCAL constant, 546
AFORK constant, 251–252, 254
AF_UNIX constant, 546, 557, 595–598, 600–601
AF_UNSPEC constant, 546, 557
agetty program, 265
Aho, A. V., 243, 885
AI_ALL constant, 559
AI_CANONNAME constant, 559, 571, 574, 578
AI_NUMERICHOST constant, 559
AI_NUMERICSERV constant, 559
aio_error function, 306
<aio.h> header, 30
aio_return function, 306
aio_suspend function, 306, 411
AI_PASSIVE constant, 559
AI_V4MAPPED constant, 556, 559
AIX, 36
alarm function, 289, 293, 306–307, 310, 313–318, 331, 348–349, 354, 575–576, 867
 definition of, 313
alloca function, 192
already_running function, definition of, 433
ALTWERASE constant, 636, 642, 645
American National Standards Institute, *see* ANSI
Andrade, J. M., 521, 885
ANSI (American National Standards Institute), 25
ANSI C, xxvi–xxvii
Apple Computer, xxii
Application Environment Specification, *see* AES
apue_db.h header, 711, 719, 723, 727
apue.h header, 6, 9–10, 229, 299, 449–450, 597, 721, 843–846
Architecture, UNIX, 1–2
argc variable, 778
ARG_MAX constant, 39, 42, 46, 48, 233, 404
arguments, command-line, 185
argv variable, 774
Arnold J. Q., 188, 885
<arpa/inet.h> header, 29, 550
asctime function, 175, 402
 definition of, 175
<assert.h> header, 27
ASU constant, 251, 254
asynchronous I/O, 473, 481–482
asynchronous socket I/O, 582–583
at program, 431
atexit function, 42, 182, 184, 207, 218, 365, 696, 863
 definition of, 182
ATEXIT_MAX constant, 40, 42, 48, 51
atol function, 781
atomic operation, 39, 43, 57, 61, 74–75, 77, 109, 139, 333, 340, 448, 515, 528, 530, 532, 883
AT&T, xix, 5, 33–34, 159, 311, 460, 462, 479, 885–886
automatic variables, 187, 197, 199, 201, 207
avoidance, deadlock, 373
awk program, 44–46, 243–246, 514, 887
AXSIG constant, 251, 254–255
B0 constant, 652
B110 constant, 652
B115200 constant, 652
B1200 constant, 652
B134 constant, 652
B150 constant, 652
B1800 constant, 652
B19200 constant, 652
B200 constant, 652
B2400 constant, 652
B300 constant, 652
B38400 constant, 652
B4800 constant, 652
B50 constant, 652
B57600 constant, 652
B600 constant, 652
B75 constant, 652
B9600 constant, 652
Bach, M. J., xix, xxviii, 70, 77, 104, 108, 211, 461, 855, 886
background process group, 272, 275, 277, 279, 281–282, 284–285, 296–297, 344, 349, 882
backoff, exponential, 562

- Barkley, R. E., 886
- basename function, 402
- bash program, 81, 158, 250
- .bash_login file, 265
- .bash_profile file, 265
- Bass, J., 445
- baud rate, terminal I/O, 652–653
- Berkeley Software Distribution, *see* BSD
- bibliography, alphabetical, 885–890
- big-endian byte order, 549
- bind function, 306, 560, 564, 580–581, 596–598, 600–601
- definition of, 560
- /bin/false program, 163
- /bin/true program, 163
- <bits/signum.h> header, 290
- block special file, 89, 128–129
- Bolsky, M. I., 510, 886
- Bostic, K., xxviii, 33, 70, 104, 108, 461, 487, 888
- Keith, 211, 218
- Bourne, S. R., 3
- Bourne shell, 3, 52, 86, 158, 192, 204, 265, 275, 278, 346, 457, 504, 510, 662, 877, 887
- Bourne-again shell, 3, 51–52, 81, 86, 192, 204, 265, 275, 510
- BREAK character, 637, 642, 645, 648, 650, 654, 668
- BRKINT constant, 635, 645, 648, 666–668
- BS0 constant, 645
- BS1 constant, 645
- BSD (Berkeley Software Distribution), 34, 62, 83, 262, 265–266, 268–269, 271, 273–274, 442, 473, 481–482, 493, 552–553, 595, 683, 685–686, 689, 691, 699, 706–707
- BSD Networking Release 1.0, xxvii, 34
- BSD Networking Release 2.0, xxvii, 34
- BSD/386, xxvii
- BSDLY constant, 637, 644–645, 649
- bss segment, 187
- buf_args function, 618–620, 628–629, 845
- definition of, 619
- buffer cache, 77
- buffering, standard I/O, 135–137, 213, 217, 247, 342, 513–514, 680, 718
- BUFSIZ constant, 49, 137, 202
- build_qonstart function, 780, 785
- BUS_ADRALN constant, 327
- BUS_ADRERR constant, 327
- BUS_OBJERR constant, 327
- byte order
- big-endian, 549
- little-endian, 549
- byte ordering, 549–550
- C, ANSI, xxvi–xxvii
- ISO, 25–26, 887
- C shell, 3, 52, 204, 265, 275, 510
- c99 program, 56, 67
- cache
- buffer, 77
- page, 77
- caddr_t data type, 57
- CAE (Common Application Environment), 32
- calendar time, 20, 24, 57, 117, 173–175, 246, 251–252
- calloc function, 189–190, 207, 467, 506, 726, 863
- definition of, 189
- cancellation point, 410–411
- canonical mode, terminal I/O, 660–663
- Carges, M. T., 521, 885
- cat program, 85, 104, 114, 276, 279, 699, 714, 882
- catclose function, 412
- catgets function, 402, 412
- catopen function, 412
- CBAUDEX constant, 635, 645
- cbreak terminal mode, 632, 664, 668, 673
- cc program, 6, 55, 189
- CCAR_OFLOW constant, 635, 645, 649
- cc_t data type, 634
- CCTS_OFLOW constant, 635, 645
- cc(1) program, 6
- cd program, 126
- CDSR_OFLOW constant, 635, 645
- CDTR_IFLOW constant, 635, 645
- cfgetispeed function, 306, 637, 652
- definition of, 652
- cfgetospeed function, 306, 637, 652
- definition of, 652
- cfsetispeed function, 306, 637, 652
- definition of, 652
- cfsetospeed function, 306, 637, 652
- definition of, 652
- character special file, 89, 128–129, 461, 466, 659
- CHAR_BIT constant, 38
- CHARCLASS_NAME_MAX constant, 39, 48
- CHAR_MAX constant, 38
- CHAR_MIN constant, 38
- chdir function, 7, 113, 125–127, 131, 204, 264, 306, 427, 860
- definition of, 125
- Chen, D., 886
- CHILD_MAX constant, 39, 42, 48, 215
- chmod function, 99–101, 113, 117, 306, 520, 600–601, 687, 689–690, 882
- definition of, 99
- chmod program, 93, 521

- chown function, 54, 102–103, 112–113, 117, 264, 306, 520, 687, 689, 882
 - definition of, 102
- chroot function, 131, 439, 858, 871
- CIBAUEXT constant, 635, 645
- CIGNORE constant, 635, 645
- Clark, J. J., xxviii
- CLD_CONTINUED constant, 327
- CLD_DUMPED constant, 327
- CLD_EXITED constant, 327
- CLD_KILLED constant, 327
- CLD_STOPPED constant, 327
- CLD_TRAPPED constant, 327
- clearenv function, 194
- clearerr function, 141
 - definition of, 141
- cli_args function, 618–620, 628
 - definition of, 620
- cli_conn function, 592–593, 600, 621, 626, 845
 - definition of, 592, 594, 600
- client_add function, 623, 625–627
 - definition of, 623
- client_alloc function, 623
 - definition of, 622
- client_cleanup function, 787, 792
- client_del function, 625, 627
 - definition of, 623
- client-server
 - model, 439, 541–543
- client_thread function, 787
- CLOCAL constant, 294, 635, 645
- clock function, 57
- clock tick, 20, 42, 48, 57, 251–252, 257
- clock_gettime function, 306
- clock_nanosleep function, 411
- CLOCKS_PER_SEC constant, 57
- clock_t data type, 20, 57, 257
- clone device, STREAMS, 683
- clone function, 211, 360, 416
- close function, 8, 51, 59, 63, 77, 115, 118, 306, 411, 427, 433, 452, 461, 493, 499–501, 506, 511–512, 515, 521–522, 539–540, 543, 547–548, 564, 571, 573, 581, 587–588, 592–594, 598–599, 601, 616–617, 619, 625, 627–628, 684–685, 688, 690, 693, 704–705
 - definition of, 63
- closedir function, 5, 7, 120–125, 412, 658, 858
 - definition of, 120
- closelog function, 412, 430
 - definition of, 430
- close-on-exec flag, 76, 79, 234, 452
- clrasync function, definition of, 881
- clr_fl function, 81, 442–443, 844, 879
- clri program, 114
- msgcred structure, 610–613
- MSG_DATA function, 607–608, 610, 612, 614
 - definition of, 607
- MSG_FIRSTHDR function, 607, 614
 - definition of, 607
- msgshdr structure, 607–609, 611, 613
- MSG_LEN function, 607–609, 611, 613
 - definition of, 607
- MSG_NXTHDR function, 607, 612, 614
 - definition of, 607
- CMSPAR constant, 637, 645, 650
- codes, option, 31
- COLL_WEIGHTS_MAX constant, 39, 42, 48
- COLUMNS environment variable, 193
- Comer, D. E., 710, 886
- command-line arguments, 185
- Common Application Environment, *see* CAE
- Common Open Software Environment, *see* COSE
- communication, network printer, 753–805
- <complex.h> header, 27
- comp_t data type, 57
- Computing Science Research Group, *see* CSRG
- cond_signal function, 385
- connect function, 306, 411, 561–563, 565–566, 577, 597, 601
 - definition of, 561
- connection establishment, 561–565
- connect_retry function, 569, 763, 797
 - definition of, 562
- connld STREAMS module, 518, 590, 592, 600
- controlling
 - process, 272, 294
 - terminal, 61, 215, 234, 251, 268, 271–274, 276, 278–279, 281, 284, 286–287, 294, 296–297, 349, 423–425, 428, 439, 463, 468, 640, 645, 651, 654, 660, 662, 676, 683, 685, 689, 691–692, 846, 890
- cooked terminal mode, 632
- cooperating processes, 455, 717, 883
- Coordinated Universal Time, *see* UTC
- coprocesses, 510–514, 680, 701
- copy-on-write, 211, 417
- core dump, 70, 870
- core file, 104, 116, 256, 291, 293, 296, 307, 340, 641, 663, 857, 863, 865
- COSE (Common Open Software Environment), 32
- cp program, 131, 490
- cpio program, 117, 131–132, 858–859
- <cpio.h> header, 30
- CR terminal character, 638, 640, 663

- CR0 constant, 645
 - CR1 constant, 645
 - CR2 constant, 645
 - CR3 constant, 645
 - CRDLY constant, 637, 644–645, 649
 - CREAD constant, 635, 646
 - creat function, 59, 62–63, 65, 75, 85, 95, 97, 110, 113, 117, 139, 306, 411, 451, 592, 857, 860
 - definition of, 62
 - creation mask, file mode, 97–98, 119, 131, 215, 234, 425
 - cron program, 354, 425, 430–432, 434, 868
 - CRTSCTS constant, 635, 646
 - CRTS_IFLOW constant, 635, 646
 - CRTSXOFF constant, 635, 646
 - crypt function, 263, 273, 279–280, 402
 - crypt program, 273, 660
 - CS5 constant, 644, 646
 - CS6 constant, 644, 646
 - CS7 constant, 644, 646
 - CS8 constant, 644, 646, 666–668
 - .cshrc file, 265
 - CSIZE constant, 635, 644, 646, 666–667
 - csopen function, 615–616
 - definition of, 616, 621
 - CSRG (Computing Science Research Group), xx, xxii, 35
 - CSTOPB constant, 635, 646
 - ctermid function, 401, 412, 654, 660–661
 - definition of, 654
 - ctime function, 174–176, 402
 - definition of, 175
 - ctime_r function, 402
 - <ctype.h> header, 27
 - cu program, 473
 - cupsd program, 425, 757
 - curses library, 672–673, 887, 890
 - cuserid function, 257
 - uninitialized, 187
 - data transfer, 565–579
 - data types, primitive system, 56
 - database library, 709–752
 - coarse-grained locking, 718
 - concurrency, 718–719
 - fine-grained locking, 718
 - implementation, 712–715
 - performance, 747–752
 - source code, 719–747
 - database transactions, 889
 - Date, C. J., 719, 886
 - date functions, time and, 173–176
 - date program, 175, 178, 346, 862, 882
 - DATEMSK environment variable, 193
 - db library, 710, 889
 - DB structure, 722–724, 726–728, 731–734, 739, 742, 748
 - _db_alloc function, 723, 726–727
 - db_close function, 710–711, 715, 727
 - definition of, 710
 - db_delete function, 711, 718, 734–735, 737, 883
 - definition of, 711
 - _db_dodelete function, 734–735, 738, 742, 746–747, 752, 883
 - db_fetch function, 711, 713, 715, 718, 728, 733
 - definition of, 711
 - _db_find_and_lock function, 728–729, 733–734, 740–741, 743, 752
 - _db_findfree function, 741, 743–744, 747
 - _db_free function, 724, 727
 - DBHANDLE data type, 715
 - _db_hash function, 730, 752
 - DB_INSERT constant, 711, 715, 740
 - dbm library, 709–710, 890
 - dbm_clearerr function, 402
 - dbm_close function, 402, 412
 - dbm_delete function, 402, 412
 - dbm_error function, 402
 - dbm_fetch function, 402, 412
 - dbm_firstkey function, 402
 - dbm_nextkey function, 402, 412
 - dbm_open function, 402, 412
 - dbm_store function, 402, 412
 - db_nextrec function, 712, 715, 718, 735, 745, 747, 752, 883
 - definition of, 712
 - db_open function, 710–712, 715, 718, 721–723, 725–727, 747
 - definition of, 710
 - _db_readdat function, 728, 734, 883
 - _db_readidx function, 730–731, 746, 883
-
- daemon, 423–439
 - coding, 425–428
 - conventions, 434–439
 - error logging, 428–432
 - daemonize function, 425, 428, 439, 571, 573, 578, 624, 778, 844, 871–872
 - definition of, 426
 - Dang, X. T., 188, 887
 - Darwin, xxiii, 35
 - data, out-of-band, 581–582
 - data segment
 - initialized, 187

- `_db_readptr` function, 729, 731, 752
- DB_REPLACE constant, 711, 740
- `db_rewind` function, 712, 745, 747
 - definition of, 712
- DB_STORE constant, 711, 740
- `db_store` function, 711–712, 715, 718, 720, 735, 737, 740, 747, 750, 752
 - definition of, 711
- `_db_writedat` function, 735, 737–738, 741–742, 747, 752, 883
- `_db_writeidx` function, 484, 725, 738, 741–742, 747, 752, 883
- `_db_writeptr` function, 725, 739, 741–742
- `dcheck` program, 114
- `dd` program, 256
- deadlock, 216, 373, 450, 513, 680
 - avoidance, 373
 - record locking, 450
- delayed write, 77
- descriptor set, 475, 477, 493, 875
- `detachstate` attribute, 389–390
- `/dev/fb` device, 466
- `/dev/fd` device, 84–85, 132, 656
- `/dev/fd/0` device, 85
- `/dev/fd/1` device, 85, 132
- `/dev/fd/2` device, 85
- `devfs` file system, 129
- device number
 - major, 56–57, 127, 129, 659
 - minor, 56–57, 127, 129, 659
- device special file, 127–129
- device, STREAMS clone, 683
- `/dev/klog` device, 429
- `/dev/kmem` device, 65
- `/dev/log` device, 429, 439, 871
- `/dev/null` device, 69, 82, 279, 466
- `/dev/ptmx` device, 683–685, 689–690
- `/dev/pts` device, 689
- `/dev/pty` device, 686–687
- `/dev/stderr` device, 85, 658
- `/dev/stdin` device, 85, 658
- `/dev/stdout` device, 85, 658
- `dev_t` data type, 57, 127–128
- `/dev/tty` device, 273, 279, 287, 466, 654, 660, 705
- `/dev/zero` device, 538–540
- `df` program, 131, 858
- DIR structure, 7, 121, 260, 657
- directories
 - files and, 4–7
 - reading, 120–125
- directory, 4
 - file, 88
 - home, 2, 7, 125, 193, 264, 267
 - ownership, 95
 - parent, 4, 101, 116, 119
 - root, 4, 7, 24, 129, 131, 215, 234, 260, 858
 - working, 7, 13, 43, 49, 107, 125–126, 162, 193, 215, 234, 291, 426
- `dirent` structure, 5, 7, 121, 123, 657
- `<dirent.h>` header, 29, 121
- `dirname` function, 402
- DISCARD terminal character, 638, 640, 647
- `dlclose` function, 412
- `dlopen` function, 402
- `<dlfcn.h>` header, 30
- `dlopen` function, 412
- `do_driver` function, 696, 704
 - definition of, 704
- Dorward, S., 211, 889
- DOS, 55
- dot, *see* current directory
- dot-dot, *see* parent directory
- `drand48` function, 402
- DSUSP terminal character, 638, 640, 648
- `du` program, 104, 131, 857–858
- Duff, T., 84
- `dup` function, 51, 59, 70, 73, 76–77, 138, 153, 213, 306, 428, 452–453, 548, 855–856, 864
 - definition of, 76
- `dup2` function, 62, 76–77, 86, 138, 306, 501, 506, 512, 548, 573–574, 588, 617, 693, 704–705, 855
 - definition of, 76
- E2BIG error, 526
- EACCESS error, 14–15, 433–434, 447, 459, 861
- EAGAIN error, 16, 433–434, 442, 444, 447, 456–457, 459, 525, 531–532, 564, 582, 687
- EBADF error, 51
- EBADMSG error, 470
- EBUSY error, 16, 371, 380
- ECHILD error, 308, 326, 345, 508
- ECHO constant, 636, 646–647, 661, 665–667, 696, 844
- `echo` program, 185
- ECHOCTL constant, 636, 646
- ECHOE constant, 636, 646–647, 661, 696
- ECHOK constant, 636, 647, 661, 696
- ECHOKE constant, 636, 647
- ECHONL constant, 636, 647, 661, 696
- ECHOPRT constant, 636, 646–647
- `ecvt` function, 402
- `ed` program, 342, 344–345, 456–457
- EEXIST error, 112, 520

- EFBIG error, 868
- effective
 - group ID, 91–92, 94–95, 101, 103, 130, 167, 210, 214, 237, 241, 520, 543, 605
 - user ID, 91–92, 94–95, 99, 103, 117, 130, 210, 214, 235, 237–241, 257, 262, 264, 312, 354, 520, 524, 530, 535, 542–543, 593, 600, 605, 771, 861, 866
- efficiency
 - I/O, 68–70
 - standard I/O, 143–145
- EIDRM error, 524–526, 530–532
- EINPROGRESS error, 563
- EINTR error, 16, 246–247, 303–304, 313, 334, 345, 475, 480, 507–508, 525–526, 532, 575
- EINVAL error, 42, 47, 320, 361, 367, 464, 466, 505, 507, 665–667
- EIO error, 284, 297, 686
- Ellis, M., xviii
- ELOOP error, 113–114
- EMSGSIZE error, 566
- ENAMETOOLONG error, 62
- encrypt function, 402
- endgrent function, 167–168, 402, 412
 - definition of, 167
- endhostent function, 412, 553
 - definition of, 553
- endnetent function, 412, 554
 - definition of, 554
- endprotoent function, 412, 554
 - definition of, 554
- endpwent function, 164–165, 402, 412
 - definition of, 164
- endservent function, 412, 555
 - definition of, 555
- endspent function, 166
 - definition of, 166
- endutxent function, 402, 412
- ENFILE error, 16
- ENOBUFS error, 16
- ENODEV error, 466
- ENOENT error, 15, 405, 686, 711
- ENOLCK error, 16
- ENOMEM error, 16
- ENOMSG error, 526
- ENOSPC error, 16, 405
- ENOSR error, 16
- ENOSTR error, 466
- ENOTDIR error, 548
- ENOTTY error, 466, 643, 653
- environ variable, 185–186, 193, 195, 232, 236, 404–405, 409, 863
- environment list, 185–186, 215, 233, 262–264
- environment variable, 192–195
 - COLUMNS, 193
 - DATMSK, 193
 - HOME, 192–193, 264
 - IFS, 250
 - LANG, 41, 193
 - LC_ALL, 193
 - LC_COLLATE, 42, 193
 - LC_CTYPE, 193
 - LC_MESSAGES, 193
 - LC_MONETARY, 193
 - LC_NUMERIC, 193
 - LC_TIME, 193
 - LD_LIBRARY_PATH, 719
 - LINES, 193
 - LOGNAME, 193, 257, 264
 - MAILPATH, 192
 - MALLOC_OPTIONS, 870
 - MSGVERB, 193
 - NLSPATH, 193
 - PAGER, 501, 504–505
 - PATH, 93, 193, 232–233, 235, 242, 244, 247, 264–265
 - PWD, 193
 - SHELL, 193, 264, 701
 - TERM, 193, 263, 265
 - TMPDIR, 157–158, 193
 - TZ, 174, 176, 178, 193, 862
 - USER, 192, 264
- ENXIO error, 515
- EOF constant, 10, 141, 143–144, 154, 160, 507, 509, 512–513, 587, 624, 694, 861
- EOF terminal character, 638, 640, 646–647, 660, 663
- EOL terminal character, 638, 640, 647, 660, 663
- EOL2 terminal character, 638, 640, 647, 660, 663
- EPERM error, 238
- EPIPE error, 499, 878
- Epoch, 20, 22, 117, 171, 173–174, 600
- ERANGE error, 49
- ERASE terminal character, 638, 640, 646–647, 662–663
- ERASE2 terminal character, 638, 641
- err_dump function, 340, 733, 845–846
 - definition of, 848
- err_exit function, 845–846
 - definition of, 847
- err_msg function, 845–846
 - definition of, 848
- errno variable, 14–15, 24, 42, 49, 54, 62, 64, 77, 112–113, 134, 238, 246, 284, 290, 297, 303–304, 306–308, 312–313, 320, 326, 334, 345, 353, 356, 358, 406, 413, 431, 434, 442, 444, 447, 459, 475, 480, 499, 508, 515, 526, 548, 563–564, 566,

- 582, 643, 653, 711, 767, 782, 846, 854, 868
 <errno.h> header, 14, 16, 27
 error
 handling, 14–16
 logging, daemon, 428–432
 recovery, 16
 routines, standard, 846–851
 err_quit function, 7, 778, 845–846, 860
 definition of, 848
 err_ret function, 771, 845–846, 860
 definition of, 847
 err_sys function, 7, 24, 771, 845–846
 definition of, 847
 EPIPE error, 64, 548
 ESRCH error, 312
 /etc/gettydefs file, 265
 /etc/group file, 17–18, 161, 169–170
 /etc/hosts file, 170
 /etc/inittab file, 265
 /etc/master.passwd file, 169
 /etc/motd device, 466
 /etc/networks file, 170
 /etc/passwd file, 2, 92, 125, 161–162, 164, 166,
 169–170
 /etc/protocols file, 170
 /etc/pwd.db file, 169
 /etc/rc file, 173, 266
 /etc/services file, 170
 /etc/shadow file, 92, 169–170
 /etc/spwd.db file, 169
 /etc/syslog.conf file, 429
 /etc/termcap file, 672
 /etc/ttys file, 262
 ETIME error, 763, 767
 ETIMEDOUT error, 384
 EWOULDBLOCK error, 16, 442, 564, 582
 exec function, 10–12, 22, 39, 42, 78, 94, 113,
 116–117, 179, 183, 185, 206, 211, 215–216,
 231–240, 242–243, 245–246, 248, 250, 252,
 256, 259–260, 262–264, 266–267, 270, 280,
 300–301, 347, 416–417, 452, 489, 495, 500,
 503, 519, 541, 615–616, 620–621, 629, 676,
 678, 680, 682, 692, 704, 707, 863, 870, 886
 execl function, 231–233, 242–243, 247, 253,
 255–256, 260, 264, 345–346, 501, 506, 512,
 573, 588, 617, 702, 865
 definition of, 231
 execl function, 231–233, 235–236, 263, 306
 definition of, 231
 execlp function, 11–13, 19, 231–233, 235–236,
 245, 247, 260, 704, 865
 definition of, 231
 execv function, 231–233
 definition of, 231
 execve function, 231–233, 235, 306, 865
 definition of, 231
 execvp function, 231–233, 235, 695–696
 definition of, 231
 exercises, solutions to, 853–883
 _exit function, 180, 183, 217–221, 247, 259–260,
 306, 340, 342, 345, 354, 360, 407, 864, 867
 _Exit function, 180, 183, 218–219, 221, 306, 340,
 342, 360, 407
 definition of, 180
 _exit function, definition of, 180
 exit function, 7, 140, 144, 180–184, 207, 213,
 216–221, 228, 231, 246–247, 252–253,
 255–256, 260, 264, 305, 340–341, 360, 407,
 425, 504, 665, 696, 707, 771, 780, 793, 843,
 863–864, 882
 definition of, 180
 exit handler, 182
 expect program, 679, 703–705, 888
 exponential backoff, 562
 ext2 file system, 69, 82, 95, 119
 ext3 file system, 95, 119
 EXTPROC constant, 636, 647

 Fagin, R., 710, 715, 886
 fatal error, 16
 fattach function, 589, 592–593
 definition of, 589
 fchdir function, 125–127, 548
 definition of, 125
 fchmod function, 99–101, 112, 117, 306, 458, 548
 definition of, 99
 fchown function, 102–103, 117, 306, 548
 definition of, 102
 fclose function, 138–140, 181, 183, 340–341, 412,
 507, 661
 definition of, 139
 fcntl function, 59, 73, 76–83, 86, 105, 138, 153,
 203, 234, 306, 411–412, 442, 445, 447–450, 452,
 454–455, 482, 548, 581–583, 749–751,
 880–882
 definition of, 78
 <fcntl.h> header, 29, 60
 fcvt function, 402
 fdatsync function, 77–78, 82–83, 306, 548
 definition of, 77
 FD_CLOEXEC constant, 78–79, 234
 FD_CLR function, 476, 625, 875
 definition of, 476

- fdetach function, 590
 - definition of, 590
- FD_ISSET function, 476, 625, 875
 - definition of, 476
- fdopen function, 138–140, 506, 877
 - definition of, 138
- fd_set data type, 57, 475–476, 493, 625, 779–780, 874, 879
- FD_SET function, 476, 625, 875
 - definition of, 476
- FD_SETSIZE constant, 477, 874
- F_DUPFD constant, 77–79, 548
- FD_ZERO function, 476, 625, 875
 - definition of, 476
- feature test macro, 55–56, 81
- Fenner, B., 147, 266, 429, 545, 890
- <fcntl.h> header, 27
- feof function, 141, 146
 - definition of, 141
- error function, 10, 141, 143–144, 146, 254, 500, 505, 512, 588
 - definition of, 141
- FF0 constant, 647
- FF1 constant, 647
- FFDLY constant, 637, 644, 647, 649
- fflush function, 135, 137, 139, 160, 341, 412, 508–510, 514, 662, 680, 849, 851, 854, 861
 - definition of, 137
- F_FREESP constant, 105
- fgetc function, 140–141, 144–145, 412
 - definition of, 140
- F_GETFD constant, 78–79, 548
- F_GETFL constant, 78–81, 548
- F_GETLK constant, 78, 446–450
- F_GETOWN constant, 78–79, 548, 582
- fgetpos function, 147–148, 412
 - definition of, 148
- fgets function, 9, 11–12, 19, 140, 142–145, 156, 159, 196, 198, 412, 500, 505, 509, 512–514, 570, 577, 587, 616, 703, 718, 807, 859, 861, 877
 - definition of, 142
- fgetwc function, 412
- fgetws function, 412
- FIFOs, 89, 496, 514–518, 589
- file
 - access permissions, 92–94, 130
 - block special, 89, 128–129
 - character special, 89, 128–129, 461, 466, 659
 - descriptor passing, 543, 601–614
 - descriptor passing, socket, 606–614
 - descriptor passing, STREAMS, 604–606
 - descriptors, 8–10, 59–60
 - device special, 127–129
 - directory, 88
 - group, 166–167
 - holes, 65–66, 104–105
 - mode creation mask, 97–98, 119, 131, 215, 234, 425
 - offset, 63–65, 71–74, 76, 213–214, 454, 484, 713–714, 856
 - ownership, 95
 - pointer, 134
 - regular, 88
 - sharing, 70–73, 213
 - size, 103–105
 - times, 115–116, 493
 - truncation, 105
 - types, 88–91
- FILE structure, 121, 133–134, 141, 153–154, 156, 202, 217, 254, 402–403, 500, 504–505, 507, 509, 577, 661, 720, 872
- file system, 4, 105–108
 - devfs, 129
 - ext2, 69, 82, 95, 119
 - ext3, 95, 119
 - HSFS, 105
 - PCFS, 48, 55, 105
 - S5, 62
 - UFS, 48, 55, 62, 105, 108, 119
- filename, 4
 - truncation, 62
- FILENAME_MAX constant, 38
- fileno function, 153, 506–507, 661, 861
 - definition of, 153
- _FILE_OFFSET_BITS constant, 67
- FILEPERM constant, 762, 788
- files and directories, 4–7
- FILESIZEBITS constant, 39, 43, 48
- find program, 116, 125, 234
- finger program, 131, 163, 858
- FIOASYNC constant, 583, 880–881
- FIOSETOWN constant, 583
- FIPS, 33
- Flandrena, B., 211, 889
- <float.h> header, 27, 38
- flock function, 445
- flock structure, 446, 448–449, 454
- flockfile function, 402–403
 - definition of, 403
- FLUSHO constant, 636, 640, 647
- FMNAMESZ constant, 466
- fmtmsg function, 193
- <fmtmsg.h> header, 30
- FNDELAY constant, 442

- <fnmatch.h> header, 29
- F_OK constant, 96
- foo_alloc function, 372
- foo_find function, 376
- foo_hold function, 376
- foo_rele function, 376
- fopen function, 5, 134, 138–140, 154, 202, 254, 412, 500–501, 504, 661, 872
 - definition of, 138
- FOPEN_MAX constant, 38, 42
- foreground process group, 272–278, 280–281, 286, 294, 296–297, 343, 350, 424–425, 463, 640–642, 645, 649, 670, 706, 882
- foreground process group ID, 274, 278, 637
- fork function, 11–12, 19, 22, 73, 210–219, 223–225, 227–231, 235–236, 240, 242, 245–248, 250–253, 255–256, 259, 262, 264, 266–267, 270–271, 279, 282–284, 287, 301, 306, 309, 345–347, 354, 416–421, 425–428, 430, 451–453, 458–459, 473, 489, 495–501, 503, 506–507, 511, 519, 527, 539, 541, 544, 573–574, 587, 602, 615–617, 620–621, 629, 675, 680, 682–683, 691–692, 697, 704, 747, 865–866, 870–871, 873, 876, 878, 880, 886
 - definition of, 211
- forkl function, 211
- Fowler, G. S., 125, 887, 890
- fpathconf function, 37, 39–48, 52–54, 103, 121, 306, 499, 639
 - definition of, 41
- FPE_FLTDIV constant, 327
- FPE_FLTINV constant, 327
- FPE_FLTOVF constant, 327
- FPE_FLTRES constant, 327
- FPE_FLTSUB constant, 327
- FPE_FLTUND constant, 327
- FPE_INTDIV constant, 327
- FPE_INTOVF constant, 327
- fpos_t data type, 57, 147
- fprintf function, 149, 412, 854
 - definition of, 149
- fputc function, 135, 142, 144–145, 412
 - definition of, 142
- fputs function, 136, 140, 142–145, 154, 156, 159, 412, 505, 509, 512, 587, 661, 849, 851, 859, 862, 877
 - definition of, 143
- fputwc function, 412
- fputws function, 412
- F_RDLCK constant, 446–447, 449–450, 845, 873–874
- fread function, 140, 145–147, 250, 254, 412
 - definition of, 146
- free function, 157, 159, 189–192, 306, 372, 374–376, 378, 410, 657, 728
 - definition of, 189
- freeaddrinfo function, 555
 - definition of, 555
- FreeBSD, xxii–xxiii, 3, 21, 26–27, 29–30, 35–36, 38, 48, 55, 58, 60, 62, 65, 78–79, 84, 95, 101–103, 112, 119, 122, 128, 162, 166, 169, 171–172, 176, 191, 193–194, 204, 206–207, 211, 222, 227, 242–243, 250–251, 257, 264–265, 268, 278, 285, 290–292, 295, 298, 304–305, 308, 310, 326, 330, 333, 348, 352, 357, 360, 364, 445, 452–453, 457, 459, 474–475, 496, 521, 523, 529, 534, 538, 550–551, 566–568, 583, 588, 596, 610–611, 614, 635–638, 645–651, 676, 682–683, 692, 705–706, 710, 876, 888
- freopen function, 134, 138–140, 412
 - definition of, 138
- fscanf function, 151, 412
 - definition of, 151
- fsck program, 114
- fseek function, 139, 147–148, 412
 - definition of, 147
- fseeko function, 147–148, 412
 - definition of, 148
- F_SETFD constant, 78–79, 81, 86, 548, 855
- F_SETFL constant, 78–79, 81, 86, 482, 548, 583, 855, 882
- F_SETLK constant, 78, 446–448, 450, 454, 845, 873–874
- F_SETLKW constant, 78, 446–448, 450, 845, 873
- F_SETOWN constant, 78–79, 482, 548, 581–583, 880–881
- fsetpos function, 139, 147–148, 412
 - definition of, 148
- fstat function, 4, 87–88, 112, 306, 458, 491, 497, 542, 548, 658, 725, 796
 - definition of, 87
- fsync function, 59, 77–78, 82–83, 160, 306, 411, 489, 548, 752, 861
 - definition of, 77
- ftell function, 147–148, 412
 - definition of, 147
- ftello function, 147–148, 412
 - definition of, 148
- ftok function, 519
 - definition of, 519
- ftpd program, 431, 871
- ftruncate function, 105, 117, 306, 491, 548
 - definition of, 105
- fcntllockfile function, 402–403

- definition of, 403
- fts function, 122
- ftw function, 113–114, 120–125, 131, 402, 412, 858
- <ftw.h> header, 30
- full-duplex pipes, 496
 - named, 496
- function prototypes, 807–841
- functions, system calls versus, 21–23
- F_UNLCK constant, 446–447, 449–450, 845
- funlockfile function, 402–403
 - definition of, 403
- fwide function, 134
 - definition of, 134
- fwprintf function, 412
- fwrite function, 140, 145–147, 354, 412, 868
 - definition of, 146
- F_WRLCK constant, 446–447, 449–450, 454, 845, 873
- fwscanf function, 412

- gai_strerror function, 556, 571, 574, 576, 578
 - definition of, 556
- Gallmeister, B. O., 887
- Garfinkel, S., 165, 232, 273, 887
- gather write, 483, 607
- gawk program, 243
- gcc program, 6, 26, 56
- gcvt function, 402
- gdb program, 870
- gdbm library, 710
- generic pointer, 68, 190
- getaddrinfo function, 555–557, 559–560, 569–571, 574, 576, 578, 764, 770
 - definition of, 555
- getaddrlist function, 764, 770
- GETALL constant, 530
- getc function, 10, 140–143, 145, 153–154, 412, 661–662, 861
 - definition of, 140
- getchar function, 140, 154, 160, 412, 509, 861
 - definition of, 140
- getchar_unlocked function, 402–403, 412
 - definition of, 403
- getconf program, 67
- getc_unlocked function, 402–403, 412
 - definition of, 403
- getcwd function, 49, 125–127, 132, 190, 412, 859–860
 - definition of, 126
- getdate function, 193, 402, 412
- getegid function, 210, 306
 - definition of, 210
- getenv function, 186, 192–194, 402–405, 409–410, 421, 501, 870
 - definition of, 192
- getenv_r function, 404–405
- geteuid function, 210, 238–239, 249, 306, 612, 771
 - definition of, 210
- getgid function, 17, 210, 306
 - definition of, 210
- getgrent function, 167–168, 402, 412
 - definition of, 167
- getgrgid function, 166, 402, 412
 - definition of, 166
- getgrgid_r function, 402, 412
- getgrnam function, 166, 402, 412, 687
 - definition of, 166
- getgrnam_r function, 402, 412
- getgroups function, 168, 306
 - definition of, 168
- gethostbyaddr function, 402, 412, 553, 555
- gethostbyname function, 402, 412, 553, 555
- gethostent function, 402, 412, 553
 - definition of, 553
- gethostname function, 39, 42, 172, 412, 571–573, 578, 778
 - definition of, 172
- getlogin function, 256–257, 402, 412, 439, 871–872
 - definition of, 256
- getlogin_r function, 402, 412
- getmsg function, 411, 461–463, 469–472, 493, 548, 605, 705, 875
 - definition of, 469
- getnameinfo function, 556
 - definition of, 556
- GETNCNT constant, 530
- getnetbyaddr function, 402, 412, 554
 - definition of, 554
- getnetbyname function, 402, 412, 554
 - definition of, 554
- getnetent function, 402, 412, 554
 - definition of, 554
- get_newjobno function, 783, 788
- getopt function, 402, 624, 694, 696, 770, 773–774
 - definition of, 774
- getpass function, 263, 273, 660, 662–663
 - definition of, 661
- getpeername function, 306, 561
 - definition of, 561
- getpgid function, 269
 - definition of, 269
- getpgrp function, 269, 306

- definition of, 269
- GETPID constant, 530
- getpid function, 11, 210, 212, 217, 253, 284, 306, 341, 351, 359, 434, 612, 881
 - definition of, 210
- getpmsg function, 411, 461–463, 469–470, 548
 - definition of, 469
- getppid function, 210–211, 306, 451, 697
 - definition of, 210
- get_printaddr function, 766, 782
- get_printserver function, 766, 770
- getprotobyname function, 402, 412, 554
 - definition of, 554
- getprotobynumber function, 402, 412, 554
 - definition of, 554
- getprotoent function, 402, 412, 554
 - definition of, 554
- getpwent function, 164–165, 402, 412
 - definition of, 164
- getpwnam function, 161–165, 170, 256, 263, 306–308, 402, 412, 779, 861–862
 - definition of, 163–164
- getpwnam_r function, 402, 412
- getpwuid function, 161–165, 170, 256–257, 402, 412, 771, 861
 - definition of, 163
- getpwuid_r function, 402, 412
- getrlimit function, 52, 202, 205, 426–427, 854–855
 - definition of, 202
- getrusage function, 227, 258
- gets function, 142–143, 412, 859
 - definition of, 142
- getservbyname function, 402, 412, 555
 - definition of, 555
- getservbyport function, 402, 412, 555
 - definition of, 555
- getservent function, 402, 412, 555
 - definition of, 555
- getsid function, 271
 - definition of, 271
- getsockname function, 306, 561
 - definition of, 561
- getsockopt function, 306, 579–580
 - definition of, 579
- getspent function, 166
 - definition of, 166
- getspnam function, 166, 861
 - definition of, 166
- gettimeofday function, 173, 176, 383, 398
 - definition of, 173
- getty program, 220, 262–266, 431
- gettytab file, 263
- getuid function, 17, 210, 238–239, 249, 256–257, 306, 687
 - definition of, 210
- getutxent function, 402, 412
- getutxid function, 402, 412
- getutxline function, 402, 412
- GETVAL constant, 530
- getwc function, 412
- getwchar function, 412
- getwd function, 412
- GETZCNT constant, 530
- GID, *see* group ID
- gid_t data type, 57
- Gingell, R. A., 188, 487, 887
- Gitlin, J. E., xxviii
- glob function, 412
- global variables, 201
- <glob.h> header, 29
- gmtime function, 174–175, 402
 - definition of, 175
- gmtime_r function, 402
- GNU, 2, 265, 719
- GNU Public License, 35
- _GNU_SOURCE constant, 91
- Godsil, J. M., xxviii
- Goodheart, B., 672, 887
- goto, nonlocal, 195–202, 329–333
- Grandi, S., xxviii
- grantpt function, 682–685, 688–691, 707
 - definition of, 682, 687, 690
- grep program, 20, 159, 182, 234, 887
- group file, 166–167
- group ID, 17, 237–241
 - effective, 91–92, 94–95, 101, 103, 130, 167, 210, 214, 237, 241, 520, 543, 605
 - real, 91–92, 95, 167, 210, 214, 234–235, 237, 251, 541
 - supplementary, 18, 39, 91–92, 94, 101, 103, 167–168, 214, 234, 241
- group structure, 166, 687
- <grp.h> header, 29, 166, 170
- guardsize attribute, 389, 392

- hack, 278
- half-duplex pipes, 496
- hard link, 4, 107, 109, 112, 114
- hcreate function, 402
- hdestroy function, 402
- headers
 - optional, 30

- POSIX required, 29
 - standard, 27
 - XSI extension, 30
- heap, 187
- Hein, T. R., xxviii, 889
- Hewlett-Packard, 36, 798
- Hogue, J. E., xxviii
- holes, file, 65–66, 104–105
- home directory, 2, 7, 125, 193, 264, 267
- HOME environment variable, 192–193, 264
- Honeyman, P., xxviii
- hostent structure, 553
- hostname program, 173
- HOST_NAME_MAX constant, 39, 42, 48, 172, 570–573, 577–578, 778
- HP-UX, 36
- hsearch function, 402
- HSFS file system, 105
- htonl function, 550, 797
 - definition of, 550
- htons function, 550, 797
 - definition of, 550
- HTTP (Hypertext Transfer Protocol), 756
- Hume, A. G., 159, 887
- HUPCL constant, 635, 647
- Hypertext Transfer Protocol, *see* HTTP

- IBM (International Business Machines), 36
- ICANON constant, 636, 638, 640–642, 646–647, 651, 663, 665–667
- I_CANPUT constant, 465
- iconv_close function, 412
- <iconv.h> header, 30
- iconv_open function, 412
- ICRNL constant, 635, 640, 648, 660, 666–668
- identifiers
 - IPC, 518–520
 - process, 209–210
- IDXLLEN_MAX constant, 745
- IEC, 25
- IEEE (Institute for Electrical and Electronic Engineers), xx, 26–27, 887
- IEXTEN constant, 636, 638, 640–642, 648, 666–668
- I_FIND constant, 684–685
- IFS environment variable, 250
- IGNBRK constant, 635, 645, 648
- IGNCR constant, 635, 640, 648, 660
- IGNPAR constant, 635, 648, 650
- I_GRDOPT constant, 470
- I_GWROPT constant, 468
- I_LIST constant, 466–467
- ILL_BADSTK constant, 327
- ILL_COPROC constant, 327
- ILL_ILLADR constant, 327
- ILL_ILLOPC constant, 327
- ILL_ILLOPN constant, 327
- ILL_ILLTRP constant, 327
- ILL_PRVOPC constant, 327
- ILL_PRIVREG constant, 327
- IMAXBEL constant, 635, 648
- implementation differences, password, 169
- implementations, UNIX System, 33
- INADDR_ANY constant, 561
- in_addr_t data type, 551
- incore, 70
- INET6_ADDRSTRLEN constant, 552
- inet_addr function, 552
- INET6_ADDRSTRLEN constant, 552, 559–560
- inetd program, 266–268, 425, 430–431
- inet_ntoa function, 402, 552
- inet_ntop function, 552, 560
 - definition of, 552
- inet_pton function, 552
 - definition of, 552
- INFTIM constant, 480
- init program, 171, 173, 210, 219–220, 228, 262–266, 268, 282–283, 287, 295, 312, 350, 425, 434, 866, 871
- initgroups function, 168, 264
 - definition of, 168
- initialized data segment, 187
- init_printer function, 778, 782, 796
- init_request function, 778, 781
- initserver function, 570–572, 574, 577–578, 763, 779
 - definition of, 564, 580
- inittab file, 295
- INLCR constant, 635, 648
- i-node, 57, 71–72, 88, 101, 105, 107–108, 112, 115–117, 120–121, 128–129, 163, 287, 453, 658, 853, 858
- ino_t data type, 57, 107
- INPCK constant, 635, 648, 650, 666–668
- in_port_t data type, 551
- Institute for Electrical and Electronic Engineers, *see* IEEE
- int16_t data type, 794
- International Business Machines, *see* IBM
- International Standards Organization, *see* ISO
- Internet Printing Protocol, *see* IPP
- Internet worm, 142
- interpreter file, 242–246, 260
- interprocess communication, *see* IPC

- interrupted system calls, 303–305, 317–318, 326, 329, 339, 481
- INT_MAX constant, 38
- INT_MIN constant, 38
- INTR terminal character, 638, 641, 648, 661
- <inttypes.h> header, 27
- I/O
 - asynchronous, 473, 481–482
 - asynchronous socket, 582–583
 - efficiency, 68–70
 - library, standard, 9, 133–160
 - memory-mapped, 487–492
 - multiplexing, 472–481
 - nonblocking, 441–444
 - nonblocking socket, 563–564, 582–583
 - terminal, 631–673
 - unbuffered, 8, 59–86
- IOBUFSZ constant, 799
- ioctl function, 59, 83–84, 86, 273, 297, 303–304, 412, 442, 460–462, 464–468, 470, 482, 524, 548, 583, 587, 593, 600, 604–606, 634, 670–671, 684–685, 689–690, 692–693, 695, 705–707, 880–881
 - definition of, 83
- ioctl operations, STREAMS, 464
- _IOFBF constant, 137
- _IOLBF constant, 137, 202
- _IO_LINE_BUFFERED constant, 154
- _IONBF constant, 137
- _IO_UNBUFFERED constant, 154
- iovec structure, 40–42, 483, 566, 608–609, 611, 613, 617, 621, 737, 799
- IOV_MAX constant, 40, 42, 48, 483
- IPC (interprocess communication), 495–544, 585–629
 - identifiers, 518–520
 - key, 518–520, 524, 529, 534
 - XSI, 518–522
- IPC_CREAT constant, 519–520
- IPC_EXCL constant, 520
- IPC_NOWAIT constant, 525–526, 531–532
- ipc_perm structure, 520, 524, 529, 534, 543
- IPC_PRIVATE constant, 519, 537, 542, 544
- ipcrm program, 521
- IPC_RMID constant, 524–525, 530, 535–537
- ipcs program, 521, 544
- IPC_SET constant, 524–525, 530, 535
- IPC_STAT constant, 524–525, 530, 535
- IPP (Internet Printing Protocol), 753–756
- ipp.h header, 805
- IPPROTO_IP constant, 579
- IPPROTO_RAW constant, 558
- IPPROTO_TCP constant, 558, 579
- IPPROTO_UDP constant, 558
- I_PUSH constant, 593, 685
- I_RECVFD constant, 593, 600, 604–606
- IRIX, 36
- isastream function, 464–465, 467, 594
 - definition of, 465
- isatty function, 464–465, 639, 655, 658–659, 671, 694, 702
 - definition of, 655
- I_SENDFD constant, 604–605
- I_SETSIG constant, 482
- ISIG constant, 636, 638, 640–642, 648, 666–668
- ISO (International Standards Organization), xx, xxvii, 25–27, 887
- ISO C, 25–26, 887
- <iso646.h> header, 27
- Israel, R. K., 462, 889
- I_SRDOPT constant, 470
- is_read_lockable function, 450, 845
- ISTRIP constant, 635, 648, 650, 666–668
- is_write_lockable function, 450, 845
- I_SWROPT constant, 468
- IUCLC constant, 635, 649
- IXANY constant, 635, 649
- IXOFF constant, 635, 641–642, 649
- IXON constant, 635, 641–642, 649, 666–668
- jmp_buf data type, 198, 200, 315, 318
- job control, 274–278
 - shell, 270, 274, 280, 283, 300, 333, 350, 699
 - signals, 349–352
- job_find function, 869
- job_remove function, 869
- Jolitz, W. F., 34
- Joy, W. N., 3, 71
- jsh program, 275
- Karels, M. J., 33–34, 70, 104, 108, 211, 218, 461, 487, 888
- kdump program, 457
- kernel, 1
- Kernighan, B. W., xx, xxviii, 26, 139, 145, 151, 153, 190, 243, 846, 854, 885, 887
- key, IPC, 518–520, 524, 529, 534
- key_t data type, 518
- kill function, 18, 253, 283–284, 290, 300, 306, 310–313, 327, 338, 341–342, 351–352, 354, 414, 416, 639, 641, 662, 697–698, 867, 874

- definition of, 312
 - kill program, 290–291, 296, 300, 513
 - KILL terminal character, 638, 641, 647, 662–663
 - kill_workers function, 791–793
 - Kleiman, S. R., 71, 887
 - Knuth, D. E., 730, 888
 - Korn, D. G., 3, 125, 159, 510, 886–888, 890
 - Korn shell, 3, 52, 86, 158, 192, 204, 265, 275, 457, 510, 662, 698–699, 701, 877, 886
 - Kovach, K. R., 521, 885
 - Krieger, O., 159, 492, 888
 - kttrace program, 457
-
- 164a function, 402
 - LANG environment variable, 41, 193
 - <langinfo.h> header, 30
 - last program, 171
 - layers, shell, 274
 - LC_ALL environment variable, 193
 - LC_COLLATE environment variable, 42, 193
 - LC_CTYPE environment variable, 193
 - lchown function, 102–103, 112–113, 117
 - definition of, 102
 - LC_MESSAGES environment variable, 193
 - LC_MONETARY environment variable, 193
 - LC_NUMERIC environment variable, 193
 - L_ctermid constant, 654
 - LC_TIME environment variable, 193
 - ld program, 189
 - LDAP (Lightweight Directory Access Protocol), 169
 - LD_LIBRARY_PATH environment variable, 719
 - ldterm STREAMS module, 468, 676, 685
 - leakage, memory, 191
 - least privilege, 237, 758, 779
 - Lee, M., 188, 887
 - Lee, T. P., 886
 - Leffler, S. J., 34, 888
 - Lennert, D., 888
 - Lesk, M. E., 133
 - lgamma function, 402
 - lgammaf function, 402
 - lgammal function, 402
 - Libes, D., 679, 867, 888
 - <libgen.h> header, 30
 - libraries, shared, 188–189, 207, 719, 863, 885
 - Lightweight Directory Access Protocol, *see* LDAP
 - limit program, 52, 204
 - limits, 36–52
 - C, 38
 - POSIX, 38–40
 - resource, 202–206, 215, 234, 297, 354
 - runtime indeterminate, 48–52
 - XSI, 40–41
 - <limits.h> header, 27, 38–40, 48–49
 - Linderman, J. P., xxviii
 - line control, terminal I/O, 653–654
 - LINE_MAX constant, 39, 42, 48
 - LINES environment variable, 193
 - link count, 43, 57, 107–109, 120
 - link function, 75, 107–114, 117, 306
 - definition of, 109
 - link, hard, 4, 107, 109, 112, 114
 - symbolic, 88–89, 102–103, 107, 110, 112–114, 121, 127, 131, 170, 856–857
 - LINK_MAX constant, 39, 43, 48, 107
 - lint program, 182
 - Linux, xxi, xxiii, 2–3, 14, 21, 26–27, 29–30, 35–36, 38, 40, 48, 51, 55, 58, 60, 62, 69–72, 82–84, 91, 95, 101–103, 112, 114, 119, 122, 128, 154, 162, 166, 169, 171–172, 176, 187, 191, 193–194, 203–204, 207, 211, 222, 227, 242–243, 250–251, 255, 264–265, 268, 278, 280, 290, 292–296, 298, 304–305, 308, 310, 326, 329–330, 333, 348, 352, 357, 360, 364, 424, 437, 445, 455–457, 460–461, 464, 474–475, 484, 492, 496, 521, 523, 529, 533–535, 537–538, 540, 550–552, 566–568, 583, 585, 595, 610–612, 614, 635–638, 644–651, 653, 676, 683, 686, 689, 691–692, 705–706, 710, 719, 876
 - Linux STREAMS, 496
 - Lions, J., 888
 - LiS, 496
 - listen function, 306, 561, 563–564, 581, 597–598, 762
 - definition of, 563
 - little-endian byte order, 549
 - Litwin, W., 710, 715, 888
 - LLONG_MAX constant, 38
 - LLONG_MIN constant, 38
 - ln program, 107
 - LNEXT terminal character, 638, 641
 - locale, 42
 - localeconv function, 402
 - <locale.h> header, 27
 - LOCAL_PEERCREC constant, 612
 - localtime function, 174–176, 246, 402, 862
 - definition of, 175
 - localtime_r function, 402
 - lockf function, 411, 445
 - lockf structure, 453
 - lockfile function, 433
 - definition of, 454

- locking
 - database library, coarse-grained, 718
 - database library, fine-grained, 718
- locking function, 445
- lock_reg function, 448, 845, 873–874
 - definition of, 449
- lock_test function, 449–450, 845
 - definition of, 449
- log function, 429
- LOG_ALERT constant, 431
- LOG_AUTH constant, 431
- LOG_AUTHPRIV constant, 431
- LOG_CONS constant, 428, 430
- LOG_CRIT constant, 431
- LOG_CRON constant, 431
- LOG_DAEMON constant, 428, 431
- LOG_DEBUG constant, 431
- LOG_EMERG constant, 431
- LOG_ERR constant, 431, 433, 435–436, 438,
570–574, 577–578, 850
- LOG_FTP constant, 431
- logger program, 432
- login accounting, 170–171
- .login file, 265
- login name, 2, 17, 125, 163, 171, 193, 256–257, 266,
439, 871
 - root, 16
- login program, 163, 166, 168, 171, 233, 236, 238,
257, 263–267, 431, 660, 678, 703
- LOG_INFO constant, 431, 435, 437
- LOGIN_NAME_MAX constant, 39, 42, 48
- logins
 - network, 266–268
 - terminal, 261–266
- LOG_KERN constant, 431
- LOG_LOCAL0 constant, 431
- LOG_LOCAL1 constant, 431
- LOG_LOCAL2 constant, 431
- LOG_LOCAL3 constant, 431
- LOG_LOCAL4 constant, 431
- LOG_LOCAL5 constant, 431
- LOG_LOCAL6 constant, 431
- LOG_LOCAL7 constant, 431
- LOG_LPR constant, 431
- LOG_MAIL constant, 431
- log_msg function, 845–846
 - definition of, 850
- LOGNAME environment variable, 193, 257, 264
- LOG_NDELAY constant, 430, 871
- LOG_NEWS constant, 431
- LOG_NOTICE constant, 431
- log_open function, 624, 845
 - definition of, 849
- LOG_PERROR constant, 430
- LOG_PID constant, 430, 624
- log_quit function, 793, 845–846
 - definition of, 850
- log_ret function, 845–846
 - definition of, 849
- log_sys function, 766, 782, 845–846
 - definition of, 850
- LOG_SYSLOG constant, 431
- log_to_stderr variable, 624, 776, 849, 851
- LOG_USER constant, 431, 624
- LOG_WARNING constant, 431
- LONG_BIT constant, 40
- longjmp function, 179, 195, 197–201, 206, 305,
307, 315, 317–318, 329–331, 333, 340, 354, 867
 - definition of, 197
 - _longjmp function, 330, 333
- LONG_MAX constant, 38, 51, 58, 854–855
- LONG_MIN constant, 38
- loop function, 624, 626, 629, 696, 707
 - definition of, 626, 697
- lp program, 541, 757
- lpc program, 431
- lpd program, 431, 757
- lpsched program, 541, 757
- lrand48 function, 402
- ls program, 5–6, 8, 13, 100–101, 104, 114, 116, 121,
125, 129, 131, 161, 163, 521, 853
- lseek function, 8, 57, 59, 63–67, 72–75, 84, 86,
139, 148, 306, 412, 420, 446, 449, 454, 458, 491,
548, 629, 732, 856
 - definition of, 63
- lstat function, 87–88, 90–91, 113–114, 123, 131,
306
 - definition of, 87
- L_tmpnam constant, 156
- Lucchina, P., xxviii
- Mac OS X, xxii, 3, 16, 26–27, 29–30, 35–36, 38, 48,
54–55, 58, 60, 62, 78–79, 82, 84, 95, 101–103,
112, 119, 122, 128, 162, 166, 168–169, 171–172,
176, 191, 193–194, 204, 222, 227, 242–243,
250–251, 257, 264–265, 268, 278, 290–293,
295, 298, 304–305, 308, 310, 326, 330, 348, 352,
357, 360, 445, 457, 474–475, 484, 496–497,
521, 523, 529, 534, 538, 550, 566–568, 583, 596,
610, 635–638, 645–651, 676, 683, 692,
705–706, 710, 876
- Mach, xxii–xxiii, 35, 885
- <machine/_types.h> header, 854

- macro, feature test, 55–56, 81
- MAILPATH environment variable, 192
- main function, 7, 140, 145, 179–182, 184, 186, 197–199, 207, 218–219, 231, 260, 306–308, 332–333, 428, 587, 616, 618, 624, 694, 704, 771, 774, 777, 780, 787, 793, 796, 863, 865, 867, 880, 882
- major device number, 56–57, 127, 129, 659
- major function, 128–129
- make program, 275
- makethread function, 400
- mallinfo function, 191
- malloc function, 21–23, 50, 126, 135, 157, 159, 189–192, 195, 305–306, 308, 364, 371–372, 374, 376, 391, 398, 407, 409–410, 537, 571, 573, 578, 608–609, 612–613, 622–623, 626, 656, 868, 870
 - definition of, 189
- MALLOC_OPTIONS environment variable, 870
- malloc function, 191
- man program, 239–240
- mandatory record locking, 455
- Mandrake, xxiii
- MAP_ANON constant, 540
- MAP_ANONYMOUS constant, 540
- MAP_FAILED constant, 491, 539
- MAP_FIXED constant, 488–489
- MAP_PRIVATE constant, 488, 490, 540
- MAP_SHARED constant, 488, 490–491, 538–540
- <math.h> header, 27
- Mauro, J., 70, 105, 108, 889
- MAX_CANON constant, 39, 43, 46, 48, 633
- MAX_INPUT constant, 39, 43, 48, 632
- MAXPATHLEN constant, 49
- MB_LEN_MAX constant, 38
- mbstate_t structure, 401
- McDougall, R., 70, 105, 108, 889
- McGrath, G. J., 462, 889
- McIlroy, M. D., xxviii
- McKusick, M. K., xxviii, 33–34, 70, 104, 108, 211, 218, 461, 487, 888
- M_DATA STREAMS message type, 463–464, 468
- MDMBUF constant, 635, 645, 649
- memcpy function, 145
- memcpy function, 491–492
- memory
 - allocation, 189–192
 - layout, 186–188
 - leakage, 191
 - shared, 496, 533–540
- memory-mapped I/O, 487–492
- M_ERROR STREAMS message type, 482
- message queues, 496, 522–527
 - timing, 527
- messages, STREAMS, 462
- mgetty program, 265
- M_HANGUP STREAMS message type, 482
- MIN terminal value, 647, 663–664, 668, 673, 882
- minor device number, 56–57, 127, 129, 659
- minor function, 128–129
- mkdir function, 95, 112–114, 116–117, 119–120, 306, 860
 - definition of, 119
- mkdir program, 119
- mkfifo function, 112–113, 116–117, 306, 514–515, 878
 - definition of, 514
- mkfifo program, 515
- mknod function, 112–113, 119, 515
- mkstemp function, 155–159, 412
 - definition of, 158
- mktemp function, 158
- mktime function, 174–176
 - definition of, 175
- mlock function, 203
- mmap function, 159, 203, 391, 441, 487–489, 491–493, 538–540, 543, 548, 887
 - definition of, 487
- modem, xx, xxiii, 261, 263, 272, 294, 303, 441, 481, 631, 634–635, 645, 647, 649, 652
- mode_t data type, 57
- <monetary.h> header, 30
- Moran, J. P., 487, 887
- more program, 505, 714
- MORECTL constant, 470
- MOREDATA constant, 470
- Morris, R., 165, 889
- mount function, 592
- mount program, 95, 119, 129, 455
- mounted STREAMS-based pipes, 495, 514, 518
- M_PCPROTO STREAMS message type, 463–464
- mprotect function, 489
 - definition of, 489
- M_PROTO STREAMS message type, 463–464
- mq_receive function, 411
- mq_send function, 411
- mq_timedreceive function, 411
- mq_timedsend function, 411
- <mqueue.h> header, 30
- mrnd48 function, 402
- MS_ASYNC constant, 490
- MSG_ANY constant, 469
- MSG_BAND constant, 464, 469
- msgctl function, 520–521, 524

- definition of, 524
 - MSG_TRUNC constant, 568
 - MSG_DONTROUTE constant, 566
 - MSG_DONTWAIT constant, 566, 568
 - MSG_EOR constant, 566, 568
 - msgget function, 518–519, 521–524
 - definition of, 524
 - msghdr structure, 566, 568, 606, 608–609, 611, 613
 - MSG_HIPRI constant, 464, 469
 - MSG_NOERROR constant, 526
 - MSG_OOB constant, 566–568, 581–582
 - MSG_PEEK constant, 567
 - msgrcv function, 411, 520–521, 523, 526, 541
 - definition of, 526
 - msgsnd function, 411, 520–523, 525–527
 - definition of, 525
 - MSG_TRUNC constant, 567–568
 - MSGVERB environment variable, 193
 - MSG_WAITALL constant, 567
 - M_SIG STREAMS message type, 463
 - MS_INVALIDATE constant, 490
 - msqid_ds structure, 523–524, 526
 - MS_SYNC constant, 490–491
 - msync function, 411, 489–491
 - definition of, 490
 - Mui, L., 672, 890
 - multiplexing, I/O, 472–481
 - munmap function, 490
 - definition of, 490
 - mv program, 107
 - myftw function, 123, 131
-
- named full-duplex pipes, 496
 - NAME_MAX constant, 39, 43, 48, 54, 62, 121
 - nanosleep function, 348, 398, 400, 411, 421
 - Nataros, S., xxviii
 - nawk program, 243
 - NCCS constant, 634
 - ndbm library, 709–710
 - <ndbm.h> header, 30
 - Nemeth, E., xxviii, 889
 - <netdb.h> header, 29, 170
 - netent structure, 554
 - <net/if.h> header, 29
 - <netinet/in.h> header, 29, 550–551
 - <netinet/tcp.h> header, 29
 - netinfo, 169
 - Network File System, Sun Microsystems, *see* NFS
 - Network Information Service, *see* NIS
 - network logins, 266–268
 - network printer communication, 753–805
 - Neville-Neil, G. V., 70, 104, 108, 888
 - newgrp program, 167
 - nfds_t data type, 479
 - _NFILE constant, 50
 - NFS (Network File System, Sun Microsystems), 72, 752
 - nftw function, 121–122, 402, 412
 - NGROUPS_MAX constant, 39, 42, 48, 167–168
 - Nievergelt, J., 710, 715, 886
 - NIS (Network Information Service), 169
 - NIS+, 169
 - NL terminal character, 638, 640–641, 647, 660, 663
 - NL0 constant, 649
 - NL1 constant, 649
 - NL_ARGMAX constant, 41
 - NLDLY constant, 637, 644, 649
 - nlink_t data type, 57, 107
 - nl_langinfo function, 402
 - NL_LANGMAX constant, 41
 - NL_MSGMAX constant, 41
 - NL_NMAX constant, 41
 - NL_SETMAX constant, 41
 - NLSPATH environment variable, 193
 - NL_TEXTMAX constant, 41
 - <nl_types.h> header, 30
 - nobody login name, 162–163
 - NOFILE constant, 50
 - NOFLSH constant, 636, 649
 - NOKERNINFO constant, 636, 642, 649
 - nologin program, 163
 - nonblocking
 - I/O, 441–444
 - socket I/O, 563–564, 582–583
 - noncanonical mode, terminal I/O, 663–670
 - nonfatal error, 16
 - nonlocal goto, 195–202, 329–333
 - ntohl function, 550, 804
 - definition of, 550
 - ntohs function, 550, 560, 804
 - definition of, 550
 - NULL constant, 786
 - null signal, 290, 312
 - NZERO constant, 41
-
- O_ACCMODE constant, 79–80
 - O_APPEND constant, 60, 63, 68, 72, 74, 79–80; 1457
 - O_ASYNC constant, 79, 482, 583
 - O_CREAT constant, 61, 63, 75, 85, 112, 117, 433, 456–458, 491, 520, 715, 724, 872
 - OCRNL constant, 637, 649

- od program, 66
- O_DSYNC constant, 61–62, 79
- O_EXCL constant, 61, 75, 112, 520
- OFDEL constant, 637, 644, 649
- off_t data type, 57, 64–67, 147–148, 738
- OFILL constant, 637, 644, 649
- O_FSYNC constant, 62, 79–80
- Olander, D. J., 462, 889
- OLCUC constant, 637, 649
- Olson, M., 889
- O_NDELAY constant, 36, 61, 442
- ONLCR constant, 637, 650, 696, 702
- ONLRET constant, 637, 650
- ONOCR constant, 637, 650
- O_NOCTTY constant, 61, 273, 426, 681–682, 685
- ONOEOT constant, 637, 650
- O_NONBLOCK constant, 36, 61, 79–80, 442–443, 456, 458, 515, 565, 876, 878–879
- open function, 8, 14, 59–63, 74–75, 79, 85–86, 94–97, 105, 110, 112–115, 117–118, 127, 138–139, 260, 263, 273, 306, 411, 428–429, 433, 442, 452–453, 455–458, 461, 465, 467, 487, 491, 514–515, 518, 520–522, 539–541, 544, 547, 594, 615, 618–619, 628–629, 645, 681, 684–686, 688–689, 691, 711, 723–724, 786, 855, 857, 872, 878–879
 - definition of, 60
- Open Software Foundation, *see* OSF
- opend.h header, 617, 622, 881
- opendir function, 5, 7, 113, 120–125, 234, 260, 412, 657, 858
 - definition of, 120
- openlog function, 412, 428, 430, 432, 439, 849, 871
 - definition of, 430
- OPEN_MAX constant, 39, 42, 48, 50–52, 58, 60, 854
- open_max function, 426, 506, 626–627, 844
 - definition of, 51, 855
- OpenServer, 309, 445
- OPOST constant, 637, 650, 666–668, 670
- optarg variable, 774
- opterr variable, 774
- optind variable, 770
- option codes, 31
- options, 52–55
 - socket, 579–581
- optopt variable, 774
- ordering, byte, 549–550
- O_RDONLY constant, 60, 79–80, 94, 96, 465, 467, 491, 616, 878
- O_RDWR constant, 60, 79–80, 94, 118, 428, 433, 458, 491, 539, 594, 681, 684, 688, 690–691, 715, 872
- O'Reilly, T., 672, 890
- orientation, stream, 134
- orphaned process group, 282–285, 428, 699
- O_RSYNC constant, 61–62, 79
- OSF (Open Software Foundation), 32
- O_SYNC constant, 61–62, 79–80, 82–83
- O_TRUNC constant, 61, 63, 94, 105, 117–118, 139, 456, 458, 491, 715
- out-of-band data, 581–582
- ownership
 - directory, 95
 - file, 95
- O_WRONLY constant, 60, 79–80, 94, 879
- OXTABS constant, 637, 650
- packet mode, pseudo terminal, 705
- page cache, 77
- page size, 535
- pagedaemon process, 210
- PAGER environment variable, 501, 504–505
- PAGESIZE constant, 39, 42, 48, 392
- PAGE_SIZE constant, 40, 42, 48
- P_ALL constant, 226
- PARENB constant, 635, 648, 650, 666–668
- parent
 - directory, 4, 101, 116, 119
 - process ID, 210, 215, 219, 225, 228, 234, 263–264, 283, 424
- PAREXT constant, 635, 650
- parity, terminal I/O, 648
- PARMRK constant, 635, 645, 648, 650
- PARODD constant, 635, 645, 648, 650, 673
- Partridge, C., xxviii
- passing, file descriptor, 543, 601–614
 - socket file descriptor, 606–614
 - STREAMS file descriptor, 604–606
- passwd program, 92, 166, 679
- passwd structure, 161, 164, 307, 861–862
- password
 - file, 161–165
 - implementation differences, 169
 - shadow, 165–166, 178, 861
- PATH environment variable, 93, 193, 232–233, 235, 242, 244, 247, 264–265
- path_alloc function, 123, 127, 844, 860
 - definition of, 49
- pathconf function, 37, 39–50, 52–55, 103, 113, 306, 499
 - definition of, 41
- PATH_MAX constant, 39, 43, 48–49, 62, 132, 859
- pathname, 5
 - absolute, 5, 7, 43, 49, 126, 131, 242, 859

- relative, 5, 7, 43, 49, 125
- truncation, 62
- pause function, 299–300, 303, 306, 309, 313–318, 331, 333, 340, 349, 411, 419, 671, 867, 873
 - definition of, 313
- `_PC_ASYNC_IO` constant, 54
- `_PC_CHOWN_RESTRICTED` constant, 54
- `_PC_FILESIZEBITS` constant, 43
- PCFS file system, 48, 55, 105
- `pckt` STREAMS module, 676, 705
- `_PC_LINK_MAX` constant, 43
- `pclose` function, 249, 412, 503–510, 571, 578, 877–878
 - definition of, 503, 507
- `_PC_MAX_CANON` constant, 43, 46
- `_PC_MAX_INPUT` constant, 43
- `_PC_NAME_MAX` constant, 43
- `_PC_NO_TRUNC` constant, 54–55
- `_PC_PATH_MAX` constant, 43, 50
- `_PC_PIPE_BUF` constant, 43
- `_PC_PRIO_IO` constant, 54
- `_PC_SYMLINK_MAX` constant, 43
- `_PC_SYNC_IO` constant, 54
- `_PC_VDISABLE` constant, 54, 639
- PENDIN constant, 636, 650
- Pentium, xxiii
- permissions, file access, 92–94, 130
- `perorr` function, 15–16, 24, 309, 352, 412, 556, 853
 - definition of, 15
- `pgrp` structure, 286–287
- PID, *see* process ID
- `pid_t` data type, 57, 269, 356
- Pike, R., 211, 887, 889
- `pipe` function, 116–117, 138, 306, 497, 499–500, 502, 506–507, 511, 513, 527, 588–589, 593, 595, 876
 - definition of, 497
- `PIPE_BUF` constant, 39, 43, 48, 493, 499, 515–516, 590, 876
- pipes, 496–503
 - full-duplex, 496
 - half-duplex, 496
 - mounted STREAMS-based, 495, 514, 518
 - named full-duplex, 496
 - STREAMS-based, 585–594
- Pippenger, N., 710, 715, 886
- Plan 9 operating system, 211, 889
- Plauger, P. J., 26, 153, 299, 889
- pointer, generic, 68, 190
- `poll` function, 295, 305–306, 318, 411, 441, 460–461, 474, 479–481, 493, 521, 542, 544, 548, 563–564, 582, 621, 624, 626–627, 678, 696, 707, 875, 878, 881
 - definition of, 479
- `POLL_ERR` constant, 327
- `POLLERR` constant, 480
- `pollfd` structure, 479, 626–627, 875
- `<poll.h>` header, 30, 479
- `POLL_HUP` constant, 327
- `POLLHUP` constant, 480–481, 627, 878
- `POLL_IN` constant, 327–328
- `POLLIN` constant, 480, 626–627, 878
- polling, 228, 444, 473
- `POLL_MSG` constant, 327–328
- `POLLNVAL` constant, 480
- `POLL_OUT` constant, 327–328
- `POLLOUT` constant, 480
- `POLL_PRI` constant, 327
- `POLLPRI` constant, 480
- `POLLRDBAND` constant, 480
- `POLLRDNORM` constant, 480
- `POLLWRBAND` constant, 480
- `POLLWRNORM` constant, 480
- `popen` function, 23, 224, 231, 249, 412, 503–510, 543–544, 570, 574, 577, 579, 877–878
 - definition of, 503, 505
- Portable Operating System Environment for Computer Environments, IEEE, *see* POSIX
- `portmap` program, 425
- POSIX (Portable Operating System Environment for Computer Environments, IEEE), xix, xxvii, 26–29, 34, 246
- POSIX.1, xxii, xxvii, 27, 84, 243, 342, 507–508, 515, 545, 710, 887
- POSIX.2, 243
 - `_POSIX2_LINE_MAX` constant, 41
 - `_POSIX_ARG_MAX` constant, 39
 - `_POSIX_ASYNC_IO` constant, 54
 - `_POSIX_CHILD_MAX` constant, 39
 - `_POSIX_CHOWN_RESTRICTED` constant, 54–55, 103
 - `_POSIX_C_SOURCE` constant, 55, 81
- `posix_fadvise` function, 412
- `posix_fallocate` function, 412
- `_POSIX_HOST_NAME_MAX` constant, 39
- `_POSIX_JOB_CONTROL` constant, 53, 55
- `_POSIX_LINK_MAX` constant, 39
- `_POSIX_LOGIN_NAME_MAX` constant, 39
- `posix_madvise` function, 412
- `_POSIX_MAX_CANON` constant, 39
- `_POSIX_MAX_INPUT` constant, 39
- `_POSIX_NAME_MAX` constant, 39
- `_POSIX_NGROUPS_MAX` constant, 39
- `_POSIX_NO_TRUNC` constant, 54–55, 62
- `_POSIX_OPEN_MAX` constant, 39–40

- posix_openpt function, 681–683, 686, 688–690
 - definition of, 681, 686, 689
- _POSIX_PATH_MAX constant, 39–40, 656–657
- _POSIX_PIPE_BUF constant, 39
- _POSIX_PRIO_IO constant, 54
- _POSIX_READER_WRITER_LOCKS constant, 53
- _POSIX_RE_DUP_MAX constant, 39
- _POSIX_SAVED_IDS constant, 53, 55, 92, 238, 312
- _POSIX_SHELL constant, 53
- _POSIX_SOURCE constant, 55
- posix_spawn function, 412
- posix_spawnp function, 412
- _POSIX_SSIZE_MAX constant, 39
- _POSIX_STREAM_MAX constant, 39
- _POSIX_SYMLINK_MAX constant, 39
- _POSIX_SYMLINK_MAX constant, 39
- _POSIX_SYNC_IO constant, 54
- _POSIX_THREAD_ATTR_STACKADDR constant, 391
- _POSIX_THREAD_ATTR_STACKSIZE constant, 391
- _POSIX_THREAD_PROCESS_SHARED constant, 394
- _POSIX_THREADS constant, 54–55, 356
- _POSIX_THREAD_SAFE_FUNCTIONS constant, 401
- posix_trace_clear function, 412
- posix_trace_close function, 412
- posix_trace_create function, 412
- posix_trace_create_withlog function, 412
- posix_trace_event function, 306
- posix_trace_eventtypelist_getnext_id function, 412
- posix_trace_eventtypelist_rewind function, 412
- posix_trace_flush function, 412
- posix_trace_get_attr function, 412
- posix_trace_get_filter function, 412
- posix_trace_getnext_event function, 412
- posix_trace_get_status function, 412
- posix_trace_open function, 412
- posix_trace_rewind function, 412
- posix_trace_set_filter function, 412
- posix_trace_shutdown function, 412
- posix_trace_timedgetnext_event function, 412
- _POSIX_TTY_NAME_MAX constant, 39
- posix_typed_mem_open function, 412
- _POSIX_TZNAME_MAX constant, 39
- _POSIX_V6_ILP32_OFF32 constant, 67
- _POSIX_V6_ILP32_OFFBIG constant, 67
- _POSIX_V6_LP64_OFF64 constant, 67
- _POSIX_V6_LP64_OFFBIG constant, 67
- _POSIX_VDISABLE constant, 54–55, 638–639
- _POSIX_VERSION constant, 53, 55, 172
- PowerPC, xxiii
- P_PGID constant, 226
- PPID, *see* parent process ID
- P_PID constant, 226
- pr program, 719
- pread function, 74–75, 411, 420
 - definition of, 75
- Presotto, D. L., xxviii, 211, 585, 592, 889
- pr_exit function, 221–223, 248–249, 258, 346, 844
 - definition of, 222
- primitive system data types, 56
- print program, 757, 763, 783, 787–788, 798, 805
- prind program, 757, 805
- printer communication, network, 753–805
- printer spooling, 757–758
 - source code, 758–805
- printer_status function, 799, 801, 805
- printer_thread function, 795
- printf function, 10, 21, 41, 140, 149, 151–152, 160, 175–176, 201, 204, 207, 213, 217, 260, 283, 306, 323, 412, 514, 863
 - definition of, 149
- print.h header, 778, 783, 788
- printreq structure, 763, 771, 783, 786, 790
- printresp structure, 763
- privilege, least, 237, 758, 779
- pr_mask function, 331, 334–335, 844
 - definition of, 321
- proc structure, 286–287
- process, 10
 - accounting, 250–256
 - control, 11, 209–260
 - ID, 10, 210, 234
 - ID, parent, 210, 215, 219, 225, 228, 234, 263–264, 283, 424
 - identifiers, 209–210
 - relationships, 261–287
 - system, 210, 312
 - termination, 180–184
 - time, 20, 24, 57, 257–259
- process group, 269–270
 - background, 272, 275, 277, 279, 281–282, 284–285, 296–297, 344, 349, 882
 - foreground, 272–278, 280–281, 286, 294, 296–297, 343, 350, 424–425, 463, 640–642, 645, 649, 670, 706, 882
 - ID, 215, 234
 - ID, foreground, 274, 278, 637

- ID, session, 279
- ID, terminal, 278, 423–424
- leader, 269–271, 280, 287, 425, 692
- lifetime, 269
- orphaned, 282–285, 428, 699
- processes, cooperating, 455, 717, 883
- process-shared attribute, 394
- .profile file, 265
- program, 10
- PROT_EXEC constant, 487
- PROT_NONE constant, 487
- protoent structure, 554
- prototypes, function, 807–841
- PROT_READ constant, 487, 491, 539
- PROT_WRITE constant, 487, 491, 539
- PR_TEXT constant, 771, 788, 798
- ps program, 219, 260, 278, 280–282, 423–424, 428, 701, 866
- pselect function, 306, 474, 478–479
 - definition of, 478
- pseudo terminal, 675–707
 - packet mode, 705
 - remote mode, 706
 - signal generation, 706
 - window size, 706
- psignal function, 352
 - definition of, 352
- psem STREAMS module, 468, 676, 685
- pthread structure, 357
- pthread_atfork function, 417–419
 - definition of, 417
- pthread_attr_destroy function, 389–390
 - definition of, 389
- pthread_attr_getdetachstate function, 390
 - definition of, 390
- pthread_attr_getguardsize function, 392
 - definition of, 392
- pthread_attr_getstack function, 391–392
 - definition of, 391
- pthread_attr_getstackaddr function, 391
- pthread_attr_getstacksize function, 392
 - definition of, 392
- pthread_attr_init function, 388–390
 - definition of, 389
- pthread_attr_setdetachstate function, 389–390
 - definition of, 390
- pthread_attr_setguardsize function, 392
 - definition of, 392
- pthread_attr_setstack function, 391–392
 - definition of, 391
- pthread_attr_setstackaddr function, 391
- pthread_attr_setstacksize function, 392
 - definition of, 392
- pthread_attr_t data type, 388–390, 392, 410
- pthread_cancel function, 365, 410–411, 791
 - definition of, 365
- PTHREAD_CANCEL_ASYNCHRONOUS constant, 412
- PTHREAD_CANCEL_DEFERRED constant, 412
- PTHREAD_CANCEL_DISABLE constant, 410–411
- PTHREAD_CANCELLED constant, 361, 365
- PTHREAD_CANCEL_ENABLE constant, 410–411
- pthread_cleanup_pop function, 365–366, 790, 792
 - definition of, 365
- pthread_cleanup_push function, 365–366
 - definition of, 365
- pthread_condattr_destroy function, 401
 - definition of, 401
- pthread_condattr_getpshared function, 401
 - definition of, 401
- pthread_condattr_init function, 401
 - definition of, 401
- pthread_condattr_setpshared function, 401
 - definition of, 401
- pthread_cond_broadcast function, 384, 386, 869–870
 - definition of, 384
- pthread_cond_destroy function, 383, 421
 - definition of, 383
- pthread_cond_init function, 382–383, 421
 - definition of, 383
- PTHREAD_COND_INITIALIZER constant, 382, 385, 414
- pthread_cond_signal function, 384–385, 415–416
 - definition of, 384
- pthread_cond_t data type, 382, 385, 414
- pthread_cond_timedwait function, 383–384, 395, 411
 - definition of, 383
- pthread_cond_wait function, 383–385, 395, 411, 415, 795, 869–870
 - definition of, 383
- pthread_create function, 357–360, 362–364, 366, 368, 388–390, 415, 419, 436, 780, 869
 - definition of, 357
- PTHREAD_CREATE_DETACHED constant, 389–390
- PTHREAD_CREATE_JOINABLE constant, 389–390
- PTHREAD_DESTRUCTOR_ITERATIONS constant, 388, 407
- pthread_detach function, 367–368, 389
 - definition of, 368
- pthread_equal function, 357, 382

- definition of, 357
- pthread_exit function, 180, 218, 361-363, 365-366, 406, 787-789, 792
 - definition of, 361
- pthread_getconcurrency function, 393
 - definition of, 393
- pthread_getspecific function, 408-410
 - definition of, 408
- <pthread.h> header, 30
- pthread_join function, 361-363, 366-367, 411, 869
 - definition of, 361
- pthread_key_create function, 406-407, 409
 - definition of, 406
- pthread_key_delete function, 407
 - definition of, 407
- PTHREAD_KEYS_MAX constant, 388, 407
- pthread_key_t data type, 409
- pthread_kill function, 414
 - definition of, 414
- pthread_mutexattr_destroy function, 393, 404
 - definition of, 393
- pthread_mutexattr_getpshared function, 394
 - definition of, 394
- pthread_mutexattr_gettype function, 395
 - definition of, 395
- pthread_mutexattr_init function, 393-394, 399, 404
 - definition of, 393
- pthread_mutexattr_setpshared function, 394
 - definition of, 394
- pthread_mutexattr_settype function, 395, 399, 404
 - definition of, 395
- pthread_mutexattr_t data type, 393-394, 399, 404
- PTHREAD_MUTEX_DEFAULT constant, 395
- pthread_mutex_destroy function, 371-372, 375, 378, 382
 - definition of, 371
- PTHREAD_MUTEX_ERRORCHECK constant, 394-395
- pthread_mutex_init function, 371-372, 374, 376, 399, 404
 - definition of, 371
- PTHREAD_MUTEX_INITIALIZER constant, 371, 374, 376, 385, 409, 414, 418
- pthread_mutex_lock function, 371-372, 374-375, 377, 385-386, 399, 405, 409, 415, 418-419
 - definition of, 371
- PTHREAD_MUTEX_NORMAL constant, 394-395
- PTHREAD_MUTEX_RECURSIVE constant, 394-395, 399, 404
- pthread_mutex_t data type, 371-372, 374, 376, 385, 399, 404, 409, 414, 418
- pthread_mutex_trylock function, 371, 373
 - definition of, 371
- pthread_mutex_unlock function, 371-372, 374-375, 377-378, 385-386, 399, 405, 409-410, 415, 419
 - definition of, 371
- pthread_once function, 404-405, 408, 410
 - definition of, 408
- PTHREAD_ONCE_INIT constant, 404, 408-409
- pthread_once_init function, 409
- pthread_once_t data type, 404, 409
- PTHREAD_PROCESS_PRIVATE constant, 394
- PTHREAD_PROCESS_SHARED constant, 394
- pthread_rwlockattr_destroy function, 400
 - definition of, 400
- pthread_rwlockattr_getpshared function, 400
 - definition of, 400
- pthread_rwlockattr_init function, 400
 - definition of, 400
- pthread_rwlockattr_setpshared function, 400
 - definition of, 400
- pthread_rwlockattr_t data type, 400
- pthread_rwlock_destroy function, 379
 - definition of, 379
- pthread_rwlock_init function, 379-380
 - definition of, 379
- pthread_rwlock_rdlock function, 379, 382, 412
 - definition of, 379
- pthread_rwlock_t data type, 380
- pthread_rwlock_timedrdlock function, 412
- pthread_rwlock_timedwrlock function, 412
- pthread_rwlock_tryrdlock function, 379
 - definition of, 379
- pthread_rwlock_trywrlock function, 379
 - definition of, 379
- pthread_rwlock_unlock function, 379, 381-382
 - definition of, 379
- pthread_rwlock_wrlock function, 379, 381, 412
 - definition of, 379
- pthreads, 27, 211
- pthread_self function, 357, 359, 363

- definition of, 357
- `pthread_setcancelstate` function, 410
 - definition of, 410
- `pthread_setcanceltype` function, 411–412
 - definition of, 411
- `pthread_setconcurrency` function, 393
 - definition of, 393
- `pthread_setspecific` function, 408–409
 - definition of, 408
- `pthread_sigmask` function, 413, 436
 - definition of, 413
- `PTHREAD_STACK_MIN` constant, 388
- `pthread_t` data type, 356–357, 359, 362–363, 366, 380, 390, 415, 419, 436, 869
- `pthread_testcancel` function, 411
 - definition of, 411
- `PTHREAD_THREADS_MAX` constant, 388
- `P_tmpdir` constant, 157–158
- `ptrdiff_t` data type, 57
- `pts` STREAMS module, 468
- `ptsname` function, 402, 682–685, 687–688, 690
 - definition of, 682, 687, 689
- `pty` program, 285, 675, 679–680, 692, 694–707, 773, 882–883
- `pty_fork` function, 680, 683, 691–696, 704, 706–707
 - definition of, 691–692
- `ptym_open` function, 683, 686, 691–692, 845
 - definition of, 683, 688, 690
- `ptys_fork` function, 845
- `ptys_open` function, 683, 685–686, 689, 691–693, 845
 - definition of, 683–684, 688, 691
- `putc` function, 10, 142–143, 145, 229–230, 412, 661
 - definition of, 142
- `putchar` function, 142, 160, 412, 509
 - definition of, 142
- `putchar_unlocked` function, 402–403, 412
 - definition of, 403
- `putc_unlocked` function, 402–403, 412
 - definition of, 403
- `putenv` function, 186, 194, 232, 402, 405, 421
 - definition of, 194
- `putenv_r` function, 421
- `putmsg` function, 411, 461–463, 469, 548
 - definition of, 463
- `putpmsg` function, 411, 461–463, 548
 - definition of, 463
- `puts` function, 142–143, 412, 859
 - definition of, 143
- `pututxline` function, 402, 412
- `putwc` function, 412
 - definition of, 412
- `putwchar` function, 412
- `PWD` environment variable, 193
- `<pwd.h>` header, 29, 161, 170
- `pwrite` function, 74–75, 411, 420
 - definition of, 75

- Quarterman, J. S., 33–34, 70, 104, 108, 211, 218, 461, 487, 888
- QUIT terminal character, 638, 641, 648, 662

- race conditions, 227–231, 314, 749, 865, 867
- Rago, J. E., xxiii
- Rago, S. A., xxviii, 83, 147, 266, 460, 462, 889
- `raise` function, 306, 311–313, 340
 - definition of, 312
- `rand` function, 402
- `rand_r` function, 402
- raw terminal mode, 632, 664, 668, 673, 696, 699
- Raymond, E. S., 889
- `read` function, 8–10, 20, 57, 59, 61, 67–69, 75, 84–86, 104, 115, 117, 120, 135, 144–145, 159, 284, 304, 306, 316–318, 340, 351, 411, 420, 429, 442–443, 455–456, 458–459, 461–463, 469–473, 475, 478, 481, 485–487, 492, 498–499, 502–503, 511–513, 515, 518, 543, 546, 548, 565–567, 587, 616, 618, 625–627, 629, 632, 662–664, 668–669, 697, 702–703, 705, 714, 718, 768, 799–800, 855–856, 877–878, 882
 - definition of, 67
- `read` mode, STREAMS, 470
- `read`, scatter, 483, 607
- `readdir` function, 5, 7, 120–125, 402, 412, 657, 786
 - definition of, 120
- `readdir_r` function, 402, 412
- reading directories, 120–125
- `readlink` function, 113, 115, 306
 - definition of, 115
- `read_lock` function, 449, 453, 458, 845
- `readmore` function, 800, 802
- `readn` function, 485–486, 702, 768, 844
 - definition of, 485–486
- `readv` function, 40–42, 304, 411, 441, 483–485, 493, 548, 568, 607, 718, 732
 - definition of, 483
- `readw_lock` function, 449, 725, 845
- real
 - group ID, 91–92, 95, 167, 210, 214, 234–235, 237, 251, 541

- user ID, 39, 42, 91–92, 95, 203, 210, 214–215, 234–235, 237–241, 251, 257, 262, 264, 312, 354, 541, 685, 867
- realloc function, 49, 159, 189–190, 195, 622–623, 727, 802, 859–860
 - definition of, 189
- record locking, 444–459
 - advisory, 455
 - deadlock, 450
 - mandatory, 455
 - timing, semaphore locking versus, 533
- recv function, 306, 411, 548, 566–570, 582
 - definition of, 567
- recv_fd function, 603–605, 612, 617, 621, 844
 - definition of, 603, 605, 609
- recvfrom function, 306, 411, 567–568, 575–577, 579
 - definition of, 567
- recvmsg function, 306, 411, 568, 606, 609–610, 613
 - definition of, 568
- recv_ufd function, 612
 - definition of, 613
- redirmod STREAMS module, 468
- RE_DUP_MAX constant, 39, 42, 48
- reentrant functions, 305–308
- regcomp function, 39, 42
- regexec function, 39, 42
- <regex.h> header, 29
- register variables, 199
- regular file, 88
- relative pathname, 5, 7, 43, 49, 125
- reliable signals, 310–311
- remote mode, pseudo terminal, 706
- remove function, 108–113, 117, 412
 - definition of, 111
- remove_job function, 785, 795
- rename function, 108–113, 117, 306, 412
 - definition of, 111
- replace_job function, 784
- REPRINT terminal character, 638, 641, 647, 650, 663
- request function, 618, 625–628
 - definition of, 618, 628
- reset program, 673, 882
- resource limits, 202–206, 215, 234, 297, 354
- restarted system calls, 304–305, 317–318, 326, 329, 481, 660
- restrict keyword, 26, 87, 115, 136, 138, 142–143, 146, 148–149, 151, 153, 173, 176, 320, 324, 357, 371, 379, 383, 390–392, 394–395, 400–401, 413, 469, 475, 478, 552, 555–556, 561, 563, 567, 579
- rewind function, 139, 147–148, 156, 412
 - definition of, 147
- rewinddir function, 120–125, 412
 - definition of, 120
- rfork function, 211
- Ritchie, D. M., xx, 26, 133, 139, 145, 151, 153, 190, 460, 585, 592, 846, 854, 887, 889
- RLIM_INFINITY constant, 203, 427
- rlimit structure, 202, 205, 426, 855
- RLIMIT_AS constant, 203–205
- RLIMIT_CORE constant, 203–205, 293
- RLIMIT_CPU constant, 203–205
- RLIMIT_DATA constant, 203–205
- RLIMIT_FSIZE constant, 203–205, 354
- RLIMIT_INFINITY constant, 205, 855
- RLIMIT_LOCKS constant, 203–205
- RLIMIT_MEMLOCK constant, 203–205
- RLIMIT_NOFILE constant, 203–205, 427, 855
- RLIMIT_NPROC constant, 203–205
- RLIMIT_RSS constant, 203–205
- RLIMIT_SBSIZE constant, 203–205
- RLIMIT_STACK constant, 203–205
- RLIMIT_VMEM constant, 203–205
- rlim_t data type, 57, 204
- rlogin program, 677, 705–706
- rlogind program, 677–678, 699, 705, 882
- rm program, 521, 774
- rmdir function, 111–112, 116–117, 119–120, 306
 - definition of, 120
- RMSGD constant, 470
- RMSGN constant, 470
- RNORM constant, 470
- R_OK constant, 96
- root
 - directory, 4, 7, 24, 129, 131, 215, 234, 260, 858
 - login name, 16
- routed program, 431
- RPROTDAT constant, 470
- RPROTDIS constant, 470
- RPROTNORM constant, 470
- RS-232, 634, 645–646
- RS_HIPRI constant, 464, 469
- Rudoff, A. M., 147, 266, 429, 545, 890
- runacct program, 250
- S5 file system, 62
- sa program, 250
- sac program, 266
- Saksen, J., xxviii
- SAF (Service Access Facility), 266
- SA_INTERRUPT constant, 326, 328–329

- Salus, P. H., xxviii, 889
- SA_NOCLDSTOP constant, 326
- SA_NOCLDWAIT constant, 308, 326
- SA_NODEFER constant, 326, 328
- Santa Cruz Operation, *see* SCO
- SA_ONSTACK constant, 326
- SA_RESETHAND constant, 326, 328
- SA_RESTART constant, 304, 326, 328–329, 481
- SA_SIGINFO constant, 311, 325–326, 328
- saved
 - set-group-ID, 53, 91–92
 - set-user-ID, 53, 91–92, 238–241, 260, 264, 312, 866
- S_BANDURG constant, 482
- sbrk function, 21–23, 190, 203
- scan_configfile function, 765–766
- scanf function, 41, 140, 151–153, 412
 - definition of, 151
- _SC_ARG_MAX constant, 42, 46
- _SC_ATEXIT_MAX constant, 42
- scatter read, 483, 607
- _SC_CHILD_MAX constant, 42, 203
- _SC_CLK_TCK constant, 42, 257–258
- _SC_COLL_WEIGHTS_MAX constant, 42
- SCHAR_MAX constant, 38
- SCHAR_MIN constant, 38
- <sched.h> header, 30
- _SC_HOST_NAME_MAX constant, 42, 571–573, 578, 778
- Schwartz, A., 165, 232, 273, 887
- _SC_IOV_MAX constant, 42
- _SC_JOB_CONTROL constant, 53–54
- _SC_LINE_MAX constant, 42
- _SC_LOGIN_NAME_MAX constant, 42
- SCM_CREDENTIALS constant, 611–614
- SCM_CREDS constant, 611, 613–614
- SCM_CREDTYPE constant, 612, 614
- SCM_RIGHTS constant, 607–608, 612, 614
- _SC_NGROUPS_MAX constant, 42
- SCO (Santa Cruz Operation), 36
- _SC_OPEN_MAX constant, 42, 51, 203, 855
- _SC_PAGESIZE constant, 42, 489
- _SC_PAGE_SIZE constant, 42, 489
- _SC_READER_WRITER_LOCKS constant, 53
- _SC_RE_DUP_MAX constant, 42
- script program, 675, 678–679, 699, 701, 706–707
- _SC_SAVED_IDS constant, 53–54, 92, 238
- _SC_SHELL constant, 53
- _SC_STREAM_MAX constant, 42
- _SC_SYMLINK_MAX constant, 42
- _SC_THREAD_ATTR_STACKADDR constant, 391
- _SC_THREAD_ATTR_STACKSIZE constant, 391
- _SC_THREAD_DESTRUCTOR_ITERATIONS constant, 388
- _SC_THREAD_KEYS_MAX constant, 388
- _SC_THREAD_PROCESS_SHARED constant, 394
- _SC_THREADS constant, 356
- _SC_THREAD_SAFE_FUNCTIONS constant, 401
- _SC_THREAD_STACK_MIN constant, 388
- _SC_THREAD_THREADS_MAX constant, 388
- _SC_TTY_NAME_MAX constant, 42
- _SC_TZNAME_MAX constant, 42
- _SC_V6_ILP32_OFF32 constant, 67
- _SC_V6_ILP32_OFFBIG constant, 67
- _SC_V6_LP64_OFF64 constant, 67
- _SC_V6_LP64_OFFBIG constant, 67
- _SC_VERSION constant, 49, 53
- _SC_XOPEN_CRYPT constant, 53
- _SC_XOPEN_LEGACY constant, 53
- _SC_XOPEN_REALTIME constant, 53
- _SC_XOPEN_REALTIME_THREADS constant, 53
- _SC_XOPEN_VERSION constant, 53–54
- <search.h> header, 30
- sed program, 887
- Seebass, S., 889
- seek function, 64
- SEEK_CUR constant, 64, 148, 446, 454–455
- seekdir function, 120–125, 412
 - definition of, 120
- SEEK_END constant, 64, 148, 446, 454–455, 747
- SEEK_SET constant, 64, 148, 446, 454–455, 458, 491, 873–874
- SEGV_ACCERR constant, 327
- SEGV_MAPERR constant, 327
- select function, 305–306, 318, 339–340, 411, 441, 474–481, 493, 521, 542, 544, 548, 563–564, 582, 621, 624–626, 628, 678, 696, 707, 767–768, 779–780, 870–871, 875, 878, 880–881
 - definition of, 475
- Seltzer, M., 710, 889–890
- semaphore, 496, 527–533
 - adjustment on exit, 532–533
 - locking versus record locking timing, 533
- <semaphore.h> header, 30
- sembuf structure, 531
- semctl function, 520, 524, 528–529, 532
 - definition of, 529
- semget function, 518–519, 528–529
 - definition of, 529
- semid_ds structure, 528–530
- semop function, 412, 521, 529–533
 - definition of, 530
- sem_post function, 306

- sem_timedwait function, 411
- semun union, 529
- SEM_UNDO constant, 531–533
- sem_wait function, 411
- send function, 306, 411, 548, 565–566, 570, 581–582
 - definition of, 565
- send_err function, 603, 615, 618–619, 628, 844
 - definition of, 603–604
- send_fd function, 603–604, 608, 611, 615, 618–619, 628, 844
 - definition of, 603–604, 608, 611
- sendmsg function, 306, 411, 566, 568, 606, 608–609, 612
 - definition of, 566
- sendto function, 306, 411, 566, 575, 577, 579
 - definition of, 566
- S_ERROR constant, 482
- serv_accept function, 592–593, 598, 601, 621, 625–627, 844
 - definition of, 592–593, 599
- servent structure, 555
- Service Access Facility, *see* SAF
- serv_listen function, 592–593, 597, 621, 625–626, 844
 - definition of, 592, 597
- session, 270–271
 - ID, 215, 234, 271, 286, 423–424
 - leader, 271–273, 286, 294, 424–425, 428, 685, 691–692, 707, 882
 - process group ID, 279
- session structure, 286, 294, 424
- set
 - descriptor, 475, 477, 493, 875
 - signal, 311, 318–320, 493, 875
- SETALL constant, 530, 532
- setasync function, definition of, 881
- setbuf function, 136–137, 139, 159, 229–230, 661, 872
 - definition of, 136
- setegid function, 241
 - definition of, 241
- setenv function, 194, 232, 402
 - definition of, 194
- seteuid function, 241
 - definition of, 241
- set_fl function, 82, 442–443, 458, 844, 876
 - definition of, 81
- setgid function, 237, 241, 264, 306
 - definition of, 237
- setgrent function, 167–168, 402, 412
 - definition of, 167
- set-group-ID, 91–92, 95, 100–101, 103, 119, 130, 215, 235, 292, 456, 508, 685
 - saved, 53, 91–92
- setgroups function, 168
 - definition of, 168
- sethostent function, 412, 553
 - definition of, 553
- sethostname function, 173
- setitimer function, 293, 295, 297, 354, 875
- setjmp function, 179, 195, 197–201, 206, 314–315, 318, 329–330, 333, 354, 867
 - definition of, 197
- _setjmp function, 330, 333
- <setjmp.h> header, 27
- setkey function, 402
- setlogmask function, 430–431
 - definition of, 430
- setnetent function, 412, 554
 - definition of, 554
- setpgid function, 269, 306
 - definition of, 269
- setprotoent function, 412, 554
 - definition of, 554
- setpwent function, 164–165, 402, 412
 - definition of, 164
- setregid function, 240–241
 - definition of, 240
- setreuid function, 240
 - definition of, 240
- setrlimit function, 52, 202, 354
 - definition of, 202
- setservent function, 412, 555
 - definition of, 555
- setsid function, 269, 271, 273, 286, 306, 424–425, 427, 683, 692–693
 - definition of, 271
- setsockopt function, 306, 579, 581, 613
 - definition of, 579
- setspent function, 166
 - definition of, 166
- settimeofday function, 173
- setuid function, 92, 237–241, 264, 306, 779
 - definition of, 237
- set-user-ID, 91–92, 95, 97, 100–101, 103, 119, 130, 166, 215, 235, 238–240, 249, 292, 508, 541–542, 615, 685, 689, 707, 867
 - saved, 53, 91–92, 238–241, 260, 264, 312, 866
- setutxent function, 402, 412
- SETVAL constant, 530, 532
- setvbuf function, 136–137, 139, 159, 202, 514, 877
 - definition of, 136
- SGI (Silicon Graphics, Inc.), 36

- SGID, *see* set-group-ID
shadow passwords, 165–166, 178, 861
<shadow.h> header, 170
S_HANGUP constant, 482
Shannon, W. A., 487, 887
shared
 libraries, 188–189, 207, 719, 863, 885
 memory, 496, 533–540
sharing, file, 70–73, 213
shell, *see* Bourne shell, Bourne-again shell, C shell, Korn shell
SHELL environment variable, 193, 264, 701
shell, job-control, 270, 274, 280, 283, 300, 333, 350, 699
shell layers, 274
shells, 3
S_HIPRI constant, 482
shmat function, 521, 535–538
 definition of, 536
shmatt_t data type, 534
shmctl function, 520, 524, 535–537
 definition of, 535
shmdt function, 536
 definition of, 536
shmget function, 518–519, 534, 537
 definition of, 534
shmid_ds structure, 534–536
SHMLBA constant, 536
SHM_LOCK constant, 535
SHM_RDONLY constant, 536
SHM_RND constant, 536
SHRT_MAX constant, 38
SHRT_MIN constant, 38
shutdown function, 306, 548–549, 567
 definition of, 548
SHUT_RD constant, 548
SHUT_RDWR constant, 548
SHUT_WR constant, 548
SI_ASYNCIO constant, 327
S_IFBLK constant, 124
S_IFCHR constant, 124
S_IFDIR constant, 124
S_IFIFO constant, 124
S_IFLNK constant, 107, 124
S_IFMT constant, 91
S_IFREG constant, 124
S_IFSOCK constant, 124, 596
sig2str function, 353
 definition of, 353
SIG2STR_MAX constant, 353
SIGABRT signal, 218, 222–223, 256, 289, 292–295, 340–342, 354, 867
sigaction function, 57, 298, 301, 304–306, 308, 310–311, 324–329, 341, 344–345, 349, 414, 427, 436, 438, 482, 576, 880
 definition of, 324
sigaction structure, 324, 328–329, 341, 344, 348, 426, 436, 438, 576
sigaddset function, 306, 319, 322, 334, 336, 338, 344, 349, 351, 415, 438, 661, 875
 definition of, 319–320
SIGALRM signal, 289–290, 292–293, 305, 307, 313–315, 317–318, 322, 328–329, 331–332, 339, 348–349, 576
sigaltstack function, 326
sig_atomic_t data type, 57, 330, 332, 336–337, 697
SIG_BLOCK constant, 321, 323, 334, 336, 338, 345, 349, 415, 436, 661
SIGBUS signal, 292–293, 327, 489, 491
SIGCANCEL signal, 292–293
SIGCHLD signal, 220, 264, 291–293, 307–308, 310, 326–327, 342–345, 349–350, 430, 473, 507, 682, 866, 880
 semantics, 308–310
SIGCLD signal, 293, 308–311
SIGCONT signal, 276, 283, 292–293, 312, 349–350, 352
sigdelset function, 306, 319, 341, 349, 875
 definition of, 319–320
SIG_DFL constant, 299, 308, 325–326, 340–341, 350–351, 436
sigemptyset function, 306, 319, 322, 328–329, 334, 336–337, 344, 349, 351, 415, 427, 436, 438, 576, 661, 875
 definition of, 319
SIGEMT signal, 292–293
SIG_ERR constant, 19, 300, 309, 315–318, 322–323, 328–331, 334, 336–337, 343, 511, 587, 669, 671, 697, 844
sigfillset function, 306, 319, 341, 436, 875
 definition of, 319
SIGFPE signal, 18, 222–223, 292, 294, 327
SIGFREEZE signal, 292, 294
Sigfunc data type, 328–329, 844
SIGHUP signal, 283–284, 292, 294, 427, 434–439, 508, 778, 793
SIG_IGN constant, 299, 308, 325, 341, 344, 350, 427, 844
SIGILL signal, 292, 294, 326–327, 340
SIGINFO signal, 292, 294, 642, 649
siginfo structure, 226, 326, 328, 354
<siginfo.h> header, 352
siginfo_t structure, 325

- SIGINT signal, 18-19, 275, 290, 292, 294, 296, 315-316, 322, 333-336, 339, 342-345, 347, 415-416, 508, 639, 641, 645, 648-649, 661-662, 669, 872, 874
- SIGIO signal, 79, 292, 294-295, 473, 481-482, 583
- SIGIOT signal, 292, 295, 340
- sigismember function, 306, 319, 322-323, 875
definition of, 319-320
- sigjmp_buf data type, 330
- SIGKILL signal, 253, 256, 291-292, 295, 299, 321, 353, 699
- siglongjmp function, 201, 307, 329-333, 340
definition of, 330
- SIGLWP signal, 292, 295
- signal function, 18-19, 57, 284, 298-302, 304-310, 314-318, 322-324, 328-331, 334, 336-337, 343, 351, 482, 511, 587, 669, 671, 880
definition of, 298, 328
- signal mask, 311
- signal set, 311, 318-320, 493, 875
- <signal.h> header, 27, 222, 290, 299, 319-320, 353
- signal_intr function, 305, 329, 339, 354, 481, 697, 844, 872
definition of, 329
- signals, 18-19, 289-354
blocking, 310
delivery, 310
generation, 310
generation, pseudo terminal, 706
job-control, 349-352
null, 290, 312
pending, 310
queueing, 311, 324
reliable, 310-311
unreliable, 301-303
- signal_thread function, 793
- sigpause function, 306, 411
- sigpending function, 306, 311, 322-324
definition of, 322
- SIGPIPE signal, 290, 292, 295, 469, 499, 511-512, 515, 518, 543, 587-588, 778, 878
- SIGPOLL signal, 292, 295, 327, 473, 481-482
- sigprocmask function, 306, 311, 314, 318, 320-323, 334-336, 338, 341, 345, 349, 351, 413, 415, 661
definition of, 320
- SIGPROF signal, 292, 295
- SIGPWR signal, 292, 294-295
- sigqueue function, 306, 327-328
- SIGQUIT signal, 275, 292, 296, 322-323, 336, 342, 344-345, 347, 415-416, 508, 641, 649, 662, 669
- SIGSEGV signal, 290, 292, 296, 307-308, 311, 327, 489
- sigset function, 304-306, 308
- sigsetjmp function, 201, 307, 329-333
definition of, 330
- SIG_SETMASK constant, 321, 323, 335-336, 338, 341, 345, 349, 415, 661
- sigset_t data type, 57, 311, 319, 321-322, 334, 336-337, 341, 344, 348, 351, 414-415, 661
- SIGSTKFLT signal, 292, 296
- SIGSTOP signal, 291-292, 296, 299, 321, 349-350
- SIGSUSP signal, 649
- sigsuspend function, 306, 314, 333-340, 349, 411
definition of, 334
- SIGSYS signal, 292, 296
- SIGTERM signal, 291-292, 296, 300, 435, 437-439, 669, 697-698, 707, 778, 793, 882
- SIGTHAW signal, 292, 296
- sigtimedwait function, 411
- SIGTRAP signal, 292, 296, 326-327
- SIGTSTP signal, 275, 283-284, 292, 296, 349-352, 640, 642, 661, 699
- SIGTTIN signal, 275-276, 279, 284, 292, 296-297, 349-350
- SIGTTOU signal, 276-277, 292, 297, 349-350, 651
- SIG_UNBLOCK constant, 321, 323, 351
- SIGURG signal, 79, 290, 292, 295, 297, 482, 581
- SIGUSR1 signal, 292, 297, 300, 322, 330, 332-335, 337-339, 473
- SIGUSR2 signal, 292, 297, 300, 337-339
- sigvec function, 304-305
- SIGVTALRM signal, 292, 297
- sigwait function, 411, 413-416, 435, 437, 793
definition of, 413
- sigwaitinfo function, 411
- SIGWAITING signal, 292, 297
- SIGWINCH signal, 286, 292, 297, 670-671, 706-707
- SIGXCPU signal, 203, 292, 297-298
- SIGXFSZ signal, 203, 292, 298, 354, 868
- SIGXRES signal, 292, 298
- Silicon Graphics, Inc., *see* SGI
- SI_MSGQ constant, 327
- Single UNIX Specification, *see* SUS
Version 3, *see* SUSv3
- single-instance daemons, 432-434
- S_INPUT constant, 482
- SIOCSPGRP constant, 583
- SI_QUEUE constant, 327
- S_IRGRP constant, 93, 97, 100, 130, 433, 592, 844
- S_IROTH constant, 93, 97, 100, 130, 433, 592, 844
- S_IRUSR constant, 93, 97, 100, 130, 433, 592, 687, 690, 844

- S_IRWXG constant, 100, 599
- S_IRWXO constant, 100, 599
- S_IRWXU constant, 100, 599
- S_ISBLK function, 89–90, 129
- S_ISCHR function, 89–90, 129, 658
- S_ISDIR function, 89–91, 123, 658
- S_ISFIFO function, 89–90, 497, 514
- S_ISGID constant, 92, 100, 130, 458
- S_ISLNK function, 89–90
- S_ISREG function, 89–90
- S_ISSOCK function, 89–91, 599
- S_ISUID constant, 92, 100, 130
- S_ISVTX constant, 100–102, 130
- SI_TIMER constant, 327
- SI_USER constant, 327
- S_IWGRP constant, 93, 97, 100, 130, 592, 687, 690
- S_IWOTH constant, 93, 97, 100, 130, 592
- S_IWUSR constant, 93, 97, 100, 130, 433, 592, 687, 690, 844
- S_IXGRP constant, 93, 100, 130, 458, 844
- S_IXOTH constant, 93, 100, 130, 844
- S_IXUSR constant, 93, 100, 130, 844
- size, file, 103–105
- size program, 188–189, 207
- sizeof operator, 213
- size_t data type, 57–58, 68, 479, 854
- sleep function, 212, 216, 225, 228, 253, 255, 284, 306, 309, 314–316, 323, 346–349, 353–354, 359, 363–364, 400, 411, 419, 477, 493, 562, 866–868, 870, 873, 878
 - definition of, 347–348, 871
- sleep_us function, 493, 844
 - definition of, 875
- S_MSG constant, 482
- SNDPIPE constant, 469
- SNDZERO constant, 469
- snprintf function, 149, 689, 849, 851
 - definition of, 149
- Snyder, G., 889
- sockaddr structure, 551–553, 561–562, 564, 577–578, 580, 596, 598–601
- sockaddr_in structure, 551–552, 559
- sockaddr_in6 structure, 551–552
- sockaddr_un structure, 595–601
- socketmark function, 582
 - definition of, 582
- SOCK_DGRAM constant, 547, 558, 563, 567, 576, 578
- socket
 - addressing, 549–561
 - descriptors, 546–549
 - file descriptor passing, 606–614
 - I/O, asynchronous, 582–583
 - I/O, nonblocking, 563–564, 582–583
 - mechanism, 89, 496, 545–584
 - options, 579–581
 - socket function, 138, 306, 546–547, 564, 569, 576, 581, 597–598, 600–601, 797
 - definition of, 546
 - socketpair function, 138, 306, 594–595
 - definition of, 594
 - sockets, UNIX domain, 594–601
 - timing, 527
 - socklen_t data type, 562, 564, 577, 580
 - SOCK_RAW constant, 547, 558
 - SOCK_SEQPACKET constant, 547, 558, 561, 564, 567, 581
 - SOCK_STREAM constant, 295, 547, 558, 561, 564, 567, 569–571, 574, 581, 595–597, 600
- Solaris, xxi, xxiii, 3–4, 26–27, 29–30, 35–36, 38, 48, 55–58, 60, 62, 72, 84, 95, 101–105, 112–113, 119, 121–122, 128, 162, 166, 169–172, 176, 190–191, 193–194, 204, 206, 227, 264, 266, 268, 271, 278, 290, 292–298, 304–305, 308, 310, 326, 330, 348, 352–353, 357, 360, 430, 445, 455–457, 459–461, 464, 466, 471, 474–475, 492, 496, 521, 523, 525, 527, 529, 534–535, 538, 548, 550, 563, 566–568, 583, 585, 595, 610, 635–638, 644–651, 664, 676–677, 682–683, 685, 689, 692, 705–707, 710, 749, 876, 889
- SOL_SOCKET constant, 579, 581, 607–608, 612–614
- solutions to exercises, 853–883
- SOMAXCONN constant, 563
- SO_OOBINLINE constant, 582
- SO_PASSCRED constant, 613
- SO_REUSEADDR constant, 580–581
- source code, availability, xxvi
- S_OUTPUT constant, 482
- Spafford, G., 165, 232, 273, 887
- spawn function, 216
 - <spawn.h> header, 30
- s-pipe, 603, 615–616, 618, 620–621
- s_pipe function, 587–589, 595, 617, 704, 844
 - definition of, 589, 595
- spooling, printer, 757–758
- sprintf function, 149, 511, 570, 577, 600, 617, 619, 621, 628, 725, 738, 807
 - definition of, 149
- spwd structure, 861
- squid login name, 162
- S_RDBAND constant, 482
- S_RDNORM constant, 482
- sscanf function, 151, 511, 513, 764
 - definition of, 151
- SSIZE_MAX constant, 39, 68

- ssize_t data type, 39, 57, 68
- stack, 187, 197
- stackaddr attribute, 389, 391
- stacksize attribute, 389, 391
- standard error, 8, 135, 572
- standard error routines, 846–851
- standard input, 8, 135
- standard I/O
 - alternatives, 159
 - buffering, 135–137, 213, 217, 247, 342, 513–514, 680, 718
 - efficiency, 143–145
 - implementation, 153–155
 - library, 9, 133–160
 - streams, 133–134
 - versus unbuffered I/O, timing, 144
- standard output, 8, 135, 572
- standards, 25–33
 - conflicts, 56–57
- START terminal character, 638, 640–642, 646, 649, 653
- stat function, 4, 7, 62, 87–88, 91–92, 100, 113–114, 116, 118, 121, 128, 130–131, 306, 542, 548, 584, 599–600, 658, 856, 858
 - definition of, 87
- stat structure, 87–89, 92, 103, 107, 130, 137, 155, 457, 490, 497, 514, 519, 542, 599, 657–659
- static variables, 201
- STATUS terminal character, 638, 642, 647, 649, 663
- <stdarg.h> header, 27, 151–152, 721, 724
- <stdbool.h> header, 27
- __STDC__ constant, 56
- <stddef.h> header, 27, 597
- stderr variable, 135, 443, 695, 849
- STDERR_FILENO constant, 60, 135, 573–574, 603, 606, 610, 614, 693
- stdin variable, 10, 135, 144, 196, 198, 512–513, 588, 616
- STDIN_FILENO constant, 8–9, 60, 64, 69, 135, 284, 351, 443, 471, 501, 506, 511–512, 574, 588, 617–618, 639, 644, 669, 671, 693–695, 697, 704–705
- <stdint.h> header, 27, 551
- <stdio.h> header, 10, 27, 38, 50, 135, 137, 141, 153, 155–157, 654, 721, 843
- <stdlib.h> header, 27, 190, 843
- stdout variable, 10, 135, 144, 229–230, 849, 864, 872
- STDOUT_FILENO constant, 8–9, 60, 69, 135, 212, 217, 351, 443, 471, 499, 506, 511–512, 569, 573–575, 588, 616–618, 693, 697, 704–705, 864
- Stevens, D. A., xxviii
- Stevens, E. M., xxviii
- Stevens, S. H., xxviii
- Stevens, W. R., xx–xxii, xxviii, 147, 266, 429, 478, 545, 677, 757, 890
- sticky bit, 100–102, 109, 130
- stime function, 173
- Stonebraker, M. R., 709, 890
- STOP terminal character, 638, 640–642, 646, 649, 653
- str2sig function, 353
 - definition of, 353
- strace program, 457
- Strang, J., 672, 890
- strbuf structure, 462, 471, 605
- strchr function, 733
- stream orientation, 134
- STREAM_MAX constant, 38–39, 42, 48
- STREAMS, 30–32, 83–84, 86, 133, 328, 441, 460–474, 479, 481–482, 485, 493, 495–496, 514, 518, 522, 527, 543, 548, 585, 600, 603–604, 610, 615, 629, 676–677, 680–681, 683, 685, 689, 705, 878, 889
 - clone device, 683
 - file descriptor passing, 604–606
 - ioctl operations, 464
 - Linux, 496
 - messages, 462
 - read mode, 470
 - write mode, 468
- STREAMS module
 - connld, 518, 590, 592, 600
 - ldterm, 468, 676, 685
 - pkct, 676, 705
 - ptem, 468, 676, 685
 - pts, 468
 - redirmod, 468
 - ttcompat, 468, 676, 685
- streams, standard I/O, 133–134
- STREAMS-based pipes, 585–594
 - mounted, 495, 514, 518
 - timing, 527
- strerror function, 15–16, 24, 352, 402, 412, 431, 433, 438, 556, 569–570, 572–573, 576–577, 619, 628, 846, 849, 851, 853–854, 873–874
 - definition of, 15
- strerror_r function, 402
- strftime function, 174–176, 178, 246, 862
 - definition of, 176
- <string.h> header, 27, 843
- <strings.h> header, 30
- strip program, 863
- strlen function, 11, 213

- str_list structure, 466–467
- str_mlist structure, 466–467
- Strong, H. R., 710, 715, 886
- <stropts.h> header, 30, 464, 480, 482
- strrecvfd structure, 593, 605
- strsignal function, 352
 - definition of, 352
- strtok function, 402, 619–620
- strtok_r function, 402
- stty program, 276, 651–652, 662, 673, 882
- Stumm, M., 159, 492, 888
- S_TYPEISMQ function, 89
- S_TYPEISSEM function, 89
- S_TYPEISSHM function, 89
- su program, 431
- submit_file function, 771
- SUID, *see* set-user-ID
- Sun Microsystems, xxiii, 33, 71, 683, 705, 890
- SunOS, xxvii, 33, 188, 305, 329, 533
- superuser, 16
- supplementary group ID, 18, 39, 91–92, 94, 101, 103, 167–168, 214, 234, 241
- SUS (Single UNIX Specification), xxii, 28–33, 41, 52–56, 58–59, 61, 66–67, 74, 83, 88, 99–100, 102, 105, 121–122, 125–126, 133, 147, 156–158, 164, 167, 173, 175–176, 178, 193–194, 202–203, 216, 221, 226–227, 240, 243, 269, 271, 274, 286, 291, 297–298, 304, 306, 308, 317, 327, 329, 348, 379, 387, 391–392, 394, 401, 411, 428, 430, 432, 445, 455, 460, 473, 479, 481, 483, 487, 489, 495–496, 518, 520, 522–523, 527–528, 533, 535, 552, 565, 567, 579, 582, 607, 634, 638, 643, 681–683, 709, 773, 858, 887, 890
- SUSP terminal character, 638, 640, 642, 648, 661
- SUSv3 (Single UNIX Specification, Version 3), 32, 36, 49, 56
- SVID (System V Interface Definition), xix, 32, 34, 885
- SVR2, 62, 171, 462, 672
- SVR3, 119, 183, 274, 456, 460–461, 474, 479, 846
- SVR3.0, xxvii
- SVR3.1, xxvii
- SVR3.2, xxvii, 36, 248
- SVR4, xxvii, 3, 21, 33–37, 48, 72, 112, 171, 191, 266, 271, 285, 428, 460–461, 474, 479, 481, 483, 592, 681, 709
- swapper process, 210
- S_WRBAND constant, 482
- S_WRNORM constant, 482
- symbolic link, 88–89, 102–103, 107, 110, 112–114, 121, 127, 131, 170, 856–857
- symlink function, 115, 306
 - definition of, 115
- SYMLINK_MAX constant, 39, 43, 48
- SYMLoop_MAX constant, 39, 42, 48
- sync function, 59, 77–78
 - definition of, 77
- sync program, 77
- synchronization mechanisms, 82–83
- synchronous write, 61, 82–83
- <sys/acct.h> header, 251
- sysconf function, 20, 37, 39–54, 57–58, 66, 92, 183, 203, 238, 257–258, 306, 356, 387–388, 391, 394, 401, 489, 571, 573, 578, 778, 855
 - definition of, 41
- <sys/conf.h> header, 466
- sysctl program, 291, 521
- sysdef program, 521
- <sys/disklabel.h> header, 84
- <sys/filio.h> header, 84
- <sys/ipc.h> header, 30, 520
- <sys/iso/signal_iso.h> header, 290
- syslog function, 412, 425, 428–433, 435–439, 570–574, 577–578, 849, 851, 871
 - definition of, 430
- syslogd program, 425, 429–430, 432, 434, 439
- <syslog.h> header, 30
- <sys/mkdev.h> header, 128
- <sys/mman.h> header, 29
- <sys/msg.h> header, 30
- <sys/mtio.h> header, 84
- <sys/param.h> header, 49–50
- <sys/resource.h> header, 30
- <sys/select> header, 474
- <sys/select.h> header, 29, 477, 874
- <sys/sem.h> header, 30
- <sys/shm.h> header, 30
- sys_siglist variable, 352
- <sys/signal.h> header, 290
- <sys/socket.h> header, 29, 563
- <sys/sockio.h> header, 84
- <sys/stat.h> header, 29, 91
- <sys/statvfs.h> header, 30
- <sys/sysmacros.h> header, 128
- system calls, 1, 21
 - interrupted, 303–305, 317–318, 326, 329, 339, 481
 - restarted, 304–305, 317–318, 326, 329, 481, 660
 - tracing, 457
 - versus functions, 21–23
- system function, 23, 119, 209, 231, 246–250, 258–260, 323, 342–347, 353, 411, 500, 504, 866, 878

- definition of, 246–247, 344
 - return value, 346
 - system identification, 171–173
 - system process, 210, 312
 - System V, xxi, 83, 441–442, 445, 460, 462, 473, 479, 481, 493, 681, 685
 - System V Interface Definition, *see* SVID
 - <sys/timeb.h> header, 30
 - <sys/time.h> header, 30, 474
 - <sys/times.h> header, 29
 - <sys/ttycom.h> header, 84
 - <sys/types.h> header, 29, 56, 128, 474, 518
 - <sys/uiio.h> header, 30
 - <sys/un.h> header, 29, 595
 - <sys/utsname.h> header, 29
 - <sys/wait.h> header, 29, 221
 - sysyconf function, 57
-
- TAB0 constant, 651
 - TAB1 constant, 651
 - TAB2 constant, 651
 - TAB3 constant, 650–651
 - TABDLY constant, 637, 644, 649–651
 - Tankus, E., xxviii
 - tar program, 117, 125, 131–132, 858–859
 - <tar.h> header, 29
 - tcdrain function, 297, 306, 411, 637, 653
 - definition of, 653
 - tcflag_t data type, 634
 - tcflow function, 297, 306, 637, 653
 - definition of, 653
 - tcflush function, 135, 297, 306, 633, 637, 653
 - definition of, 653
 - tcgetattr function, 306, 635, 637, 639, 643–644, 651–652, 655, 661, 665–667, 695–696
 - definition of, 643
 - tcgetpgrp function, 273–274, 306, 634, 637
 - definition of, 273
 - tcgetsid function, 273–274, 634, 637
 - definition of, 274
 - TCIFLUSH constant, 653
 - TCIOFF constant, 653
 - TCIOFLUSH constant, 653
 - TCION constant, 653
 - TCOFLUSH constant, 653
 - TCOOFF constant, 653
 - TCOON constant, 653
 - TCSADRAIN constant, 643
 - TCSAFLUSH constant, 639, 643, 661, 665–667
 - TCSANOW constant, 643–644, 693, 696
 - tcsendbreak function, 297, 306, 637, 642, 653–654
 - definition of, 653
 - tcsetattr function, 297, 306, 633, 635, 637, 639, 643–644, 651–652, 661, 665–667, 693, 696, 703
 - definition of, 643
 - tcsetpgrp function, 273–274, 276, 278, 297, 306, 634, 637
 - definition of, 273
 - tee program, 516
 - tell function, 64
 - TELL_CHILD function, 229–230, 337, 451, 458, 493, 501, 503, 539, 845
 - definition of, 338, 502
 - tellmdir function, 120–125
 - definition of, 120
 - TELL_PARENT function, 229, 337, 451, 493, 501, 503, 539, 845, 876
 - definition of, 338, 502
 - TELL_WAIT function, 229–230, 337, 451, 458, 493, 501, 539, 845, 876
 - definition of, 337, 502
 - telnet program, 472, 703, 706
 - telnetd program, 267, 472–473, 677, 699, 866, 882
 - tempfile function, 158
 - tempnam function, 155–160
 - definition of, 157
 - TENEX C shell, 3
 - TERM environment variable, 193, 263, 265
 - termcap, 672–673, 890
 - terminal
 - baud rate, 652–653
 - canonical mode, 660–663
 - controlling, 61, 215, 234, 251, 268, 271–274, 276, 278–279, 281, 284, 286–287, 294, 296–297, 349, 423–425, 428, 439, 463, 468, 640, 645, 651, 654, 660, 662, 676, 683, 685, 689, 691–692, 846, 890
 - identification, 654–660
 - I/O, 631–673
 - line control, 653–654
 - logins, 261–266
 - mode, cbreak, 632, 664, 668, 673
 - mode, cooked, 632
 - mode, raw, 632, 664, 668, 673, 696, 699
 - noncanonical mode, 663–670
 - options, 643–651
 - parity, 648
 - process group ID, 278, 423–424
 - special input characters, 638–642
 - window size, 286, 297, 670–672, 691, 706–707
 - termination, process, 180–184
 - terminfo, 672–673, 887, 890

- termio structure, 634
- <termio.h> header, 634
- termios structure, 286, 634, 637–639, 643–644, 652–653, 655, 661, 663–666, 668, 691–692, 694, 696, 702, 706–707, 844–845, 882
- <termios.h> header, 29, 84, 634
- text segment, 186
- <tgmath.h> header, 27
- The Open Group, xxii, 32, 176, 887
- Thompson, K., 71, 165, 211, 709, 889–890
- thread_init function, 404
- threads, 13, 27, 211, 355–386, 540
 - concepts, 355–357
 - control, 387–421
 - creation, 357–360
 - synchronization, 368–385
 - termination, 360–368
- thundering herd, 869
- tick, clock, 20, 42, 48, 57, 251–252, 257
- time
 - and date functions, 173–176
 - calendar, 20, 24, 57, 117, 173–175, 246, 251–252
 - process, 20, 24, 57, 257–259
 - values, 20
- time function, 173, 246, 306, 331, 599–600, 862, 871
 - definition of, 173
- time program, 20
- TIME terminal value, 647, 663–664, 668, 673, 882
- <time.h> header, 27, 57
- timer_getoverrun function, 306
- timer_gettime function, 306
- timer_settime function, 306, 327
- times, file, 115–116, 493
- times function, 42, 57, 257–258, 306, 484
 - definition of, 257
- timespec structure, 383, 398–399, 478
- time_t data type, 20, 57, 173, 175, 178, 854
- timeval structure, 173, 383, 398, 475, 478, 768, 871, 875
- timing
 - message queues, 527
 - read buffer sizes, 70
 - read/write versus mmap, 492
 - semaphore locking versus record locking, 533
 - standard I/O versus unbuffered I/O, 144
 - STREAMS-based pipes, 527
 - synchronization mechanisms, 82–83
 - UNIX domain sockets, 527
 - writew versus other techniques, 484
- TIOCGPTN constant, 689
- TIOCGWINSZ constant, 670–671, 695, 845
- TIOCPKT constant, 705
- TIOCPTLCK constant, 690
- TIOCREMOTE constant, 706
- TIOCSCTTY constant, 273, 692–693
- TIOCSIG constant, 706
- TIOCSIGNAL constant, 706
- TIOCSWINSZ constant, 670, 693, 706
- tip program, 673
- TLI (Transport Layer Interface, System V), 889
- tm structure, 174, 862
- TMPDIR environment variable, 157–158, 193
- tmpfile function, 155–159, 340, 412
 - definition of, 155
- TMP_MAX constant, 38, 155–156
- tmpnam function, 38, 155–159, 401, 412
 - definition of, 155
- tms structure, 257–258
- Torvalds, L., 35
- TOSTOP constant, 636, 651
- touch program, 117
- <trace.h> header, 30
- tracing system calls, 457
- transactions, database, 889
- Transport Layer Interface, System V, *see* TLI
- TRAP_BRKPT constant, 327
- TRAP_TRACE constant, 327
- tread function, 767–768
- treadn function, 768
- Trickey, H., 211, 889
- truncate function, 105, 113, 117, 433
 - definition of, 105
- truncation
 - file, 105
 - filename, 62
 - pathname, 62
- truss program, 457
- ttcompat STREAMS module, 468, 676, 685
- tty structure, 286
- tty_atexit function, 665, 696, 844
 - definition of, 668
- tty_cbreak function, 664, 669, 844
 - definition of, 665
- ttymon program, 266
- ttyname function, 127, 257, 402, 412, 655–656, 659, 687
 - definition of, 655, 658
- TTY_NAME_MAX constant, 39, 42, 48
- ttyname_r function, 402, 412
- tty_raw function, 664, 669, 673, 695, 844
 - definition of, 666
- tty_reset function, 664, 669, 844
 - definition of, 667

- tty_termios function, 665, 844
 - definition of, 668
- type attribute, 394
- typescript file, 678, 701
- TZ environment variable, 174, 176, 178, 193, 862
- TZNAME_MAX constant, 39, 42, 48

- UCHAR_MAX constant, 38
- <ucontext.h> header, 30
- ucontext_t structure, 328
- ucred structure, 611, 613
- UFS file system, 48, 55, 62, 105, 108, 119
- UID, *see* user ID
- uid_t data type, 57
- uint16_t data type, 551
- uint32_t data type, 551
- UIN_T_MAX constant, 38
- ulimit program, 51–52, 204
 - <ulimit.h> header, 30
- ULLONG_MAX constant, 38
- ULONG_MAX constant, 38
- UltraSPARC, xxiii
- umask function, 97–100, 204, 306, 425, 427
 - definition of, 97
- umask program, 98–99, 131
- uname function, 171, 178, 306
 - definition of, 171
- uname program, 172, 178
- unbuffered I/O, 8, 59–86
- unbuffered I/O timing, standard I/O versus, 144
- ungetc function, 141–142, 412
 - definition of, 141
- ungetwc function, 412
- uninitialized data segment, 187
- <unistd.h> header, 9, 29, 52, 60, 69, 103, 401, 474, 721, 843
- UNIX Architecture, 1–2
- UNIX domain sockets, 594–601
 - timing, 527
- UNIX System implementations, 33
- Unix-to-Unix Copy, *see* UUCP
- UnixWare, 36, 309–310
- unlink function, 107–114, 117, 131, 157, 306, 340, 412, 456, 515, 592, 598–601, 857, 859, 878
 - definition of, 109
- un_lock function, 449, 725, 728, 845
- unlockpt function, 682–685, 688, 690
 - definition of, 682, 687, 690
- Unrau, R., 159, 492, 888
- unreliable signals, 301–303
- unsetenv function, 194, 402
 - definition of, 194
- update program, 77
- update_jobno function, 782, 795
- uptime program, 568, 570, 572, 574–575, 577, 579, 584
- USER environment variable, 192, 264
- user ID, 16, 237–241
 - effective, 91–92, 94–95, 99, 103, 117, 130, 210, 214, 235, 237–241, 257, 262, 264, 312, 354, 520, 524, 530, 535, 542–543, 593, 600, 605, 771, 861, 866
 - real, 39, 42, 91–92, 95, 203, 210, 214–215, 234–235, 237–241, 251, 257, 262, 264, 312, 354, 541, 685, 867
- USHRT_MAX constant, 38
- usleep function, 411, 493, 875
- /usr/lib/pt_chmod program, 685
- UTC (Coordinated Universal Time), 20, 173, 175–176
- utimbuf structure, 116, 118
- utime function, 116–119, 131, 306, 858
 - definition of, 116
- <utime.h> header, 29
- utmp file, 170–171, 257, 287, 698–699, 866, 871
- utmp structure, 171
- <utmpx.h> header, 30
- utsname structure, 171–172, 178
- UUCP (Unix-to-Unix Copy), 172
- uucp program, 473

- va_end function, 847–848, 850
- va_list data type, 847–850
- /var/account/acct file, 251
- /var/account/pacct file, 251
- /var/adm/pacct file, 251
- <varargs.h> header, 151
- variables
 - automatic, 187, 197, 199, 201, 207
 - global, 201
 - register, 199
 - static, 201
 - volatile, 199, 201, 315, 332
- /var/log/wtmp file, 171
- /var/run/utmp file, 171
- va_start function, 847–848, 850
- VDISCARD constant, 638
- VDSUSP constant, 638
- VEOF constant, 638–639, 664
- VEOL constant, 638, 664
- VEOL2 constant, 638
- VERASE constant, 638

- VERASE2 constant, 638
vfork function, 211, 216–218, 260, 864–865
vfprintf function, 151, 412
 definition of, 151
vfscanf function, 153
 definition of, 153
vfwprintf function, 412
vi program, 350, 457, 459, 632, 671–673, 882
VINTR constant, 638–639
vipw program, 163
VKILL constant, 638
VLNEXT constant, 638
VMIN constant, 663–665, 667
v-node, 71–72, 74, 126, 287, 602, 855, 887
vnode structure, 286–287
Vo, K. P., 125, 159, 887–888, 890
volatile variables, 199, 201, 315, 332
vprintf function, 151, 412, 854
 definition of, 151
VQUIT constant, 638
vread function, 487
VREPRINT constant, 638
vscanf function, 153
 definition of, 153
vsnprintf function, 151, 849, 851
 definition of, 151
vsprintf function, 151, 431
 definition of, 151
vsscanf function, 153
 definition of, 153
VSTART constant, 638
VSTATUS constant, 638
VSTOP constant, 638
VSUSP constant, 638
vsyslog function, 432
 definition of, 432
VT0 constant, 651
VT1 constant, 651
VTDLY constant, 637, 644, 649, 651
VTIME constant, 663–665, 667
VWERASE constant, 638
vwprintf function, 412
vwrite function, 487
 definition of, 227
wait4 function, 227
 definition of, 227
WAIT_CHILD function, 229, 337, 451, 493, 501, 539, 845, 876
 definition of, 338, 502
waitid function, 226–227, 257, 411
 definition of, 226
WAIT_PARENT function, 229–230, 337, 451, 458, 493, 501, 539, 845
 definition of, 338, 502
waitpid function, 11–13, 19, 219–227, 236, 242, 246–249, 257, 259, 261, 270, 276, 291, 304, 306, 345, 411, 458, 500, 507–508, 543–544, 573, 877–878, 880
 definition of, 220
wall program, 685
wc program, 104
<wchar.h> header, 27, 134
wchar_t data type, 57
WCONTINUED constant, 224, 226
WCOREDUMP function, 221–222
wctomb function, 401
wcsrtombs function, 401
wcstombs function, 402
wctomb function, 402
<wctype.h> header, 27
Weeks, M. S., 188, 887
Weinberger, P. J., 71, 243, 709, 885, 890
Weinstock, C. B., 890
WERASE terminal character, 638, 642, 645–647, 663
WEXITED constant, 226
WEXITSTATUS function, 221–222
who program, 171, 699
WIFCONTINUED function, 221
WIFEXITED function, 221–222
WIFSIGNALED function, 221–222
WIFSTOPPED function, 221–222, 224
Williams, T., 285, 890
Wilson, G. A., xxviii
window size
 pseudo terminal, 706
 terminal, 286, 297, 670–672, 691, 706–707
winsize structure, 286, 670–671, 691–692, 694, 696, 707, 845, 882
Winterbottom, P., 211, 889
WNOHANG constant, 224, 226
WNOWAIT constant, 224, 226
W_OK constant, 96
Wolff, R., xxviii
Wolff, S., xxviii
- wait function, 22, 213–214, 219–228, 231, 237, 245, 249, 257, 259, 276, 293, 303–304, 306–310, 326, 343, 345, 347, 411, 430, 459, 508, 544, 878
 definition of, 220
Wait, J. W., xxviii
wait3 function, 227

- WORD_BIT constant, 40
 <wordexp.h> header, 29
 working directory, 7, 13, 43, 49, 107, 125-126, 162,
 193, 215, 234, 291, 426
 worm, Internet, 142
 wprintf function, 412
 Wright, G. R., xxviii
 write
 delayed, 77
 gather, 483, 607
 synchronous, 61, 82-83
 write function, 8-10, 20-21, 57, 59, 61, 65-66,
 68-69, 72, 74-75, 82-85, 117, 135-136, 145,
 155, 159, 212-213, 216, 229, 304, 306,
 317-318, 351, 354, 411, 434, 442-444, 451,
 454-458, 461-463, 468-469, 471, 475, 478,
 484-488, 491-493, 499-500, 502, 511-513,
 515-516, 521, 527, 543, 546, 548, 565, 569, 575,
 587, 603-604, 616-617, 629, 632, 718, 789,
 799, 855-856, 864, 868, 876-878
 definition of, 68
 write mode, STREAMS, 468
 write program, 685
 write_lock function, 449, 453, 458, 781, 845
 writen function, 485-486, 604, 697, 702, 772, 844
 definition of, 485-486
 writev function, 40-42, 304, 411, 441, 483-485,
 493, 548, 566, 607, 617, 621, 718, 737, 739, 795,
 799
 definition of, 483
 writew_lock function, 449, 451, 725, 735, 752,
 845
 wscanf function, 412
 WSTOPPED constant, 226
 WSTOPSIG function, 221-222
 WTERMSIG function, 221-222
 wtmp file, 170-171, 287, 866
 Wulf, W. A., 890
 WUNTRACED constant, 224

 xargs program, 234
 XCASE constant, 651
 Xenix, 33, 445, 685
 xinetd program, 268
 X_OK constant, 96
 X/Open, xxii, 32, 890
 X/Open Portability Guide, 32
 Issue 3, *see* XPG3
 Issue 4, *see* XPG4
 _XOPEN_CRYPT constant, 32, 53
 _XOPEN_IOV_MAX constant, 41

 _XC 32, 53
 _XOPEN_MAX 41
 _XOPEN_PATH_MAX 41
 _XOPEN_REALTIME constant, 32, 53
 _XOPEN_REALTIME_THREADS constant, 32, 53
 _XOPEN_SOURCE constant, 55
 _XOPEN_STREAMS constant, 32
 _XOPEN_UNIX constant, 29, 55
 _XOPEN_VERSION constant, 53, 55
 XPG3 (X/Open Portability Guide, Issue 3), xxvii,
 34, 890
 XPG4 (X/Open Portability Guide, Issue 4), 32
 XSI, 29-32, 52-55, 74, 88, 100, 102, 105, 121-122,
 125, 133, 150, 152, 156-158, 164, 167, 173,
 193-194, 202, 204, 221, 224, 226-227, 240,
 269, 271, 274, 291-292, 297, 304-305, 308,
 317, 325-326, 329, 391, 394, 401, 428,
 430-431, 445, 460, 474, 483, 487-489, 496,
 515, 523-525, 528, 533, 538, 540, 543-544,
 626, 634-635, 637, 645, 647, 649-651, 681,
 683, 686, 709-710, 858
 XSI IPC, 518-522
 XTABS constant, 650-651

 Yigit, O., 710, 890

 zombie, 219-220, 224, 260, 308, 326, 866

