Compatibility Package Features 2

This chapter describes the contents of the Source Compatibility Package. The contents are organized as follows:

- Commands
- System calls
- Signals
- Library routines
- Header files

For more information on the interfaces described here, see the *SunOS Reference Manual*.

Source Compatibility Package Contents

This section lists the contents of the Source Compatibility Package, and then describes the individual interfaces.

Commands

The commands below are described in alphabetical order. Following those descriptions, some common usage of related commands is discussed. These commands are located in `/usr/ucb`.

Table 2-1 below lists the commands available with the Source Compatibility Package.

2

Two versions of some commands exist: one in the default SunOS 5.x command location, and the other in the compatibility package found in `/usr/ucb`. These commands are shown in **`bold`** in the table below. The descriptions in this guide pertain only to the `/usr/ucb` versions, although, where relevant, the differences between the two versions are discussed.

Table 2-1 Commands

Commands		
`Mail`	`hp722`	`renice`
`aedplot`	`hpplot`	`plotreset`
`arch`	`implot`	`rusage`
`atoplot`	`install`	`sed`
`basename`	`ldd`	**`shutdown`**
`bgplot`	`lint`	**`stty`**
`biff`	`ln`	`sum`
`cc`	`logger`	`t300`
`chown`	`lpc`	`t300s`
`crtplot`	`lpq`	`t4013`
`df`	`lpr`	`t450`
`du`	`lprm`	`tek`
`dumbplot`	`lptest`	**`test`**
`echo`	`ls`	**`touch`**
`expr`	`mach`	**`tr`**
`fastboot`	`mail`	`tset`
`fasthalt`	`mkstr`	`ucblinks`
`file`	`pagesize`	`rusers`
`from`	`plot`	`vipw`
`gigiplot`	`plottoa`	`vplot`

Table 2-1 Commands

Commands		
groups	printenv	whereis
hostid	ps	whoami
hostname		

`arch`
Displays your machine architecture. `arch` uses the SunOS 5.x `uname -m` command. `arch` is not available in the default SunOS 5.x release.

`basename`
Strips the full path name off a file. If you follow the path name with a suffix, that suffix is stripped, too:

```
$ basename /usr/src/rundog.c .c

rundog

$
```

`basename` is useful for writing shell scripts. This version parses arguments differently than the SunOS 5.x version.

`biff`
Enables and disables immediate notification of the arrival of mail. When notification is enabled, `biff` displays the header and first few lines of mail messages when they are received. This command is not available in the default SunOS 5.x release.

`cc`
This command is identical to the unbundled C compiler in `/usr/ccs/bin/acc`, except that the following directories are included:

- User-specified include directories and libraries
- The compatibility include files and directories

- The default SunOS 5.x headers and libraries, if unresolved symbols remain

Compatibility package header files and libraries referenced by the compiler (`cc`) and linker (`ld`) are located in `/usr/ucbinclude` and `/usr/ucblib`.

`chown`
Changes the owner of a file. This version allows you to optionally change the group as well. The group may be either a decimal group ID (GID) or a group name found in the GID file.

`df`
Displays the free disk space of currently mounted file systems. This command differs from `/usr/bin/df` in several ways:

- Blocks sizes are reported in units of 1024-bytes in this version of `df`, while `/usr/bin/df` uses 512-byte blocks.
- Perhaps the most obvious difference between the two versions of this command is their output format, as shown by the example below.

```
$ /usr/ucb/df
Filesystem    kbytes       used      avail           capacity      Mounted on
/dev/root     1750592      826496    749032          52%           /
$ df
/             (/dev/root   ):       231024 blocks    105323 files
/proc         (/proc       ):            0 blocks    123 files
```

`du`
Displays the number of disk blocks used by files in a directory. It is useful for seeing how disk space is being used. This version reports disk usage in kilobytes, while the default `du` command reports disk usage in blocks.

`echo`
Echoes arguments to the standard output. This version and the default `echo` command recognize certain escape sequences differently.

`expr`
Evaluates an expression. This version supports the following additional operators and keywords: `match, substr, index, length()`.

`fastboot`
Reboots your machine. This command is not available in the default SunOS 5.x release.

`fasthalt`
Halts your machine. This command is not available in the default SunOS 5.x release.

`file`
Determines the file type by examining the file's contents. This version supports symbolic links. When the –L option is specified for a file that is a symbolic link, file tests the file referenced by the link rather than the link itself.

`from`
Displays mail header information, including the sender and time the mail was received. This command is not available in the default SunOS 5.x release.

`groups`
Displays the groups to which a user belongs. Although the `id` command in the default SunOS 5.x release has been enhanced to perform this function, `groups` is also available for compatibility.

`hostid`
Prints the numeric identifier (in hexadecimal) of the current host. This command is not available in the SunOS 5.x release.

`hostname`
Displays the name of your machine and allows you to rename it. This command is not available in the base release. `uname -n` in the SunOS 5.x release displays the machine name, but does not allow the super-user to rename the machine.

`install`
Installs files into a destination directory. This version has different options and syntax than the SunOS 5.x one.

`ld`
This command is identical to the linker in `/usr/ccs/bin/ld`, except that it ensures that libraries in `/usr/ucblib` are linked in by `ld`, *before* the default SunOS 5.x libraries.

When unresolved symbols remain, the libraries in `/usr/lib` are referenced. Note that `/usr/ucblib/libucb.so` is always linked when this command is invoked.

`lint`
Checks C programs for bugs, argument type errors, and potential portability problems. This command uses the lint libraries under `/usr/ucblib`.

`ln`
Creates links to files or directories. The default `ln` removes an existing target if it has the requisite permissions. Because the SunOS 4.1 behavior is not obtained by default, the compatibility package contains a version of `ln` that does not remove an existing target.

`logger`
Adds entries to a system log. Priorities, tags, and files to write to may be specified.

`lpc`
`lpc`, the line printer control program, is the same as in SunOS 4.1, except that it does not make use of the file `/etc/printcap`. Use `lpc` to start and stop a printer, disable or enable a spooling queue, rearrange the order of printing jobs, or display the status of each printer. This command is not available in SunOS 5.x.

`lpq`

Returns information on all print jobs on the local printer; however, for remote printers, it provides the status only for jobs belonging to the user. `lpq` is the equivalent of the SunOS 5.x command `lpstat`.

The command `lpq` *username* provides the status of all files, on all machines (both local and remote), belonging to the named user. In these examples, user `wesson` sends files `fileX` and `fileY` to the local printer `printerA`, and `fileZ` to a remote printer:

```
$ lpr fileX

$ lpr fileY

$ lpq

Rank Owner Job Files

active smith printerA-111230 file1

1st wesson printerA-111231 fileX

2nd wesson printerA-111232 fileY

3rd jones printerA-111233 file9

$

$ lpr -Pfar_printer fileZ

$ lpq -Pfar_printer

Rank Owner Job Files

4th wesson far_printer-111332 fileZ

$

$ lpq wesson
```

```
printerA:

Rank Owner Job Files

1st wesson printerA-111231 FileX

2nd wesson printerA-111232 FileY

far_printer:

Rank Owner Job Files

4th wesson far_printer-111332 FileZ
$
```

`lpr`
Sends files to the printer. The compatibility package version of `lpr` does not support the `-r` option (which removes a file after spooling it). Nor does it allow for specific fonts to be mounted on font positions 1, 2, 3, or 4. `lpr` is the equivalent of the SunOS 5.x `lp` command.

`lprm`
Removes files from the print queue. This is the SunOS 4.1 equivalent of the SunOS 5.x `cancel` command.

`lptest`
Produces a ripple pattern for testing printers.

`ls`
Lists a file or the contents of a directory. The compatibility package version of `ls` differs from the default `ls` command in two ways:

- The `-l` option does not print the group.
- The `-s` option reports block sizes in blocks of 1024 bytes, instead of 512.

```
$ ls -ls

total 24

8 -rw-r--r-- 1 jah staff 268 Nov 10 21:28 test1
```

```
8 -rw-r--r-- 1 jah staff 121 Nov 10 21:28 test2

8 -rw-r--r-- 1 jah staff 42 Nov 10 21:28 test3

$ /usr/ucb/ls -ls

total 12

4 -rw-r--r-- 1 jah 268 Nov 10 21:28 test1

4 -rw-r--r-- 1 jah 121 Nov 10 21:28 test2

4 -rw-r--r-- 1 jah 42 Nov 10 21:28 test3

$
```

`mach`
Reports the machine type. The SunOS 5.x `uname -p` command provides the same behavior. However, because many scripts and makefiles use `mach`, it has been placed in the compatibility package. This command is not available in default SunOS 5.x release.

`Mail` and `mail`
Reads and sends mail messages. This command is a version of `mailx`(1) with defaults set to behave like the BSD `mail` command. See the `mailx`(1) manual page in the *SunOS Reference Manual* for complete information on the BSD default settings.

`mkstr`
Creates a file of all error messages in a C program source file. This is useful for programs with a large number of diagnostic error messages, it reduces the size of the program, and the system overhead when running the program. This command is not available in the default SunOS 5.x release.

`pagesize`
Reports the size of a page of memory in bytes. This is useful for scripts. This command is not available in the default SunOS 5.x release.

`plot`
Graphics filters for various plotters. The following filters allow you to run `plot` on a variety of terminals: `aedplot`, `atoplot`, `bgplot`, `crtplot`, `dumbplot`, `gigiplot`, `hp7221plot`, `hpplot`, `implot`, `plottoa`, `t300`, `t300s`, `t4013`, `t450`, `tek`, and `vplot`.

`printenv`
Reports environment variables as currently set. It is similar to the `env` command. This command is not available in the default SunOS 5.x release.

`ps`
Displays the status of the current processes. This version displays more information, in a different format, than that of the SunOS 5.x version.

`renice`
Alters the *nice* value of running processes. This behavior is now present in the SunOS 5.x `priocntl` command; however, for SunOS 4.1 compatibility reasons, `renice` is included in the compatibility package. This command is not available in the default SunOS 5.x release.

`reset` and `tset`
SunOS 5.x These commands are similar to `tput reset` and `tput init` respectively; that is, they reset and set terminal-behavior characteristics. However they differ enough to be included here. `reset` and SunOS 5.x `tset` are not available in the default SunOS 5.x release.

`rusage`
Executes the given command and reports on the resource usage. `rusage` reports the total time elapsed (wall clock), the system CPU, the user CPU, plus 11 other fields in a one line summary following the commands output, if any. This command is not available in the default SunOS 5.x release.

`sed`
`/usr/ucb/sed` is similar to `/usr/bin/sed` except it strips initial SPACE and TAB characters from text lines.

`shutdown`
This version of `shutdown` allows you to specify a time and a warning message to be sent. It also sends shutdown messages to other systems if they have mounted filesystems from the machine being shut down. See the `shutdown(1B)` manual page for any limitations on SunOS 4.1 functionality.

`stty` 5.2
The SunOS 5.x version of `stty` incorporates many of the features of SunOS 4.1 `stty`. The compatibility package version is included for full SunOS 4.1 functionality.

`sum`
Calculates checksums in a machine-dependent way. Also, it displays the file in kilobytes instead of 512-byte blocks.

`test`
Evaluates conditional expressions. The SunOS 4.1 and SunOS 5.x versions of `test` differ with respect to the `-f` option. In the SunOS 4.1 release, this option means "is the target not-a-directory?" while in the SunOS 5.x release it means "is the target a plain file?".

`touch`
Updates the access and modification times of a file. The SunOS 5.x version of `touch` does not support the `-f` option, which attempts to force the touch in spite of read and write permissions on the file.

`tr`
Translates characters from the standard input to the standard output. There are two differences between `/usr/bin/tr` and this version:

- In SunOS 4.1, when translating from *string1* to *string2*, if *string2* is shorter than *string1*, *string2* is padded out to the length of *string1* by repeating *string2*'s last character.
- This version does not require enclosing alphanumeric ranges inside square brackets.

`tset`
See `reset`.

`ucblinks`
Creates symbolic links in `/dev` for SunOS 4.1 devices whose names are different in the SunOS 5.x release. When these links are no longer needed, they must be removed by hand.

`users`
Displays a short list of logged-in users. This command is not available in the default SunOS 5.x release. The SunOS 5.x equivalent is `who -q`.

`vipw`
Edits the `/etc/passwd` file, and sets the appropriate locks. `vipw` runs consistency checks on the password file when the edits are complete and the locks have been removed. If these checks fail, `vipw` does not allow the incorrect file to be installed. This command is not available in the default SunOS 5.x release.

`whereis`
Locates the binary, source, and manual page files for the given command. `whereis` searches a standard list of common locations for these files. This command is not available in the default SunOS 5.x release.

`whoami`
Displays your effective user ID; `whoami` works even if you have used `su` to change your current user ID. This command is not available in the default SunOS 5.x release, although it is equivalent to `id`.

Printing Commands

The compatibility package contains five printing commands that are not in the default SunOS 5.x release: `lpr`, `lpq`, `lprm`, `lpc`, and `lptest`. This is how they compare with the base commands:

Table 2-2 Printing Commands

SunOS 4.1 Command	SunOS 5.x Equivalent
lpr	lp
lpq	lpstat
lprm	cancel
lpc	no equivalent
lptest	no equivalent

Here is an overview of some important features:

- There are no SunOS 4.1 commands for administration, maintenance, or accounting of printers; you must use SunOS 5.x schemes. The exception is the `lpc` command, which is included.

- The SunOS 4.1 `printcap` file does not exist. The printer information is now contained in the files in the directories `/etc/lp` and `/etc/lp/printers/`*printer-name*. You can change this information using the `lpadmin` command. For more information on this command, refer to the `lpadmin`(1) manual page in the *SunOS Reference Manual*.
- Printing spooling commands do not depend on the `lpd` daemon, as in SunOS 4.1. Instead, they use an HPI (High Performance Interface) with the SunOS 5.x printing daemon, `lpsched`.

Mail Commands

The SunOS 4.1 version of `mail`, based on `sendmail`, has been included in the Source Compatibility Package; it has a different user interface than the base SunOS 5.x version of `mail`. Additionally, some useful mail facilities are included for compatibility.

Table 2-3 Mail Commands

Commands	Description
biff	immediate notification of mail
from	display mail header lines
mail	send and receive mail messages
Mail	send and receive mail messages

System Management

Because of limitations in underlying SunOS 5.x support, the commands in this section for stopping and starting your system cannot provide full SunOS 4.1 functionality. Nonetheless, they are included here for compatibility, so that existing shell scripts invoking them can still function. For example, the compatibility package versions of `fastboot` and `fasthalt` do not prevent disk checks. See the manual pages on following commands to find out about specific limitations: `fastboot`, `fasthalt`, and `shutdown`.

2

System Calls

The following system calls are implemented in the compatibility package as library routines:

Table 2-4 System Calls

	System Calls	
WIFEXITED	getpagesize	setregid
WIFSIGNALED	getpriority	setreuid
WIFSTOPPED	gettimeofday	settimeofday
flock	killpg	statfs
fstatfs	mctl	syscall
getdtablesize	reboot	wait3
gethostid	sethostname	wait4
gethostname	setpriority	

The following routines reside in `libucb`.

`flock`
Applies or removes an advisory lock on a file. The compatibility version of `flock` has been implemented on top of `fcntl`(2) locking. This version is not completely compatible with the SunOS 4.1 version. The following differences exist:

- To obtain a shared lock (LOCK_SH), read permission on file is required.
- To obtain an exclusive lock (LOCK_EX), write permission on file is required.
- EOPNOTSUPP may not be returned as a result of a call to `flock`.
- Locking a segment that is already locked by the calling process causes the old lock type to be removed and the new lock type to take effect. That was not the case with `flock` in the SunOS 4.1 release.
- Locks are not inherited by a child process in a `fork`(2) function.
- Locks obtained through the flock mechanism under SunOS 4.1 were known only within the system on which they were placed. This is no longer true.

`fstatfs`
Gets statistics about an open file on a mounted file system.

`getdtablesize`
Gets the size of a process's descriptor table. This routine calls the `getrlimit`(2) system call.

`gethostid`
Returns the 32-bit identifier for the current host. This routine calls the `sysinfo`(2) system call.

`gethostname`
Returns the standard hostname for the current processor. This routine calls the `sysinfo`(2) system call.

`getpagesize`
Returns the number of bytes in a page. This routine calls the `sysconf`(3C) routine.

`getpriority`
Gets the process scheduling priority. Returns the highest priority (lowest numerical value) of any of the specified processes. This routine calls the `priocntl`(2) system call.

`gettimeofday`
Gets date and time. This routine calls the default version of `gettimeofday`(3C). Note that this version of `gettimeofday` accepts two arguments for compatibility purposes, although it ignores the second one.

`killpg`
Sends a signal to a process group. This routine calls the `kill`(2) system call.

`mctl`
Applies memory management control functions. This routine calls the `memcntl`(3) routine.

`reboot`
Restart system or halt processor. This routine calls the `uadmin`(2) system call.

`sethostname`
Sets the name of the host machine. This routine calls the `sysinfo`(2) system call.

`setpriority`
Sets the program scheduling priority. The default priority is `0`. It is not possible for the process executing `setpriority` to lower any other process down to its current priority, without appropriate privileges. This routine calls the `priocntl`(2) and `priocntlset`(2) system calls.

`setregid`
Set real and effective group ID's. This routine calls the `setgid`(2) or `setegid` system calls. Exact BSD semantics are not always maintained. The real `gid` is not always changed. Not all cases can be handled.

`setreuid`
Sets real and effective user ID's. Calls SunOS 5.x's `setuid` or `seteuid`. Exact BSD semantics are not always maintained. The real `uid` is not always changed. Not all cases can be handled.

`settimeofday`
Sets date and time of day. This routine calls the `stime`(2) system call. Note that this function accepts two arguments for compatibility purposes, although it ignores the second.

`statfs`
Gets file system statistics about mounted file systems. Calls `statvfs`(2).

`syscall`
Performs indirect system calls.

`wait3`
Alternative system call to `wait`(2) for waiting for a process to stop. This routine calls the `waitid`(2) system call.

`wait4`
Alternative system call to `wait`(2) for waiting for a process to stop. This routine calls the `waitid`(2) system call.

Signals

The SunOS 4.1 signal handlers and default actions are different from the SunOS 5.x signals.

The following routines are in the compatibility package:

Table 2-5 Signals

	Signals	
_longjmp	sigblock	sigpause
_setjmp	siginterrupt	sigsetmask
longjmp	sigmask	sigstack
psignal	signal	sigvec
setjmp		

The SunOS 4.1 style of signal handling is provided in the compatibility package. These routines invoke existing SunOS 5.x signal handling routines. These routines reside in `libucb`.

`_longjmp`, `longjmp`, `_setjmp`, and `setjmp`
In the SunOS 4.1 release, `setjmp` and `longjmp` save and restore the signal mask. This functionality has been preserved with the `sigsetjmp` and `siglongjmp` signals, respectively.

`psignal`
In the SunOS 4.1 release, `psignal` flushes stdio buffers, The SunOS 5.x version of `psignal` does not. This behavior has been preserved in the compatibility version.

`sigblock` and `sigsetmask`
In the SunOS 4.1 release, `sigblock` blocks a signal; `sigsetmask` sets a signal mask. `sigblock` and `sigsetmask` call the `sigprocmask`(2) system call.

`siginterrupt`, `signal`, `sigvec` and `sigaction`
In the SunOS 4.1 release, `siginterrupt` is a library function that allows signal interruptions; `signal` is a library function for simplified signal processing. Both now call `sigvec`, which itself calls the `sigaction` routine.

`sigpause`
Releases block signals; waits for interrupts. Implemented in the compatibility package through `sigsuspend`.

2

Library Routines

The Source Compatibility Package includes a set of compatibility libraries. These libraries can be found in `/usr/ucblib`, and they consist of:

- Routines not in the default SunOS 5.x system
- Routines having different interfaces or different system call "wrappers" than their SunOS 5.x counterparts

The routines in Table 2-6 are available with the compatibility package. Routines shown below in bold also exist in the default system:

Table 2-6 Library Routines

Library Routines		
alphasort	printf	setvbuf
bcmp	rand	sleep
bcopy	random	sprintf
bzero	re_comp	srand
fopen	re_exec	srandom
fprintf	readdir	sys_siglist
freopen	rindex	times
ftime	scandir	ualarm
getwd	setbuf	usleep
index	setbuffer	vfprintf
initstate	setlinebuf	vprintf
nice	setstate	vsprintf
nlist		

Compatibility library routines are placed in one of the following libraries: `libcurses`, `libdbm`, `librpcsoc`, `libtermcap`, and `libucb`.

The descriptions of the routines found in these libraries are divided by the libraries. Many of these routines are not included in the default SunOS 5.x libraries, but are useful enough to be retained for backward compatibility.

The routines in Table 2-6 are included in compatibility libraries for backwards compatibility purposes. At some later date these routines may not be supported and may eventually be discarded.

Routines in `libcurses`

The routines listed in Table 2-7 are available in the `/usr/ucblib` version of `libcurses`. Those routines shown in **`bold`** are also available in the default `libcurses`. The source compatibility version of `libcurses` uses `termcap`, while the base version uses `terminfo`.

Table 2-7 Curses Routines

Curses Routines		
box	**overlay**	**wclrtobot**
delwin	**overwrite**	**wclrtoeol**
endwin	plod	**wdelch**
fgoto	plodput	**wdeleteln**
fullname	**printw**	**werase**
getcap	**scanw**	**wgetch**
gettmode	**scroll**	**wgetstr**
idlok	**setterm**	**winsch**
initscr	**subwin**	**winsertln**
longname	tabcol	**wmove**
mvcur	**touchline**	**wprintw**
mvprintw	touchoverlap	**wrefresh**
mvscanw	**touchwin**	**wscanw**
mvwin	tstp	**wstandend**
mvwprintw	**waddch**	**wstandout**
mvwscanw	**waddstr**	zap
newwin	**wclear**	

2

Routines in `libdbm`

`libdbm` contains the following routines: `dbmclose`, `dbminitdelete`, `fetch`, `firstkey`, `nextkey`, and `store`. These routines maintain key/content pairs in a database using database routines. They are packaged in `/usr/ucblib/libdbm`. For more information, see the appropriate manual pages. These routines are superceded by the `libndbm` routines in the default system.

Routines in `libtermcap`

`libtermcap` contains the following routines: `tgetent`, `tgetnum`, `tgetflag`, `tgetstr`, `tgoto`, and `tputs`.

Routines in `librpcsoc`

`librpcsoc` contains the following routines: `clnttcp_create`, `clntudp_bufcreate`, `clntudp_create`, `rtime`, `svcfd_create`, `svctcp_create`, `svcudp_bufcreate`, `svcudp_create`, and `svcudp_enablecache`.

Routines found in this library expect a socket descriptor as an argument.

Routines in `libucb`

`alphasort` and `scandir`

These routines combine the functions of `readdir` and `qsort` to process directory information. `scandir` reads a directory and builds an array of pointers to directory entries, returning the number of entries in the array and a pointer to the array through the parameter, *namelist*. *namelist* is a pointer to an arrangement of directory structure pointers. If the directory cannot be opened for reading or if `malloc` cannot allocate enough memory to hold all the data structures, `scandir` returns `-1`. `alphasort` is a routine that alphabetically sorts an array of pointers to directory entries (built by the routine, `scandir`).

These routines are not available in the default SunOS 5.x release.

`bcmp`, `bcopy`, and `bzero`
Functionally equivalent to the `memcmp`, `memcpy`, and `memset` routines, see `memory`(3C). `bcopy` copies bytes, handling overlapping strings correctly. `bcmp` compares bytes, returning zero if they are identical, non-zero otherwise. `bzero` zeros out bytes. These routines are not available in the default SunOS 5.x release.

`fopen` and `freopen`
The SunOS 5.x and SunOS 4.1 versions differ in the handling of the `a` type. If mode `a` is specified, the SunOS 5.x version opens the file with `O_APPEND` set, while the SunOS 4.1 version doesn't. Also, the SunOS 4.1version always seeks to the end of the file when mode `a` is specified, while the SunOS 5.x version seeks to the end of the file if update mode is not specified.

`ftime`
Returns a structure that contains elements showing the elapsed time since the epoch (00:00:00, January 1st, 1970). See `ctime`(3C). This routine is not available in the default SunOS 5.x release.

`getwd`
Similar to `getcwd`. `getwd` returns the pathname of the current working directory, or NULL if there is an error. The return value of `getwd` is placed in a character string allocated by the caller. This routine is not available in the default SunOS 5.x release.

`index` and `rindex`
`index` returns a pointer to the first occurrence of a single character in a null-terminated character string. If the character does not occur in the string, `index` returns a NULL pointer. In the SunOS 5.x release, `strchr` replaces `index`, see `string`(3C).

`rindex` returns a pointer to the last occurrence of a single character in a null-terminated string, or a NULL pointer if the letter does not occur in the string. In the SunOS 5.x release, `strrchr` replaces `rindex`, see `string`(3C).

These routines are not available in the default SunOS 5.x release.

`initstate`, `random`, `setstate`, and `srandom`
Random number generator routines relied on by many user programs in the SunOS 4.1 release. These routines are not available in the default SunOS 5.x release.

See `rand`(3C), `srand`(3C), and `drand`(3C) in the *SunOS Reference Manual*.

`nice`
The SunOS 4.1 version always returns 0 if successful. The SunOS 5.x version returns the new `nice` value minus 20.

`nlist`
If unsuccessful, returns the number of symbols that were not located in the symbol table. (The SunOS 5.x version returns –1 if unsuccessful.) Also, `nlist` takes a file name as an argument, rather than a file descriptor.

`printf`, `fprintf`, `sprintf`, `vfprintf`, `vprintf`, and `vsprintf`
The following differences exist for these functions:

- In the SunOS 4.1 release, `sprintf` returns a pointer to its first argument; in the SunOS 5.x release, it returns the count of characters printed.
- In the SunOS 4.1 release, `printf` and `vprintf` always return the number of characters printed or EOF, while the SunOS 5.x version returns only EOF in case of an error.
- `fprintf` and `vfprintf` share the same differences as `printf` and `vprintf` (above); additionally, the buffering scheme for these routines is different for the SunOS 5.x and SunOS 4.1 release.

`rand` and `srand`
The SunOS 5.x returns the generated number shifted to the right by 16 and anded with 0x7fff, while the SunOS 4.1 release returns the generated number anded with 0x7fffffff.

`re_comp` and `re_exec`
These routines are similar to the `regcmp`(3G) and `regex` routines. `re_comp` compiles the regular expression in a string, returning 0 if it compiled successfully or returning a pointer to an error message if it did not. `regcmp`, on the other hand, returns a pointer to the compiled form or NULL if given an incorrect argument. `re_exec` returns 1 if the string *s* matches the last compiled regular expression, 0 if it fails to match, and –1 if the regular expression is invalid. `regex`, however, returns NULL on failure or a pointer to the next unmatched character on success.

`readdir`
Reads a directory entry. The SunOS 4.1 release `readdir` routine returns a pointer to `struct direct`, while the SunOS 5.x version returns a pointer to `struct dirent`.

`setbuffer` and `setlinebuf`
These routines are used to provide control over the buffering used for input and output.

Similar to `setbuf`(3S) and `setvbuf`, `setbuffer` may be used after a stream has been opened but before it is read or written. It uses the array pointer instead of an automatically allocated buffer. If the buffer is the NULL pointer, input/output is completely unbuffered.

`setlinebuf` is used to change the buffering on a stream from block buffered or unbuffered to line buffered. It can be used at any time that the file descriptor is active.

These routines are not available in the default SunOS 5.x release.

`sleep`
The SunOS 5.x version returns the "unslept" amount, that is, the requested time minus the time actually slept. The SunOS 4.1 version does not return anything.

`sys_siglist`
An array containing a message string for each SunOS 4.1 signal number.

`times`
The SunOS 4.1 version returns 0 when successful. The SunOS 5.x version returns the elapsed real time in clock ticks per second from an arbitrary point in the past.

`ualarm` and `usleep`
Simplified interfaces to the system calls handling high-resolution timers. These routines are not available in the default SunOS 5.x release.

Header Files

The Source Compatibility Package header files can be found in `/usr/ucbinclude`. Differences remain in the case of header files, and you should be aware of them; in some cases it was impossible to achieve complete

source compatibility. Some SunOS 4.1 header files do not exist in the default SunOS 5.x release. The header files in **bold** are also available in the default system.

Table 2-8 Header FIles

	Header Files	
a.out.h	struct.h	sys/tty.h
assert.h	sys/dir.h	sys/ttychars.h
curses.h	sys/fcntl.h	sys/types.h
dbm.h	sys/file.h	sys/vfs.h
fcntl.h	sys/ioctl.h	sys/wait.h
regexp.h	sys/param.h	syscall.h
resolv.h	sys/reboot.h	sysexit.h
setjmp.h	sys/resource.h	ufs/quota.h
sgtty.h	sys/rusage.h	unistd.h
signal.h	**sys/signal.h**	vfont.h
stdio.h	**sys/sysmacros.h**	vfork.h
strings.h		

As with the library routines, there are some header files under SunOS 5.x that differ slightly from their SunOS 4.1 counterparts. In those instances, the affected header files are included in this compatibility package with as many additions as possible to keep them compatible with SunOS 4.1. The following list shows those files affected and the reasons for their inclusion in this package:

`a.out.h`
Included for compatibility.

`assert.h`
Included for compatibility.

`curses.h`
Included for compatibility.

`dbm.h`
Used by routines in `libdbm` and user programs invoking `libdbm`.

`fcntl.h`
Included for compatibility.

`regexp.h`
Includes `ctype.h` for compatibility.

`resolv.h`
Included for compatibility.

`setjmp.h`
Includes additional `#defines` and `#includes` for compatibility.

`sgtty.h`
Included for compatibility.

`signal.h`
Contains `sys/signal.h`. Included here for compatibility.

`stdio.h`
Includes SunOS 4.1-specific definitions.

`strings.h`
Contains the same type of information as `string.h`.

`struct.h`
Contains useful macros.

`sys/dir.h`
Included for compatibility.

`sys/fcntl.h`
Includes missing `#defines`.

`sys/file.h`
Includes missing `#defines` such as `L_SET`, and `L_INCR`.

`sys/ioctl.h`
Included for compatibility.

`sys/param.h`
Includes missing `#defines`.

`sys/reboot.h`
Included for compatibility.

`sys/resource.h`
Includes the `rusage` structure missing from the SunOS 5.x version of `resource.h` and used by the `getrusage` function.

sys/rusage.h
Includes the rusage structure.

`sys/signal.h`
Includes SunOS 4.1-specific data structures and definitions.

`sys/sysmacros.h`
Included for compatibility.

`sys/tty.h`
Included for compatibility.

`sys/ttychars.h`
Included for compatibility.

`sys/types.h`
Includes the `sys/sysmacros` header files and additional type definitions.

`sys/vfs.h`
Includes `sys/statfs` header file.

`sys/wait.h`
Includes declarations required by the `wait3` system call wrapper.

`syscall.h`
Included for compatibility.

`sysexits.h`
Contains useful exit codes.

`ufs/quota.h`
Includes `sys/fs.ufs_quota.h`. Added for compatibility.

`unistd.h`
Includes `#defines` missing in the SunOS 5.x version of this file.

`vfont.h`
Included for compatibility.

`vfork.h`
Included for compatibility.