Using Unix-specific Functions

The unix module

The unix module is used to access Unix specific functionality that is not found elsewhere, e.g. forking a process, accessing the system log etc. The unix module does not aim at being a complete set of all Unix functions and system calls, it merely contains those functions that where needed at some point of the arcapos development.

Process related functions

chdir(path)

Change the current working directory to path.

dup2(oldfd, newfd)

dup2() makes newfd be the copy of oldfd, closing newfd first if necessary.

fork()

Fork the current process. Returns the PID in the parent process, 0 in the child process, or, -1 in case of an error (no child process is created in this case).

kill(pid, signal)

Send the signal to the process with PID pid.

getcwd()

Returns the current working directory.

getpid()

Returns the process id of the process calling the function.

setpgid(pid, pgid)

Set the process group id of process pid to pgid.

getuid()

Returns the user id of the process calling the function.

getgid()

Returns the group id of the process calling the function.

File related functions

chown(path, uid, gid)

Change file ownership of the file at path to the (numerical) user id uid and (numerical) group id gid.

chmod(path, mode)

Change the file access mode of the file at path to mode.

rename(old, new)

Rename the file at old to new.

unlink(path)

Unlink (delete) the file at path.

Accessing User Information

setpwent()

Start accessing the user database.

endpwent()

Stop accessing the user database.

getpwent()

Get the next password entry. Returns a table with the following fields:

- pw_name
- pw_passpwd
- pw_uid
- pw_gid
- pw_gecos
- pw_dir
- pw_shell

getpwnam(username)

Return the password entry for user username.Returns a table with the following fields:

- pw_name
- pw_passpwd
- pw_uid
- pw_gid
- pw_gecos
- pw_dir

- pw_shell

getpwuid(uid)

Return the password entry for the user with the given user id uid. Returns a table with the followi - ng fields:

- pw_name
- pw_passpwd
- pw_uid
- pw_gid
- pw_gecos
- pw_dir
- pw_shell

getgrnam(name)

Return the group entry for the group name. Returns a table with the following fields:

- gr_name
- gr_passwd
- gr_gid
- gr_mem

The gr_mem field is itself a table containing all members of this group.

getgrgid(gid)

Get the group entry for the group with the numerical id gid. The result is the same table as for the getgrnam() function.

Getting and setting the system hostname

gethostname()

Return the hostname or nil if an error occurs.

sethostname(hostname)

Set the hostname, returns true on success, nil on error.

Using the system log

openlog(ident, option, facility)

Open the system log with the given ident, option, and, facility.

syslog(level, message)

Log message at the given level.

closelog()

Close the system log.

setlogmask(mask)

Sets the log mask to mask and returns the old value.

Select

fd_set()

Obtains a file descriptor set for later selecting on it. The set is initially zeroed.

fdset:clr(fd)

Clear fd in the file descriptor set.

fdset:isset(fd)

Set fd in the file descriptor set.

fdset:set(fd)

Check if fd is set in the filedescriptor set. Returns true if fd is set, false othwise.

fdset:zero()

Zero (clear) the file descriptor set.

select(nfds, readfds, writefds, errorfds [, timeout])

Perform select() on the specified file descriptor sets. Pass nil to omit one or more of the file descriptor sets. nfds is highest file descriptor number passed in the sets plus one. Timeout is either a single value representing milliseconds or two comma separated integers representing seconds and milliseconds. select() returns tthe number of file descriptors ready or -1 if an error occurs. If no timeout is specified, select() effectively becomes a poll and returns 0 if no file descriptors are ready.

Miscellaneous functions

arc4random()

The arc4random() function uses the key stream generator employed by the arc4 cipher, which uses 8*8 8 bit S-Boxes. The S-Boxes can be in about (2^1700) states. The arc4random() function returns pseudo-random numbers in the range of 0 to (2^32)–1.

errno()

Returns the last error code.

signal(sigcode, action)

Set the action for a signal.

sleep(seconds)

Sleep for the number of seconds passed.

getpass(prompt)

Display the prompt and get a password on the console.