NAME

mkDir, rmDir, getRootDir, chDir, cwd, gwd – using Shore directories

SYNOPSIS

```
VASResult       shore_vas::mkDir(
    const Path          path,
    mode_t              mode,
    lrid_t              *result
);

VASResult       shore_vas::rmDir(
    const Path          path
);

enum sm_lock_mode_t { NL = 0, IS, IX, SH, SIX, UD, EX };
typedef sm_lock_mode_t    LockMode;


VASResult shore_vas::chDir(
    const Path          path,
    LockMode            optional = NL
);

lrid_t   shore_vas::cwd()   const;

char     *shore_vas::gwd(
    char                *result,
    int                 resultlen,
    lrid_t              *dir = 0
);

VASResult shore_vas::getRootDir(
    lrid_t              *dir
);
```

DESCRIPTION

The methods **mkDir** and **rmDir** create and destroy Shore directory objects.

The method **chDir** changes the current working directory for SVAS instance. It acquires share locks on each directory in the given path. The caller may specify a lock to be acquired on the final directory in the path, if a lock stronger than a share lock is desired.

Cwd returns the OID of the current working directory. **Gwd** returns the path of the current working directory, or of the directory given. **Gwd**computes**the**path**by** successively changing directories to ".." from the initial working directory to the root, and by reading the directories along the way.

All these methods except **cwd** acquire locks and manipulate objects, so they must be run in transactions.

GetRootDir returns the oid of the root directory for the server.

ARGUMENTS

Path identifies the object of interest. It can be a full pathname, a relative pathname, or simply the file name. If it is a file name, the current working directory is assumed to be the directory in which the object is to reside (or already resides, in the case of removal).

Mode is the permissions bits for the object. When a directory is created, the given mode is modified by the client process's **umask.**

The *result* must point to an area in the caller's address space into which the SVAS writes the OID of the resulting directory object. It may not be a null pointer.

ChDir will obtain a lock on the final component of the path if a lock mode is given for the argument *optional.* If *optional* is not given, a share lock is acquired.

Gwd takes a pointer to a buffer in the caller's address space (*result*) and a length, *resultlen*, and writes the pathname of the current working directory into the buffer. A null character (ASCII NUL) terminates the pathname. The SVAS returns an error if the pathname and its terminating NUL do not fit in the buffer given. If the caller provides a non-null pointer to the object identifier of a directory in *dir*, **gwd** calculates a pathname of the given directory instead of the current working directory.

ENVIRONMENT

MkDir, rmDir and **chDir** are available to both client and server processes. **Cwd,** and **gwd** are on the server and in the client library.

ERRORS

Deadlocks can occur while locks are being acquired. See **transaction(svas)** for information about deadlocks.

A complete list of errors is in **errors(svas).**

VERSION

This manual page applies to Version 1.1 of the Shore software.

SPONSORSHIP

The Shore project is sponsored by the Advanced Research Project Agency, ARPA order number 018 (formerly 8230), monitored by the U.S. Army Research Laboratory under contract DAAB07-91-C-Q518.

SEE ALSO

getdirentries(svas), mount(svas), lookup(svas), errors(svas), transaction(svas), lockObj(svas), svas_server(svas), and **text(svas).**