

Published: 05/03/68

Identification

Working Directory Table Commands

change_wdir, restore_wdir, get_wdir, default_wdir
E. Q. Bjorkman

Purpose

The working directory table commands are parts of a single module, the working directory table module. For each process this module is responsible for:

1. Keeping the path name of the current working directory in the working directory table. That is, it keeps a record of the way the working directory, at any one time, was accessed.
2. Changing the working directory to the directory defined by a given path name. The directory defined by the given path name is made known to the process requesting the change and the contents of the working directory table are updated to make this change.
3. Remembering the path names defining previous working directories if the user wishes when a change is requested.

Usage

```
change_wdir path -`save`-
```

change_wdir establishes the directory specified by path as the current working directory for the process in which the call is made. If `save` is specified the path name of the working directory at the time the command was issued is saved in a push down stack, the working directory table. If `save` is not specified the old working directory is deleted when the new one becomes effective. The directory defined by path is made known to the process. If an error occurs while executing change_wdir, the contents of the working directory table are not changed.

```
restore_wdir
```

restore_wdir simply pops up the working directory table to the path name of the directory at the time change_wdir was last called with `save` specified. If an error occurs while executing restore_wdir, the contents of the working directory table are not changed.

get_wdir

get_wdir returns the pathname of the current working directory. It should be invoked at a console since the path name is written in the output stream. Note: The library procedure wdir (see BY.17.01) should be called by other procedures to obtain the pathname of the working directory. wdir returns the pathname as its value.

default_wdir path

When a user logs in, he may specify a working directory to use, or else he uses his default working directory. The path name of his default working directory is kept in his user profile (see BQ.4.03). The command default_wdir changes the path name of the user's default working directory to path. On subsequent logins, path is the default working directory. Note: The working directory specified by the user (whether implicitly in his login or by default) applies only to his working process. The working directories for Overseer and Device Manager processes (see BQ.3.00) are standard and may not readily be changed by the user.

Comments

Path must define a directory and the user must have at least one attribute on in the entry defined by path. If this condition is not satisfied, change_wdir and default_wdir make no changes. The user is notified by comments typed out on the console if a command is unsuccessful.

Data Base

Each process directory in the system contains a link to a data base which is a push down stack, the working directory table. The working directory table contains a character string which is the path name of the current working directory. Initially this segment is created when the process directory is constructed.

To implement the working directory table as a push down stack, a pointer associated with each new path name is provided. This pointer points back to the path name that was previously saved. If no path name was previously saved then this pointer is null. Another pointer, the working directory table entry (wdt_entryp), points to the structure, wdt_entry, which contains the last path name (i.e., path name of current working directory) and back pointer stored in the table. The following PL/I statement describes the structure of an entry of this table:

```
dc1 1 wdt_entry based (wdt_entryp),
      2 (bkptr, fwdptr) ptr,
      2 charct fixed bin (17),
      2 path char (wdt_entryp->wdt_entry.charct);
```

New generations of the working directory table entry are created by the "allocate" statement in the area, wdt, which is where the structure wdt_entry is located. Returns to previous path names are accomplished with the "free" statement.

Implementation

get_wdir

Get_wdir calls wdir, a library procedure (see BY.17.01) which returns the path name of the current working directory as its value. Get_wdir writes the path name of the current working directory in the output stream.

restore_wdir

In restore_wdir the pointer to the current generation of wdt_entry is set equal to the back pointer and the current generation of wdt_entry is freed. Before wdt_entry is freed, restore_wdir checks for a null back pointer, i.e., no 'save' was requested by any previous call to change_wdir. A null back pointer makes returning to a previous working directory not immediately possible. This error is relayed to the user as console comments.

change_wdir path -save-

Change_wdir calls the library procedure entryarg (BY.2.04) to determine whether path is a path name acceptable to the file system primitives, and if not, to make it so. (A path name must contain entry names separated by ">" (i.e., "<" is forbidden) to be acceptable to the file system primitives).

Change_wdir calls entry_status (BY.2.10) for information about path. The segment effectively pointed to must be a directory. If path is a link the link must be followed to determine whether it eventually defines a directory. If this condition is satisfied, change_wdir calls smm\$initiate (BD.3.05), to make the directory known to the process. If initiate

is successful or path is already known to the process at least one attribute for the user with respect to path is on. Failure to initiate path (if path is not already known to the process) is considered an error and change_wdir does not change the working directory table.

A new generation of the working directory table is allocated and the new path stored there. If save was specified, the former generation is pushed down and the new generation placed on the top of the stack. If save was not specified, the former generation is freed.

default_wdir

The file system routine entryarg (BY.2.04) is called and a path name acceptable to the file system is returned and stored in the string newpath. The contents of the entry defined by path are obtained by a call to entry_status (by.2.10). The type of the entry and access attributes of the user calling default_wdir are checked to be sure the user has at least one attribute on for the entry, and that the entry effectively defines a directory.

The record of the default working directory stored in the user's profile is updated to contain path. See BQ.4.03 for a description of the record of the default working directory in the user's user profile.