TO: MSPM Distribution
FROM: M. R. Thompson
DATE: May 27, 1969
SUBJECT: BY.2.02

BY.2.02 is being reissued to correct the explanation of newnamerr.

Published: 05/27/69
(Supersedes: BY.2.02, 04/28/69;
BY.2.02, 05/06/68;
BY.2.02, 10/03/67)

Identification

Decode basic file system error codes
check_fs_errcode
E. Q. Bjorkman

Purpose

Since errors detected by the basic file system primitives most likely are not static--that is, new errors may be added and codes may change--a procedure is needed to provide a stable interface to the codes returned. Check_fs_errcode interprets file system error codes using a data base that documents the codes. This data base is consistent with the hardcore ring data base that lists file system errors and their corresponding codes.

The file system interface procedures (BY.2.01) call check_fs_errcode to decipher error codes received from the file system. In general, only those procedures which call Basic File System primitives directly need use check_fs_errcode.

Usage

```
call check_fs_errcode(errcode, shortinfo, longinfo);

dcl errcode fixed bin (17),

    shortinfo char (8),

    longinfo char (N);(or char (N) var;)
```

where N is specified by the user and is sufficiently large to contain the information. The longinfo returned by check_fs_errcode does not exceed 100 characters.

check_fs_errcode scans the structure for fscodedinfo for the value errcode. If the value is not in the structure, shortinfo contains the string "xxxxxxxx" and longinfo equals "the error code (character representation of errcode) not found". If the errcode is located in fscodedinfo the contents of longinfo and shortinfo in the structure fscodedinfo corresponding to errcode are returned in longinfo and shortinfo respectively. The returned values are designed to be transmitted back to a caller by seterr (BY.11.01). shortinfo is the error code argument to seterr and longinfo is the error information argument. The 8 - character mnemonic (which may include blanks) is intended to be a very brief printable comment for the benefit of the

knowledgable user who doesn't want to see a wordy error comment. It is of a fixed maximum length to facilitate checking in a program.

Error Code Data Segment

The data segment fscodedinfo contains the relationship between file system error codes and user explanations. It is arranged so that each error may be externally referenced by its file system error name. For example, a user may see if the error code that was returned to him is the noentry error by comparing his code with fscodedinfo$noentry. (See the last part of this section for a list of file system errors and associated mnemonic codes and explanations). The EPLBSA coding for each error is given below and corresponds to the level 2 substructures of the EPL structure declaration in the implementation.

```
          segdef    noentry
noentry:  dec       1009
          dec       37
          aci       `noentry '
          aci       `File system could not find the entry.xxxxxx'
          aci       `xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
          aci       `xxxxxxxxxxxxxx'
```

Implementation

The epl structure which check_fs_errcode references is:

```
dcl 1    fscodedinfo based (cdptr),
           2 onesize fixed bin (17),
           2 twosize fixed bin (17),
           2 one (cdptr  fscodedinfo.onesize),
             3 fscode fixed bin (17),
             3 length fixed bin (17),
             3 shortinfo char (8),
             3 longinfo char (100),
```

The codes returned by the file system primitives can be classified as 1000 or 2000 codes. By subtracting 1000 from the 1000 codes and 2000 from the 2000 codes, indexes into the substructures one and two are produced. If the index is greater than the corresponding onesize or twosize the code is not contained in the structure.

Errors

The only error apprehended by check_fs_errcode is the inability to find the data segment fscodedinfo. This is considered to be a serious error, since fscodedinfo is a command system data base. Check_fs_errcode calls seterr to record the error (code="nodata", info="file system error data base cannot be found") and signals the condition check_fs_errcode_err.

APPENDIX:

FS CODE	FS NAME	MNEMONIC	CHARACTER INFORMATION
1001	moderr	moderr	Access is incorrect to the segment.
1002	dirseg	dirseg	This operation is not allowed for a directory.
1003	argerr	argerr	There is an inconsistency in arguments to the file system.
1004	newnamerr	newnme	User name to be added on writeacl not acceptable to file system.
1005	oldnamerr	oldname	Oldname to be removed from an entry is not on the entry.
1006	invalid_move	invlmove	Attempt to move a directory or a file already being moved by multi-level.
1007	noalloc	noalloc	There is no room to make requested allocations in the directory.
1008	bad_ring_ brackets	badbrack	Ring brackets input to directory control are not self-consistent or lower than validation level.

FS CODE	FS NAME	MNEMONIC	CHARACTER INFORMATION
1009	noentry	noentry	The file system could not find the entry
1010	toomanylinks	>links	There are too many links to get to a branch. (Note: maximum number allowed is 10.)**
1011	linkmoderr	linkmode	The execute access is needed to directory containing the link.
1012	clnzero	nonzero	There was an attempt to move segment to non-zero length entry.
1013	seg_in_use	in_use	Segment accessed is currently in use. Access is permitted anyway.
1014	fulldir	fulldir	There was an attempt to delete a non-empty directory.
*1015	full_ hashtbl	fullhash	The directory hash table is full (Note: probably won't happen). **
*1016	nohashtbl	nohash	There is no hash table on the accessed directory.
1017	user_not_ found	usernfd	The user name is not on the ACL for the branch.
*1018	retrieval_ trap_on	retrap	Retrieval trap on for a file special user is trying to access.
1019	noaccess	noaccess	The required access mode is absent in the directory of the entry. (Note: this is also returned when a directory cannot be found.)**
1020	notadir	notadir	A name specified as a directory is not a directory.
1021	nonamerr	noname	The operation would leave no names on entry.

FS CODE	FS NAME	MNEMONIC	CHARACTER INFORMATION
2001	boundviol	outbnd	There was an attempt to access beyond the end of the segment.
2002	invalidsegno	badsegno	There was an attempt to use an invalid segment.
2003	segknown	segnknow	The segment is already known on a call to initiate.
2004	namedup	namedup	There is a name duplication
*2005	nrmkst	nrmkst	There is no more room in the KST.
2006	name_not_ found	namenfd	The name is not found in the KST.
2007	infcnt_non_ zero	makunk	There was an attempt to make a directory unknown that has inferior segments.
2008	illegal_ deactivation	deactive	There was an illegal attempt to elete an AST entry.
2009	invalid_ ring_ crossing	ringerr	There was an attempt to inward wall cross to illegal segment or illegal gate.
2010	execute_ data	exdata	There was an attempt to execute in a data segment.

* These errors are internal to the file system and should never reach the user. They are documented for completeness and in case file system policy about any of them changes.

** Comments enclosed in (Note:....) are notes to the reader and do not appear as part of the error message.