MULTICS TECHNICAL BULLETIN

TO: Distribution
FROM: Joan Scott
DATE: 23 June 76
SUBJECT: Approved MCR's

Attached are the Multics Change Requests which were approved from June 1, 1976 through June 15, 1976.

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.
**TITLE:** Add -brief option to binder  

**AUTHOR:** M. Weaver

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PL/I X AIM X other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DATE:** 10 May 76

**STATUS:** 25/12 06/01/76

**DOCUMENTATION CHANGES**

- **Document**: Specify One or More

- **Specify One or More**: Document

**SUMMARY:**

Add -brief (-bf) option to the bind command to suppress warning messages as well as the "Binding..." message.

Print bad control argument message for any unrecognizable argument beginning with "-".

The names returned by expand_path will be used in messages printed on the terminal. The full primary names will be used in the source map.

**REASONS:**

To allow suppression of irritating messages. More standard treatment of command arguments.
Name: bind, bd

This command produces a single bound object segment from one or more unbound object segments that are called the components of the bound segment. (Compilers and the assembler produce unbound object segments.) A reference in one component to an external symbol defined in another component may be resolved during the binding. This prelinking avoids the cost of dynamic linking, and it also ensures that the reference is linked to the component regardless of the state of a process at the moment that dynamic linking would take place. References to a symbol are prelinked unless the contrary is specified by an instruction in the bindfile. The bindfile is a segment containing instructions that control various aspects of the binding operation (see "The Bindfile" below).

Usage:

bind archive_paths -update- -update_paths- -control_arg-

where:

1. archive_paths are the pathnames of archive segments containing one or more component object segments to be bound. Up to 16 input archive segments can be specified. They are logically concatenated in a left-to-right order to produce a single sequence of input component object segments. The specified pathname of the archive segment need not contain an explicit archive suffix.

2. -update, -ud is an optional functional argument to the binder indicating that the following list of archive segments (update_paths) specifies update rather than input object segments. If this optional argument is used, it must be preceded by a hyphen.

3. update_paths are pathnames of optional archive segments containing update object segments. Up to a combined total of 16 input and update segments can be specified. The contained update object segments are matched against the input object segments by object segment name. Matching update object segments replace the corresponding input object segments; unmatched ones are appended to the sequence of input object segments. If several update object segments have the same name, only the last one encountered is bound into the bound segment. The specified pathname of the archive segment need not contain an explicit archive suffix.

4. control_arg can be one of the following optional control arguments:

   -list, -ls produces a listing segment whose name is derived from the name of the bound object segment plus a suffix of list. The listing segment is generated for the purpose of printing; it contains the bound segment's bind control segment (see "The Bindfile" below), its bind map, and that information from the bound object segment that would be printed by the print_link_info command. (See the description of the dprint command in this document and the print_link_info command in the MPM Subsystem Writers' Guide.)
The binder produces as its output two segments: an executable bound procedure object segment and an optional, printable ASCII listing segment. The name of the bound object segment is, by default, derived from the entryname of the first input archive segment encountered by stripping the archive suffix from it. The name of the listing segment is derived from the name of the bound segment by adding the list suffix to it. Use of the Objectname master statement in the bindfile (see "Master Key Words" below) allows the name of the bound object segment to be stated explicitly. In addition, use of the Addname master statement in the binding instructions causes additional segment names to be added to the bound segment. The primary name of the bound object segment must not be the same as the name of any component.

The Bindfile

The bindfile is a segment containing symbolic instructions that control the operation of the binder. Its entryname must contain the suffix bind and it must be archived into any one of the input archive segments (at any location within that archive segment) where it is automatically located and recognized by the binder.

In case two bindfiles are specified, one in an input archive segment and the other in an update archive segment, the latter takes precedence and an appropriate message is printed to that effect.

The binder's symbolic instructions have their own syntax that allows for statements consisting of a key word followed by zero or more parameters and then delimited by a statement delimiter. Master statements pertain to the entire bound object segment; normal statements pertain to a single component object within the bound object segment. Master statements are identified by master key words that are distinct from normal key words in that they begin with a capital letter; normal key words begin with a lowercase letter. A key word designates a certain action to be undertaken by the binder pertaining to parameters following the keyword.
Function: produces a single bound object segment from one or more unbound object segments.

Syntax: `bd paths -control_arg-`

Arguments: `paths` are the pathnames of archive segments; the archive suffix need not be given.

Control arguments:
- `update paths`, `-ud paths` pathnames are update archives.
- `list`, `-ls` produces listing segment; it contains the bound segment's bind control segment, its bind map, and list of entries and links.
- `map` produces a listing segment (with the suffixes list and map) that contains only the bind map information.
- `brief`, `-bf` suppresses warning messages.
This segment documents changes to the binder in reverse chronological order.

05/19/76
-brief (-bf) has been added as a command control argument to suppress warning messages.

12/15/75
The binder no longer binds segments containing break maps; these segments were never bound correctly anyway.

8/15/75
The following changes have been installed:
-binder bugs 8, 9, and 10 have been fixed,
-the binder now refuses to bind nonstandard alm segments and no longer accepts the control argument -no_old_alm or the bindfile keyword No_Old_Alm,
-components with separate static sections can now be bound; the bound segment will have a separate static section only if all nonzero length static sections are separate.

2/23/73
The binder is now capable of deleting symbol tables. Two new keywords have been added to the bindfile language, one master and one local. Neither one requires parameters. They are:

No_Table causes the symbol tables from all the component symbol sections containing them to be omitted from the bound segment, except when they are needed for v2pl1 I/O. If not given, all tables will be kept.

table overrides the No_Table keyword and causes the symbol table of the component to be retained.

NOTE
Some standard version II pl1 procedures have a flag indicating that their symbol tables are needed for get data or put data statements. The binder issues a warning in these cases. However, there is currently a bug in the pl1 compiler that often causes the flag to be set when it isn't necessary, so the warning can usually be ignored.

1/5/73 (Version 8 binder installed)

If you have problems with the new version, use oldbind$bind, which is the previous version.

This version of the binder has the following new features:
- It creates standard object segments if all input components are standard object segments
- It enables the user to specify that no non-standard alm segments are to be bound (they will not work on the 6180);
### SUMMARY
Add sequential capability to rdisk_ I/O module.

### REASON
Required by USAF DSC contract. (We are paying the USAF $100 per day until they are satisfied with the "sequential" capability in the rdisk_ I/O module.)

### IMPLICATIONS
Users will be able to do reads and writes without having to supply a key.

### DETAILED PROPOSAL
The additional opening modes to be supported are sequential_input, sequential_output, and sequential_update.

The additional operations to be supported are write_record and position.

Attached to this MCR is a proposed MPM writeup, with the changed sectors flagged. The MPM writeup for the installed rdisk_ I/O module was used as the starting document in which the changes were made.

Also attached to this MCR are marked_up copies of some of the tables from the MPM Reference Guide relating the supported opening modes, file types, and I/O operations to each other.
Table 5-2. Opening Modes Supported by I/O Modules

<table>
<thead>
<tr>
<th>I/O Module</th>
<th>discard</th>
<th>rdisk_</th>
<th>record_stream_</th>
<th>tape_ansi_</th>
<th>tape_ibm_</th>
<th>tape_mult_</th>
<th>tty_</th>
<th>vfile_</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opening Mode</td>
<td>1 stream_input</td>
<td>X</td>
<td>X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 stream_output</td>
<td>X</td>
<td>X</td>
<td>X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 stream_input_output</td>
<td></td>
<td></td>
<td>X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 sequential_input</td>
<td></td>
<td></td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 sequential_output</td>
<td></td>
<td></td>
<td>X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 sequential_input_output</td>
<td></td>
<td></td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 sequential_update</td>
<td></td>
<td></td>
<td>X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 keyed_sequential_input</td>
<td></td>
<td></td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 keyed_sequential_output</td>
<td></td>
<td></td>
<td>X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 keyed_sequential_update</td>
<td></td>
<td></td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 direct_input</td>
<td></td>
<td></td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 direct_output</td>
<td></td>
<td></td>
<td>X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 direct_update</td>
<td></td>
<td></td>
<td>X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The syn_ I/O module is not included in this table because the allowed modes are a function of the switch to which the syn_ module is being attached.

X denotes the new capabilities to be provided with the addition of sequential I/O to the rdisk_ I/O module.
[NOTE TO REVIEWERS]

This is a draft to show the proposed additions for sequential capabilities. This is a modified version of the writeup of the rdisk_ I/O module that is in the MP
Supervisors manual, AG93B. The line numbers in the left margin are present to facilitate the making of comments about this draft. The meaning of the symbols which follow the line numbers are:

AAA The meaning of the text was changed in the 05/07/76 version which was sent to FSO for submission to the USAF.

BBB This change took place after the 05/07/76 version was sent to FSO.

--- Text was deleted from the MPM writeup for the 05/07/76 version which was sent to FSO.

END OF NCTL.)

Name: rdisk_

The rdisk_ I/O module supports I/O from/to removable disk packs. Sequential and indexed file types are supported.

Entries in this module are not called directly by users; rather, the module is accessed through the I/O system. See the "Multics Input/Output System" for a general description of the I/O system, and "File Input/Output" for a discussion of files, both in Section IV of the MPM Reference Guide.
Attach Description

The attach description has the following form:

\texttt{disk device_id pack_id -control_args-}

where:

1. \texttt{device_id} is a character string identifying the type number of the required disk device. The supported disk devices are listed in the table below, along with the character string to use for \texttt{device_id}:

<table>
<thead>
<tr>
<th>Character String</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>d181</td>
<td>DSU181</td>
</tr>
<tr>
<td>d190</td>
<td>DSU190</td>
</tr>
<tr>
<td>d191</td>
<td>DSU190 with the high-efficiency format (40 sectors/track)</td>
</tr>
<tr>
<td>m450</td>
<td>MSU0450</td>
</tr>
<tr>
<td>m450</td>
<td>MSU0451</td>
</tr>
</tbody>
</table>

2. \texttt{pack_id} is a character string identifying the disk pack to be mounted.

3. \texttt{control_args} may be chosen from the following, may occur only once.
-write indicates that the disk pack may be written on. If omitted, the operator is instructed to mount the pack with the PROTECT button pressed so that writing is inhibited.

-size n indicates that the value of n is to override the value of the nuff_len parameter as a record size limit for the read_record operation. (See "Notes" below.)

-priv indicates that the attachment is being made by a system process and that a disk drive reserved for system functions is to be assigned.

The attachment causes the specified disk pack to be mounted on a drive of the specified type.

Opening

The following opening modes are supported:

- sequential_input
- sequential_output
- sequential_update
- direct_input
- direct_update

Note that if the opening mode is of the output or update type, the attach description must include the -write
control argument so that the operator will not press the
PROTECT button when the pack is mounted.

Delete_Record_Operation

This operation is not supported.

Read_Length_Operation

This operation is not supported.

Position_Operation

This operation is supported for only the
sequential_input and sequential_update opening modes. The
type and quantity values are interpreted as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>position to the beginning of the file.</td>
</tr>
<tr>
<td>+1</td>
<td>position to the end of the file.</td>
</tr>
<tr>
<td>0</td>
<td>skip n sectors (forward if n &gt; 0; backward if n &lt; 0).</td>
</tr>
<tr>
<td>2</td>
<td>position to sector n.</td>
</tr>
</tbody>
</table>
Key_Record_Operation

If the amount of data to be read does not terminate on a sector boundary, the excess portion of the last sector is discarded. A code of 0 is returned in this case. (See "Notes" below.) This operation is not supported for the sequential_output opening mode.

Rewrite_Record_Operation

If the amount of data to be written does not terminate on a sector boundary, the remaining portion of the last sector is filled with spaces in sequential modes and binary zeros in direct modes. A code of 0 is returned in this case. (See "Notes" below.) This operation is supported for only the update opening modes.

Seek_Key_Operation

This operation returns a status code of 0 for any key that is a valid sector number. The record length returned is always 256 (current physical sector size in characters) for any valid key. The specified key must be a character string that could have been produced by editing through a PL/I picture of "(B)". (See "Notes" below.) This operation is supported for only the direct opening modes.

Control_Operation
rdisk

The following orders are supported when the I/O switch is open, except for getbounds, which is supported while the switch is attached.

```
change pack  causes the current pack to be dismounted and another pack to be mounted in its place. The info_ptr should point to a variable length character string (maximum of 32 characters) containing the identifier of the pack to be mounted.

get bounds  causes the lowest and highest sector numbers accessible by the caller under the current modes to be returned. The info_ptr should point to a structure of the following form:

```
struct
```
```
  act low bounds,
  2 low fixed bin(35),
  2 high fixed bin(35);
```

set size  causes the value of the record size override setting to be reset. The info_ptr should point to an allocated fixed binary(35) quantity containing the new override value.

Modes Operations

The modes operation is supported when the I/O switch is attached. The recognized modes are listed below. Each mode
has a complement indicated by the circumflex character (^) that turns the mode off.

label, ^label specifies that a system-defined number of sectors at the beginning of the pack are reserved for a pack label, and that a seek_key or position operation is to treat any key within this area as an invalid key. (The default is on.)

alttrak, ^alttrak specifies that the pack has been formatted with the assignment of alternate tracks, so that a system-defined number of sectors at the end of the pack are reserved for an alternate track area. Therefore, a seek_key or position operation is to treat any key within that area as an invalid key. (The default is off.)

wrtcmp, ^wrtcmp specifies that the write-and-compare instruction, rather than the write instruction, is used for the rewrite_record operation. This causes all data written to be read back and compared to the data as it was prior to being written. This mode should be used with discretion, since it doubles the data transfer time of every write. (The default is off.)
Write_Record_Operation

If the amount of data to be written does not terminate on a sector boundary, the remaining portion of the last sector is filled with spaces. A code of 0 is returned in this case. (See "Notes" below.) This operation is supported for only the sequential_output opening mode. A series of writes will write successive records.

Closing

The closing has no effect on the physical device. For the sequential_output opening mode, the effect is as if an End Of File flag is placed just beyond the end of the available disk area.

Detaching

The detachment causes the disk pack to be dismounted.

Notes

This I/O module is a very elementary, physical-device-oriented I/O facility, providing the basic user-level interface to a disk device. All operations are performed through calls to various I/O interface (I/OI) mechanisms and resource control package (RCP) entries. Certain conditions must be satisfied before a user process
can make use of this facility:

1. The system must be configured with one or more disk drives available as I/O disks.

2. The user must have access to assign the disk drive with RCP, access to the I/O gates, and access to the "acs" segment (e.g., »sc1.rcp »iskb_18.acs) that is used by the site to control access to the disk drive.

For input and update opening modes, the file occupies the entire available disk area (see the getbounds control order). For the sequential_output opening mode, the file is considered to be empty. That is, an open followed by a write will record data in the first sector of the available disk area.

For direct opening modes, the entire disk pack is treated as an indexed file, with keys interpreted literally as physical sector numbers. Hence, the only allowable keys are those that can be converted into fixed binary integers that fall within the range of valid sector numbers for the given disk device under the current modes, as returned by the getbounds control operation.

For the sequential_input and sequential_update opening modes, if an attempt is made to read beyond the end of the user-accessible area, the code error_table$end_of_info is returned. For all other opening modes, if an attempt is made to read or write beyond the end of the user-accessible area or disk, the code error_table$device_end is returned. If a defective track is encountered or if any other unrecoverable data transmission error is encountered, the
code error_table:device-parity is returned.

The record length is specified through the buff_len parameter in the read_record operation, and through the rec_len parameter for the write and rewrite operations, unless overridden by a size control argument in the attach descriptor, or by a setsize control order.

The following items must be considered when using this I/O module with language input/output:

1. Device Attachment and File Opening:
   a. PL/I: A file can be attached to a disk pack in PL/I by specifying the appropriate attach description in the title option of an oP statement. After opening, the desired modes should be set and the current sector bounds should be obtained through direct calls to lox_find_locb, lox_modes, and lox_control. These lox subroutine entry points are described in Section II.
   b. FORTRAN: It is not possible to attach a file to a disk pack within FORTRAN. Here, the attachment must be made external to the FORTRAN program, e.g., through the lc_call command (described in the MPM Commands) or through use of a PL/I subroutine. FORTRAN automatically opens the file with the appropriate attributes. Also, it is impossible to set modes or obtain sector counts from within FORTRAN. This should be
done through use of a PL/I subroutine prior to the first FORTRAN reference to the file.

2. Input

a. PL/I: The input record length (buff_len) is determined by the size of the variable specified in the into option.

For the sequential_input and sequential_update opening modes, use the PL/I read statement with the into option to read data. Use the ignore option to skip forward within the file. An open statement followed by a read statement will read in the first record. Successive reads will obtain successive records.

For the direct_inout opening mode, use the PL/I read statement with the into and key options. The set option should not be used. The key should be a character string containing the character representation of the desired sector number.

The PL/I get statement can be used with the sequential_input opening mode if the record_stream_ I/O module is referenced in the attach description of the open statement.

b. FORTRAN: In FORTRAN, buff_len has no relationship to input variable size. Hence, the size control argument must be specified in the attach description if the disk pack is to be read through FORTRAN. The size should
be set to the length of the longest expected record.

For the sequential_input opening mode, use the unformatted sequential read statement.

For the direct_input opening mode, use the unformatted keyed version of the FORTRAN read statement. The key must be an integer, whose value is the desired sector number.

3. Output

6. PL/I: The size of the variable referenced in the from option determines the length of the record written to disk.

For the sequential_output opening mode, use the write statement with the from option. An open statement followed by a write statement will start writing at the beginning of the available area on the disk pack.

For the sequential_update opening mode, use the rewrite statement with the from option. A previous read statement must have been used to designate which record will be updated.

For the direct_update opening mode, use the rewrite statement with the from and key options. The key should be a character string containing the character representation of the desired sector number.

The PL/I put statement can be used with the sequential_output opening mode if the
The record_stream I/O module is referenced in the attach description of the open statement.

FORTRAN: The size of the output record is determined by the amount of data specified in the write list.

For the sequential_output opening mode, use the unformatted sequential write version of the FORTRAN write statement.

For the direct_update opening mode, use the unformatted keyed version of the write statement. The key should be a character string containing the character representation of the desired sector number.
MULTICS CHANGE REQUEST

TITLE: Fix erroneous calls to dir_control_error in append, reclassify and quota

AUTHOR: L. Scheffler

Planned for System: MR4.0
Fixes Bug Number(s): unreported
Documented in MIB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no

CODED IN: (X)PL/I ( )ALM ( )other-see below

Performance: ( )better (X)same ( )worse

DOCUMENTATION CHANGES (specify one or more)

MPW (vol,sect) MPAM (sect)
MDSN (sect) MSAM (sect)
PLMs (AN#)
Info Segs
Other

None (reason) None needed

OBJECTIONS/COMMENTS:

Summary: Change instances in append, reclassify, and quota, where calls to dir_control_error$attributes should pass a pointer to the directory on which access is insufficient, instead of (as currently) passing a pointer to that directory's branch in its parent.

Reasons: These calls result in dir_control_error computing error_table_codes based on the user's access to the wrong directory (the parent of the directory of interest), and to auditing messages reporting access denials on wrong directories.

Implications: error_table_codes will be computed properly and auditing messages will report the proper directories in these cases.

Error messages for the create and create_dir commands and error_table_codes from append will change in some (rare) cases where they are incorrect today. (See Detailed Explanation below.)

Error messages for the move_quota command and error_table_codes produced by hcs_quota_move will not change because the logic of access checking in quota precludes cases where error codes would be different. However, the right codes will be computed for the right reasons instead of, as currently, for the wrong reasons.

Detailed Explanation: dir_control_error$attributes accepts a pointer argument to determine the directory containing attributes (of real or soon-to-be-created branches) or quota cells to which access is insufficient. This pointer may be a directory pointer (offset = 0), or an entry pointer, in which case a directory pointer is made from the entry pointer and an offset of 0. dir_control_error computes the
error_table_ code to be returned based on the user's access to the
directory pointed to. In append, reclassify, and quota, some of the
errors being reported are instances of incorrect access to attributes
or quota cells contained in the directory whose entry pointer (in
its parent) is currently being passed. This causes dir_control_error
to check access on the wrong directory before deciding what error
code to return.

In the case of append, this results in erroneous error codes (and
therefore wrong error messages from the create and create_dir
commands) in the case of "cr A>B>C>D" where the user's access on B
and C is null. Currently, the user's access on A determines the
error codes: null access on A results in error_table_$no_info while
non-null access on A results in error_table_$incorrect_access. This
is one directory level off.
SUMMARY:

Install `lv_attached` active function, which returns "true" if the named logical volume is attached to the user's process.

REASONS:

User exec_com files may wish to check whether a volume is attached before proceeding.
Draft MPM documentation

**Name:** lv_attached

This active function returns "true" if the named logical volume is attached to the user's process or is a public volume.

**Usage:**

```c
[lv_attached volname]
```

where volname is the name of a logical volume.
**SUMMARY:**

Install new command `set_flagbox` and new command/active function `get_flagbox`.

**REASONS:**

These commands allow a privileged process to read and set the flagbox flags. This allows modification of `system_start_up.ec` to control the rebooting of the system.

**IMPLICATIONS:**

New `system_start_up.ec` and BOS runcoms must be created in order to put automatic rebooting into effect. This change installs the last program needed for this facility.
**Name:** get_flagbox

This active function returns either "true" or "false" depending on the value of a specified flag in the BOS/Multics communication area.

**Usage:**

[get_flagbox keyword]

where keyword may be either a number from 1 to 36, or the name of one of the flagbox flags. This active function can also be used as a command. Privileged access to phcs_ is required to use this program.

**Name:** set_flagbox

This command sets the value of a specified flag in the BOS/Multics communication area.

**Usage:**

set_flagbox keyword value

where keyword may be either a number from 1 to 36 or the name of one of the flagbox flags. Privileged access to phcs_ is required to use this program.

**Notes:** The names of the flagbox flags are:

1. auto_reboot
2. booting
3. crashed
4. rebooted
5. bit5
6. bit6
7. bit7
8. bit8
9. bit9
10. bit10
11. bit11
12. bit12
13. bit13
14. bit14
15. bit15
16. bit16
17. bit17
18. bit18
19. bit19
20. bit20
21. bit21
Title: Install display_kst_entry

Authors: Richard Brat

Planned for System: 5.0
Fixes Bug Number(s): not applicable
Documented in MTA: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: (B) PL/I  ( ) ALM  ( ) other-see below
Performance: ( ) better ( ) same ( ) worse

DOCUMENTATION CHANGES (specify one or more): MPM (vol,sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLIs (AN#) 51
Info Segs
Other

OBJECTIONS/COMMENTS:

Summary: Install a private tool of mine which prints the contents of a KST entry given either a segment number or a relative pathname.
**Name**: display_kst_entry

The `display_kst_entry` command prints the contents of a KST entry. The KST entry to be dumped may be indicated by either a segment number or a relative pathname of the associated object. If the relative pathname looks like a segment number then it must be preceded by the `-name` (`-nm`) control argument.

**Usage**: `display_kst_entry [-nm | -name] target`

where: target is either a segment number or relative pathname.

**Note**: This command uses phcs only when needed.

display_kst_entry start_up.ec

`secco: 256 at 155:470`
`usage: 0, 0, 0, 0, 2, 0, 0, 0`
`entrvn: 243:5452`
`uid: 033100743603`
`dtbm: 416334652254`
`node: 7 (4, 4, 4)`
`ex node: 0 (0, 0, 0)`
`infcount: 0`
`hdr: 4`
`flags: write`
**TITLE:** Put useful information in tty DIM event messages  
**AUTHOR:** Robert S. Coren

<table>
<thead>
<tr>
<th>Planned for System:</th>
<th>MR 5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixes Bug Number(s):</td>
<td>not applicable</td>
</tr>
<tr>
<td>Documented in MTB:</td>
<td>not applicable</td>
</tr>
<tr>
<td>Incompatible Change:</td>
<td>no</td>
</tr>
<tr>
<td>User/Operations-visible Interface Change:</td>
<td>no</td>
</tr>
<tr>
<td>Coded in:</td>
<td>(E)PL/I ( )ALM ( )other-see below</td>
</tr>
<tr>
<td>Performance:</td>
<td>( )better ( )same ( )worse</td>
</tr>
</tbody>
</table>

**DOCUMENTATION CHANGES (specify one or more)**
- MPM (vol,sect)
- MOSN (sect)
- PLMs (AN#)
- Info Segs
- AN85
- Other

**OBJECTIONS/COMMENTS:**

**SUMMARY:** Make the ring-zero typewriter DIM put the device index (devx) and a type code in the event message for all wakeups.

**REASONS:** At present the event message contains zero except for hangups, when it contains the devx; no system programs examine the event message. It would be useful, especially for a process that has attached extra terminals, to be able to tell from the event message what the meaning of a tty DIM wakeup is.

**IMPLICATIONS:** None.

**DETAILED PROPOSAL:** Use the event message format described by the attached include file. This mechanism won't work for fast ipc channels (normally used for tty I/O).
BEGIN INCLUDE FILE ... tty_event_message.incl.pl1 */

/*
describes event message passed with wakeups from the tty DIM */

Created 5/24/76 by Robert S. Coren */

dcl tty_event_message fixed bin (71);

dcl tty_msg based (addr (tty_event_message)),
    ev_devx fixed bin (17) unaligned,              /* device index */
    ev_type fixed bin (17) unaligned,              /* reason for wakeup (see below) */
    pad bit (36);

dcl UNSPECIFIED_MSG fixed bin internal static options (constant) init (0); /* used for "start" order, etc

dcl DIALUP_MSG fixed bin internal static options (constant) init (1); /* dialup */

dcl HANGUP_MSG fixed bin internal static options (constant) init (2); /* hangup */

dcl DIALOUT_MSG fixed bin internal static options (constant) init (3); /* dialout status returned */

dcl QUIT_MSG fixed bin internal static options (constant) init (4); /* quit */

dcl READ_MSG fixed bin internal static options (constant) init (5); /* input arrived */

dcl WRITE_MSG fixed bin internal static options (constant) init (6); /* output completed */

END INCLUDE FILE ... tty_event_message.incl.pl1 */
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE:</td>
<td>Change mailbox initial ACL</td>
</tr>
<tr>
<td>AUTHOR:</td>
<td>S. Herbst</td>
</tr>
<tr>
<td>STATUS</td>
<td>MCR 1907</td>
</tr>
<tr>
<td>DATE</td>
<td>5/20/76</td>
</tr>
<tr>
<td>Objections/Comments:</td>
<td></td>
</tr>
</tbody>
</table>

**SUMMARY:**

Change the ACL placed on a newly created mailbox from:

```
    adros creator
    as    *.SysDaemon.*
    aow   *.* *
```


  to:

```
    adrosw creator
    asw   *.SysDaemon.*
    aow   *.* *
```

**REASON:**

So that the creator of the mailbox can send himself wakeup messages, and the I/O Daemon can send him wakeup notifications.
Sending a Segment

The contents of the segment specified by path is sent to the mailbox:

`>user_dir_dir>Project_idi>Person_idi>Person_idi.mbx`

for each Person_id-Project_id pair specified in the command line.

The segment to be mailed must be less than one record long (4096 ASCII characters).

Composing Mail

If path is *, mail responds with "Input:" and accepts lines from the terminal until a line consisting only of a period (.) is typed; the typed lines are then sent to the specified user(s).

Creating a Mailbox

A default mailbox is created automatically the first time a user prints his own mail. The default mailbox is:

`>user_dir_dir>Project_id>Person_id>Person_id.mbx`

Access on a newly created mailbox is automatically set to adro for the user who created it, ad for *.SysDaemon.*, and adro for *.*. The types of extended access for mailboxes are:

- add a message
- delete any message
- read any message
- read or delete only your own messages, that is, those sent by you
- find out how many messages are in the mailbox

The modes "n", "null", or '"" specify null access.
MULTICS CHANGE REQUEST

TITLE: Change charging for dprint requests

AUTHOR: Jim Homan

Planned for System: not applicable
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: yes
Coded in: (M) PL/I () ALM () other-see below
Performance: ( ) better (M) same ( ) worse

DOCUMENTATION CHANGES (specify one or more)
MPM (vol, sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLIs (AN#) SamSysDaemon/Admin
Info Segs pending_changes
Other

OBJECTIONS/COMMENTS:
Whitmore or VanVleck should audit before installation.

Headings are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY
Charge for dprint requests on the basis of lines printed instead of blocks printed.

REASONS
The current method of charging for dprint usage is to charge for "blocks" printed. This is confusing to the user and is not as accurate a measure of printing costs as lines printed would be. Most other computer systems charge on the basis of lines printed. Originally, charges were based on blocks because the line count was not available. The block size (700 bits) was intended to be equal to the average line size. Now that the line count is available, dprint charges should be based on the actual line count.

IMPLICATIONS
The primary implication to users will be a change in dprint charges. The change will vary depending on the nature of the data being printed, but it is estimated that most users will see an increase if rates remain the same, although users who dprint longer than average lines will see a decrease.

Charges for dprinting a segment may differ for different printer page widths, if lines wrap around at one width but not at another.
User programs which calculate dprint charges based on the bit count of a segment will no longer give correct costs.
System programs and info files (e.g. billing programs, ed_installation_parms, resource_usage, rates.info) have always referred to lines rather than blocks and so will require no changes.
DETAILED PROPOSAL

Change output_request_ to fill in ordata.line_count with the line count supplied by the DIM, if non-zero. If zero (as is the case for card DIMs), use block count as before. Change io_daemon_account_ to use ordata.line_count instead of calculating the block count. Change tail_sheet_ to print out "xxx lines at $y.yy per 1000 lines."
### MULTICS CHANGE REQUEST

**TITLE:** Add alarm order to operator console DIM

**AUTHOR:** Larry Johnson

**Planned for System:** Mk 4.1

**Fixes bug Number(s):** not applicable

**Documented in MTB:** not applicable

**Incompatible Change:** no

**User/Operations-visible Interface Change:** yes

**Coded in:** ( )PL/I ( )ALM ( )other-see below

**Performance:** ( )better ( )same ( )worse

---

**DOCUMENTATION CHANGES (specify one or more)**

- MPAM (sect)
- MSAM (sect)
- PLms (AN#)
- Info Segs
- Other

None (reason)

---

**OBJECTIONS/COMMENTS:**

---

**Summary**

Implement an "alarm" order in oc_ which will cause the next message written by the DIM to turn on the console beeper.

**Reasons**

This feature is needed by the message coordinator, which currently simulates it in an awkward manner.

**Detailed Proposal**

Currently, if the message coordinator needs to turn on the console beeper, it calls phcs_tring_0_message to print:

```
Initializer.Sysdaemon.z: *****
```

which turns on the beeper when it is printed. There are two problems with the mechanism:

1. The operator's console and the syserr log are cluttered with uninteresting messages.
2. The alarm is often turned on too soon, as syserr messages have priority over messages from the DIM.

The message coordinator will be changed to use the "alarm" order instead of the phcs_ call.
MULTICS CHANGE REQUEST

**TITLE:** Fix dump_cdt

**AUTHOR:** Mike Grady

**Planned for System:** MINI.0  
**Fixes Run Number(s):** not applicable  
**Documented in MTR:** not applicable  
**Incompatible Changes:** yes  
**User/Operations-visible Interface Change:** yes  
**Coded in:** (X)PL/I  
**Performance:** ( )better (X)same ( )worse

**DOCUMENTATION CHANGES (specify one or more):**

<table>
<thead>
<tr>
<th>VPX (vol,sect)</th>
<th>MPAM (sect)</th>
<th>MOSN (sect)</th>
<th>MSAM (sect)</th>
<th>Appendix A</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLUs (AH#)</td>
<td>Info Seqs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DEJECTIONS/COMMENTS:**

Headlines are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

**SUMMARY:**

Have dump_cdt use proper form of control arguments and require control arguments be used to specify dumping of a particular channel or a specified CDT.

**REASONS:**

Commands should be consistent in support of various control arguments and should not have confusing usages.

**IMPLICATIONS:**

Users of dump_cdt will find certain incompatibilities.

**DETAILED PROPOSAL:**

Have dump_cdt accept the following control arguments:

- **-all, -a**  
  print all CDT entries
- **-no_header, -nha**  
  don't print CDT header info
- **-fnp**  
  print the entries for FNPs
- **-channel X, -chn X**  
  print specified channel entry
- **-cmf X**  
  dump CDT in the form of CMF into segment X
- **-pathname X, -pn X**  
  dump info from specified CDT X
dump_cdt

**Name:** dump_cdt

The dump_cdt command enables a system administrator to dump the channel definition table (CDT) residing in `>system_control_1`. Optional arguments control whether the header information is printed, and whether all CDT entries, or only a selected CDT entry, is printed.

**Usage**

```
dump_cdt [cdt_path] name [control_args]
```

- **cdt_path** is the relative pathname of the CDT to be dumped. If not specified, `>system_control_1.cdt` is dumped.
- **name** is the name of a channel whose CDT entry is to be dumped. It must be of the form "ttyXXX", where XXX is the numeric channel designation. If not given, all CDT entries are printed.
- **control_args** may be selected from the following list:
  - `-all` dump all CDT entries (this is the default).
  - `-nhe`, `-no_header` do not print the CDT header variables.
  - `-channel name` takes name as a literal channel name. This allows any prefix, instead of just "tty", etc.

**Example**

To dump the entire CDT, give the command:

```
dump_cdt -all
```

To dump only the CDT entry for "tty000", give the command:

```
dump_cdt tty000 -channel tty000 -nhe
```

- `-fnp` dump entries for the configured FNP's.
- `-cmf X` dump the CDT in the form of a CMF. The output of this command will be accepted by CV-Cmf. Use of this control arg is incompatible with `-nhe`, `-fnp` or `-chn`.
- `-pathname X` dump the CDT whose pathname is X.
- `-pn X`
**TITLE:** Install display_branch tool

**AUTHOR:** Bernard Greenberg

<table>
<thead>
<tr>
<th>Planned for System:</th>
<th>MR 4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixes Bug Number(s):</td>
<td>not applicable</td>
</tr>
<tr>
<td>Documented in MTB:</td>
<td>not applicable</td>
</tr>
<tr>
<td>Incompatible Change:</td>
<td>no</td>
</tr>
<tr>
<td>User/Operations-visible Interface Change:</td>
<td>no</td>
</tr>
<tr>
<td>Coded in:</td>
<td>( )PL/I ( )ALM ( )other-see below</td>
</tr>
<tr>
<td>Performance:</td>
<td>( )better ( )same ( )worse</td>
</tr>
</tbody>
</table>

**DOCUMENTATION CHANGES (specify one or more):**
- MPM (vol,sect)
- MOSN (sect)
- PLMs (AN#) 51
- Info Segs
- Other

**OBSTRUCTIONS/COMMENTS:**
- Add example of output to documentation

**HEADINGS:** SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

**SUMMARY:** Install print_branch, a tool to dump directory branches.

**REASON:** In the analysis of system problems, particularly those involving branch/vtoc connection failures, it is often necessary to find out information in the branch of a segment either not returned at all by the status command, or not returned without accessing the vtoc of the segment. Such information includes the UID of the segment, the Physical Volume name/id, and the VTOC index. Also, the segment number of the father of the segment is often difficult to determine.

**IMPLICATIONS:** Improved debuggability of the operating system.

**DETAILED PROPOSAL:** See attached documentation.
Name: display_branch

This command is used to print out information about directory entries not returned by the status command. It also lists the segment UID and the location of the branch. No attempt is made to access the VTOCE of the segment for any information.

Usage: display_branch path
OR display_branch segno
OR display_branch segoffset
OR display_branch -name path

where

path is the pathname of the segment whose branch is to be displayed.

segno is the octal segment number of the segment whose branch is to be displayed.

segoffset is octal pointer representation of address of the branch to be displayed (for example, 26011664).

Notes:

If the pathname of a branch is a valid octal number or can be construed as a valid octal pointer, then the "-name" ("-nm") argument must precede the segment name.

The user of this command must have access to the gate phcs_, which is necessary in order to initiate directories.
display_branch >system_library_standard>bound_fscom1.

Branch for bound_fscom1_ in >system_library_standard at 231137162

JID 035427542510, is vtocx 5777 on root3 (of log vol. root)
Switches: safety_sw
Hinge brackets (Y-5 5)
Entry modified 06/22/76  1356.7 mst Tue
Never dumped.
36 names.
r 1058 1.154 3.066 77
SUMMARY: Install the tool vtoc_pathname, which prints out the pathname of a segment given its volume and vtoc index.

REASONS: There is currently no way to determine the pathname of a segment given its vtoc. Not only is this important for system debugging, but the Physical Volume Salvager reports errors in terms of volume name and VT0C index, and cannot determine pathnames for the foreseeable future. Pathnames can be determined from vtoces only when the system is up and the entire root logical volume is mounted and salvaged.

DETAILED PROPOSAL: See attached documentation.
Name: vtoc_pathname,

This command is used to determine the pathname of a segment from the location of its VTOC entry (vtoce). The location of the vtoce is specified by giving its volume name (or Physical Volume Table index, if known), and index into the VTOC of that volume.

Usage: vtoc_pathname volname vtoce -control-args-
OR vtoc_pathname pvtx vtocx -control-args-

Where:

- **volname** is the physical volume name of the volume on which the vtoce resides. This volume must be mounted, and part of a mounted logical volume.

- **pvtx** is the physical volume table index of the volume on which the vtoce resides, if known. It must be given in octal.

- **vtoce** is the vtoce index of the vtoce. It must be given in octal.

- **-control-args-** can be "-brief", ("-bf"), which suppresses the printing of an error message if the vtoce is free. This facilitates the production of maps of an entire volume.

Notes:

This command requires access to the gate phcs_, as it must copy out directories.

The user's process must have status access to each of the containing directories of the segment in question. The command will supply "NO-ACCESS-" as the entry name at the level at which further access is necessary, if needed. If one of the containing directories specified in the vtoce does not exist in its containing directory, the command will supply "-NOT-LISTED-" as the entry name at that level. The command supplies "????" as the entry name at any level below that at which either of the above problems exist.
**SUMMARY:** The force_pd_abandon command incorrectly reports failure when invoked. This is a problem in interface to delete_pd_records.

Change this command to correctly interpret delete_pd_records' return parameter.

**REASONS:** Fix the bug.
MULTICS CHANGE REQUEST

| TITLE: Introduce RPV-only directories. |
| AUTHOR: Bernard Greenberg |

| Planned for System: MR 4.0 |
| Fixes Bug Number(s): not applicable |
| Documented in MTB: not applicable |
| Incompatible Change: no |
| User/Operations-visible Interface Change: no |
| Coded in: (I)PL/I ( )ALM ( )other-see below |
| Performance: ( )better (I)same ( )worse |

**DOCUMENTATION CHANGES (specify one or more):**

- MPM (vol, sect) MPAM (sect)
- MOSN (sect) MSAM (sect)
- PLMs (AN#) 61
- Info Segs
- Other

**OBJECTIONS/COMMENTS:**

---

**SUMMARY:** Add a bit dir.force_rpv to the directory header, valid only for directories whose sons-LVID is that of the root Logical Volume, which forces all segments of this directory to be created on the Root Physical Volume (RPV) and not migrate off of it.

**REASONS:** In order for the Ring 1 volume administration and registration software to operate properly, the volume registration segments must be accessible at all times, including before the entire root has been accepted during system startup. Lack of these segments causes segment faults when registration lists are requested, or salvages of non-root volumes are attempted.

**IMPLICATIONS:** Correct operation of the ring 1 disk administration software.

**DETAILED PROPOSAL:** Implement a gate, `hphcs_$set_rpv`, which given the pathname of a directory whose sons-LVID is that of the root logical volume, forces the bit dir.force_rpv on. This entry will be called by system `startup` (program "hdx") to force the directory ">lv" (volume registration segments) to live on the RPV. As this bit is set each time the system is brought up, this bit need not be dumped or reloaded.
### SUMMARY

In certain crashes involving VTOC write errors, or disk problems in general, vtoc buffers may be left in an inconsistent state, causing lost notifies during emergency shutdown. Add an entry to the vtoc_man to requeue all operations at ESD time. This will involve running the disk dim until the queues are empty.

### REASON

Make ESD more certain.

### IMPLICATIONS

Greater reliability.
SUMMARY: Change the ring zero truncate primitive to no longer set the directory modify switch. In the new storage system, truncation does not involve modification of a directory.

REASONS: Performance. Avoid storing processid into directory modifier-id word, avoiding subsequent paging I/O.
Ver. 3
741022 MULTICS CHANGE REQUEST

TITLE: Implement gates for segment migration.

AUTHOR: Bernard Greenberg

Planned for System: MR 4.0
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: ( )PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol,sect)         MPAM (sect)
MOSN (sect)            MSAM (sect)
PLMs (AN#)             61
Info Segs
Other

OBJECTIONS/COMMENTS:

Headings are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY: Provide a ring-zero file system interface to the segment mover, and gates into it.

REASONS: A facility for compressing a logical volume on line is desired. A facility for forcing segments off a given physical volume is also desired. Although the segment mover moves segments automatically in response to segment faults induced by page control in out-of-physical-volume situations, an administrative interface to this facility is needed. Although full support of a segment migration facility is not planned for release 4.0, the existence of these gates will allow experimentation along these avenues for future releases.

DETAILED PROPOSAL: Implement hphcs$_$vacate_pv (pvid, pvtx, code) to start vacation of a physical volume

hphcs$_$stop_vacate (pvid, pvtx, code) to stop vacation.

hphcs$_$move_seg_file (dir, ent, code) to move a segment off its physical volume of residence to another in the same volume.

hphcs$_$move_seg_seg (segptr, code) (similar).
| **TITLE:** | Fix fs_get mode bug. |
| **AUTHOR:** | Bernard Greenberg |

- **Planned for System:** MR 4.0
- **Fixes Bug Number(s):** not applicable
- **Documented in MTB:** not applicable
- **Incompatible Change:** no
- **User/Operations-visible Interface Change:** no
- **Coded in:** ( )PL/I ( )ALM ( )other-see below
- **Performance:** ( )better ( )same ( )worse

**DOCUMENTATION CHANGES (specify one or more):**
- **MPM (vol,sect):** MPAM (sect)
- **MOSN (sect):** MSAM (sect)
- **PLMs (AN#):**
- **Info Segs:** none
- **Other:** none

**OBJECTIONS/COMMENTS:**

---

**SUMMARY:** An uninitialized variable in the hardcore program fs_get causes process terminations upon certain calls.

Properly initialize this variable (ep) in all cases.

**REASONS:** Fewer system crashes when the Initializer is the unlucky process.
**SUMMARY:** A recent change to the module "activate" to unlock the AST during VTOC reads exposed a window during which the demounting of a physical volume can cause a segment to be incorrectly activated, and potentially deactivated onto a wrong volume.

**REASONS:** A PVT index is used without protection after the AST was unlocked. It must be revalidated after the AST is re-locked.

**IMPLICATIONS:** More reliable system operation.
SUMMARY: Setting a maximum length of zero on a segment does not work. Code was recently removed which signalled seg_fault error in the case of connecting to a segment at an address above its maximum length, letting the resulting boundsfault signal the correct condition (out_of_bounds). This does not work in the case of a zero maximum length, as no valid SDW is capable of being constructed to indicate this condition.

PROPOSAL: Put an explicit check in the segment fault handler to check for any out_of_bounds reference, and signal out_of_bounds directly from the segment fault handler if this be the case.

IMPLICATIONS: This is quite messy, and subverts the entire hardcore fault signalling mechanism, but is necessary to preserve the documented user interface.
TITLE: Remove AIM function from phcs_$initiate

AUTHOR: Bernard Greenberg

Planned for System: MR 4.0
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: (W)PL/I ( )ALM ( )other-see below
Performance: ( )better (W)same ( )worse

DOCUMENTATION CHANGES (specify one or more)
MPM (vol,sect)  MPAM (sect)
MOSN (sect)     MSAM (sect)
PLMs (AN#)    75
Info Segs
Other

HEADINGS ARE: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY: Restore phcs_$initiate to its original function, i.e., initiate at a validation level of 0. Remove AIM features which counteract AIM checks and invoke directory privilege.

REASONS: Under AIM, the phcs_$initiate gate tries to invoke directory privilege for the caller. This allows access outside of AIM restrictions. However, the original function of the phcs_$initiate interface is simply to allow segment numbers to be developed for lower-ring segments. Since access to phcs_ implies the ability to obtain passwords by dumping teletype buffers, etc., it seems that no process at a lower access class than system_high should be allowed access to this gate. Thus, the invocation of directory privilege here gains nothing. phcs_$initiate has not been used in the past, and is not intended to be used to initiate segments outside of AIM restrictions (system_privilege_$initiate exists for this purpose.) On the other hand, it causes a large number of auditing messages, and segfaults, as it clears SDW's to force access recomputation. This destroys the usefulness of this gate for several hardcore debugging tools (e.g., print_aste_ptp).

IMPLICATIONS: Processes at lower authorization than system_high, who have access to phcs_, cannot use it to obtain segment numbers for directories at higher authorization levels.

DETAILED PROPOSAL: Redirect call form level_0 to initiate$priv_init to standard entry.
TITLE: Recode BOS SSTN to ignore AST hierarchy.

AUTHOR: Bernard Greenberg

Planned for System: MR 4.0

TITLE: Recode BOS SSTN to ignore AST hierarchy.

AUTHOR: Bernard Greenberg

Planned for System: MR 4.0

Fixes Bug Number(s): not applicable

Documented in MTB: not applicable

Incompatible Change: no

User/Operations-visible Interface Change: no

Coded in: ( )PL/I ( )ALM ( )other-see below

Performance: ( )better ( )same ( )worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol,sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#) BOS
Info Segs
Other

DOCUMENTATION CHANGES (specify one or more)

HEADINGS are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY: Change the BOS SST name table filler to walk the AST ASTE by ASTE, filling in names from vtoces as he goes. Remove the clever but useless tree walk of the AST.

REASONS: In the pre-4.0 storage system, it was necessary to recursively abs-seg down the AST in order to extract names from directories. This was carried over into the 4.0 SSTN package, as a method of scanning the AST. However, the AST may be linearly walked, and SSTN can identify applicable segments by their non-zero unique ID. This is a fairly safe method, as 28-5 emergency shutdown uses it. This removes BOS' dependence on correctness of the AST-imbedded hierarchy, increasing reliability of the SSTN package.

IMPLICATIONS: Better reliability of BOS.
SUMMARY:

Reinstall changes to init_printer, backup_util, and scs_init which are installed in Phoenix but not at MIT.

REASONS:

The systems should be identical. These changes may have been lost due to the bug in the hardcore updater.

It is not certain whether these fixes are installed in the object code; at any rate, reinstalling the correct version will get the systems in synch again.
TITLE: Recompile all hardcore programs not recently recompiled
AUTHOR: VanVleek

Planned for System: MR 4.0
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: (M)PL/I ( )AIM ( )other-see below
Performance: ( )better (M)same ( )worse

DOCUMENTATION CHANGES (specify one or more)

MMP (vol,sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#)
Info Seqs
Other

OBJECTIONS/COMMENTS:

Headings are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY:
Recompile all hardcore programs not compiled with the installed compilers or the immediately previous PL/I compiler.

REASONS:
Fewer compilers are needed to re-create the system from its source.

IMPLICATIONS:
Better optimization may make the system slightly smaller and faster.
MULTICS CHANGE REQUEST

TITLE: Give volume owner access without ACS

AUTHOR: Van Vleck

Planned for System: not applicable
Fixes Bug Number(s): not applicable
Documented in MIB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: (d)PL/I ( )ALM ( )other—see below
Performance: ( )better (d)same ( )worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol,sect) MPAM (sect)
MSMN (sect) MSAM (sect)
PLMs (Am#)     anal
Info Segs
Other

OBJECTIONS/CUMMENTS:

Headings are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY:

When checking access to a volume, give Initializer.SysDaemon READ access without checking the ACS, and if the ACS does not exist, give the volume owner READ access.

REASONS:

This change means that the owner can always access a volume, even if the ACS is lost.

The change for Initializer.SysDaemon simplifies system cold boot, operation with incomplete HLV, and the management of logical volumes.
Title: Change salvager printing for unattended operation

Author: A. Kobzier

Objections/Comments:

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

Summary:

Change utility_print to switch salvager output to the operator's console if unable to successfully initialize any printer after one minute of trying, and to abandon a printer and repeat initialization if an unrecoverable error is not corrected within three minutes. The time interval will be controlled by "PDLY min" placed on the SALV card.

Reasons:

The ability of the system to purge itself of faulty peripherals is a goal of unattended operation. Thus the salvager should not hang indefinitely waiting for printer corrections, but should switch its output to a device indispensable to Multics operation, i.e., the SYSERR console.
NEW SALVAGER

The storage system structure repair procedures known collectively as the salvager have been extensively modified in MR 4.0, both for automating crash recovery and for handling the structural changes in the storage system data. The salvager programs are now part of the standard system tape; new salvaging modes have been introduced and some old modes redefined; and control of the salvager is now from command options rather than from the processor switches. There are now four kinds of salvaging operation which the system can perform. These are emergency shutdown, volume salvage, directory hierarchy salvage, and online salvage. The system has been modified to invoke the salvager automatically whenever data in the storage hierarchy may have been damaged. The salvager will automatically switch its messages from a printer to the SYSGEN console if it cannot find a readable printer after 1 minute of searching, or if a printer stops and no printer is ready within 4 minutes.

Emergency shutdown has been reworked completely and greatly improved. It is much more reliable than in pre-4.0 systems.

Sometimes the system crashes before the storage system has been turned on. Previously, an attempt to emergency shutdown in such a situation would lead to another crash, and might do serious damage to the directory hierarchy. This has been changed so that if a premature emergency shutdown is attempted, the following message is printed:

    ESD BEFORE FS ENABLED
    SHUTDOWN COMPLETE

and the directory hierarchy is untouched.

It is always safe to attempt an ESD, and if emergency shutdown crashes it may be retried.

Volume Salvaging

Volume salvaging insures that there are no reused addresses or ill-formed VTOC entries in the VTOC and volume map of a single physical volume.

If a volume was in use by Multics and was not demounted or shut down by normal or emergency shutdown, it will be salvaged automatically the next time that the physical volume is accepted for paging.

The operator may force the volume salvage of a particular volume by typing the command

    salvage_vol <vol_name> <drive> <options>

Example: salvage_vol new dska_02 -check_dir
**TITLE:** Restructuring directories by salvager

**AUTHOR:** A. Kobziar

<table>
<thead>
<tr>
<th>Code in</th>
<th>PL/I</th>
<th>ALM</th>
<th>other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explained in DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planned for System MR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fixes Bug Number(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documented in MTB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User/Operations-visible Interface change?</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Incompatible change?</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Performance:</td>
<td>X</td>
<td>Better</td>
<td>Same</td>
</tr>
<tr>
<td>Replaces MCR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Category (Check One):**
- Lib. Maint. Tools
- Sys. Anal. Tools
- Sys. Prog. Tools
- 355
- BOS
- X: Salvager
- Ring Zero
- Ring One
- SysDaemon/Admin.
- Runtime
- User Cmmd/Subr.

**Objections/Comments:**

<table>
<thead>
<tr>
<th>Document</th>
<th>Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPM (Vol, Sect.)</td>
<td></td>
</tr>
<tr>
<td>FLMS (AN #)</td>
<td></td>
</tr>
<tr>
<td>MOSN (Sect.)</td>
<td></td>
</tr>
<tr>
<td>MPAM (Sect.)</td>
<td></td>
</tr>
<tr>
<td>MSAM (Sect.)</td>
<td></td>
</tr>
<tr>
<td>Info Segs</td>
<td></td>
</tr>
<tr>
<td>Other (Name)</td>
<td></td>
</tr>
<tr>
<td>None (Reason)</td>
<td>performance improvement</td>
</tr>
</tbody>
</table>

**SUMMARY:**

Rearrange the structure of a directory when it is rebuilt by the salvager to be as follows:

header, hash table, access name lists, entries (including acls and names), and initial acls.

**REASONS:**

Objects in directories are allocated on a first come, first served basis. Thus shared access names are scattered on several if not all pages of a directory. Although the salvager pre pages (request to read into core all pages of) a directory, it still encounters considerable page waiting time because one of the first operations done is the checking of access names.

**IMPLICATIONS:**

An improvement of 5-10% in the directory salvager's running time was noted at CISL, while a performance run showed an insignificant improvement.
TITLE: Rebuilding Disk Packs

AUTHOR: A. Kobziar

SUMMARY:
Provide for the rebuilding of disk packs in order to add or delete partitions, and/or to increase or decrease partitions and vtocs. The command interface is equivalent to the ones used in volume salvaging disk pack copies and in initializing disk packs. The ability to interlace addresses has been added.

REASON:
Performance optimization and overcoming logical boundaries (for example, running out of vtocs when disk only 2/3 full) are necessary to meet long term changes in usage.

IMPLICATIONS:
Directory salvaging time improved by 5 - 10% on the CISL hierarchy when an interlace =2 was used (i.e. alternate physical address assignment to consecutive directory pages). An insignificant improvement was also noted in performance run testing.
Name: rebuild_disk

Function: Rebuild a disk by copying all information onto a spare disk. Make changes to partitions and/or vtoc as directed.

Syntax: rebuild_disk pvname drive_name1-copy drive_name2 control_args

Arguments:
- pvname is the name of a mounted physical volume
- drive_name1 is pvname's drive in the form <subsys>_ <nn>
- drive_name2 is the spare drive to be used for the copy

Control Arguments:
- -console Output to SYSERR console instead of printer
- -check_dir Delete VTOCE if no branch
- -dump Dump damaged objects
- -debug System programmer use

If the disk drives specified are ok, the command proceeds to list the average used to calculate the VTOC size and the partitions defined on the source disk. The request portion, where partitions must be restated and the VTOC size may be changed is entered when the command types: request.

The following are acceptable request lines:

part NAME HILOW SIZE Where NAME is a 4-character partition name, HILOW is either "high" or "low", and SIZE is the partition size in records

avg FFF.FFF Specifies the average segment length for segments on the physical volume. The default length is that of the original disk. This number is used to calculate the number of VTOC entries on the volume.

vtoc NNN specifies the new VTOC size numerically
lace N requests that physical address assignment be interlaced by N. The default lace is 2 as it is impossible to read in consecutive physical address on the same rotation via the page fault mechanism.

list lists the partitions and VTOC size defined so far.

dis causes the disk rebuild to start. This operation takes about 1/2 hr. for D191's.

quit causes an exit without doing anything.

startover causes all portions defined so far to be discarded. (done automatically if a partition is redefined.)
**Summary:** Most KST overflows, those rarely expected and often fatal gifts from hardcore, are the result of directories which are no longer of interest to a process choking its address space. This situation occurs because processes map many directories into their address space for the sole purpose of accessing inferiors of those directories. Since hardcore has never kept track of the reason for a directory being mapped into a process' address space it has been left to the user to clean up unwanted directory segment numbers. Unfortunately, it is nearly impossible for a process to do the necessary accounting. I propose that hardcore be careful to distinguish those directories which are known for the sole purpose of accessing inferior objects. This will allow uninteresting KST entries to be garbage collected when necessary.

**Reason:** Motherhood and apple pie. Simplify hierarchy walking programs and allow the removal of special case (and often fallible) code.

**Detailed Proposal:** At the heart of the problem is the fact that whenever find_dirsegno, the workhorse of find_, must make a directory known to a process, the usage count of the appropriate KST entry is incremented. Since no one ever decrements this usage count after using the target of the find_, the directory looks as if it is currently in use when in fact this may not be the case. It should be noted that we cannot take the obvious route of not incrementing usage counts when find_ is responsible for making a directory known since usage counts serve the important dual purpose of protecting one ring's segment numbers from termination by an outer ring and of protecting segment number from automatic garbage collection, the very feature we are trying to implement. But enough of this, on to the proposal:
1.) Provide two ring zero primitives to increment and decrement the usage counts kept in KST entries.

2.) Modify find_dirsegno in such a way that when it returns, the usage count of the terminal object has been incremented but its parent usage counts have not. The exact algorithm, which must take into account the pathname associative memory, is too complicated to describe here but it should be noted that find_dirsegno will have to initially increment the usage count of segments it makes known (especially when find_moves out of ring zero). Only when an inferior of a directory made known by find_dirsegno is itself made known can the usage count of the father be decremented, thus establishing the module post condition stated above. As should be obvious, the reason this can be done is that a non-zero inferior count also protects a KST entry from garbage collection. Thus, when find_ returns, the terminal directory segment number involved is explicitly protected from garbage collection by the usage count field and its parents are implicitly protected from garbage collection by having non-zero inferior counts. If the terminal directory segment number left find_unprotected by usage count, then it might be invalidated by KST garbage collection before find_'s caller was finished using it!

3.) To make this scheme workable it is necessary to deal with the usage of the directory recorded by find_ (really find_dirsegno). This must be done by modifying every caller of find_ to call a cleanup procedure when it is done with the segment number in question. One approach would be to call terminate_. If we desired we could even have terminate_recursively inspect the parent of a KST entry when that entry is removed and remove the parent KST entry if its usage counts and inferior count are zero. This scheme has the advantage of keeping the KST ever clean. Unfortunately its a loser. Such a scheme would cause severe KST entry thrashing. The scheme proposed here, which is vastly superior (even if I do say so), is to merely decrement the usage count. In effect this marks the segment number as uninteresting but leaves it in the process' address space for possible future use.

4.) Modify makeknown_ to call a new module, garbage_collect_kst (please, lets not quibble over names), when it detects a KST overflow. Only if garbage_collect_kst fails need a KST overflow be signaled.

5.) Create garbage_collect_kst which locates all "uninteresting" KST entries and terminates them. To perform this task it must (at least logically) walk the hierarchy subtree defined by the KST from the leaves up. This order is necessary since terminating a leaf node may make its father eligible for garbage collection.

Note: The concept of an "uninteresting" segment number (otherwise known as a "twsn" or truly worthless segment number) developed above provides a powerful handle on a problem I shall call the "resource speculation problem". This problem deals with the conflicting desires of a module to release a resource so others may use it and the desire to keep the resource on speculation that it may be needed again in the near future. The introduction of a "twsn" allows a module to in effect have its cake and eat it too. A "twsn" stays bound to the same object, its SDW is not invalidated, nor is its KST entry. Yet, if KST space is exhausted, a mechanism exists for recovering "twsn"s. If the mechanism for creating "twsn"s were available outside of ring zero (a trivial accomplishment), then the expense of the makeknown, terminate, makeknown, terminate, ... sequences which result so frequently from users doing compile, edit,
compile, edit, ... sequences (and from many other sequences of user activities) would be greatly reduced! Many segfaults, setfaults, and directory lockings would be eliminated. As a result, system performance might be significantly enhanced. But this note is just a teaser, a future MCR will deal with this issue in more detail.
TITLE: Improve signalling of *system link

AUTHOR: M. Weaver

SUMMARY:

Fix link_snap to signal linkage_error (bad self reference) for *system links.

Force metering bin number to be non-negative.

REASONS:

Currently link_snap thinks that *system links contain trap-before-links and proceeds to trap to garbage. This produces confusing error messages and also makes it impossible to set up a handler to snap *system links, which is a capability desired to enable new fortran programs with COMMON to be run under Multics.

Occasionally the bin number is negative, which causes the linker to fault.
TITLE: Save copy before message segment salvage

AUTHOR: S. Herbst

<table>
<thead>
<tr>
<th>STATUS</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Written</td>
<td>5/28/76</td>
</tr>
<tr>
<td>Status</td>
<td>12/08/76</td>
</tr>
</tbody>
</table>

-Coded in [ ] PL/I [ ] ALM [ ] Other
-explain in DETAILED PROPOSAL
-Planned for System MR 5.0
-Fixes Bug Number(s)
-Documented in MTB
-User/Operations-visible
-Interface change? [ ] yes [X] no
-Incompatible change? [ ] yes [X] no
-Performance: [ ] Better [X] Same
-Wrong: [ ] Better [X] Same
-Replaces MCR

Category (Check One)

- Lib. Maint. Tools
- Sys. Anal. Tools
- Sys. Prog. Tools

DOCUMENTATION CHANGES

- Document Specify One or More
- BOS
- Salvager
- MPM (Vol. Sect.)
- Ring Zero
- PLMS (AN #)
- Ring One
- SysDaemon/Admin.
- MPAM (Sect.)
- Runtime
- MOSN (Sect.)
- User Cmd/Subr.
- MSAM (Sect.)

Objections/Comments:

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

SUMMARY:

Modify the message segment salvager to make a copy of the message segment it is salvaging in \texttt{\textless dumps\textgreater \textbackslash ms\_salvager\_copy} if it deletes one or more messages in the process of salvaging.

REASON:

Allows a possible analysis of what went wrong.

DETAILED PROPOSAL:

Ring one segment \texttt{\textless dumps\textgreater \textbackslash ms\_salvager\_copy} is written into if empty; contains a copy of the message segment plus its name, date salvaged, etc. The segment will be created by \texttt{asu\_ec} as a ring 1 segment with multi-class privileges. \texttt{ph\_access} needed to read segment.

IMPLICATIONS:

Only the first of a series of salvaged message segments gets saved, until the copy is manually reset to zero. However, we want this diagnostic feature to be implemented as simply as possible.
TITLE: Implement &ec_dir for exec_com's

AUTHOR: Steve Herbst

SUMMARY: Implement the &ec_dir parameter in exec_com's and absin segments, to be replaced by the directory portion of the exec_com or absin segment's pathname. The &0 facility should be removed at some future time. It should remain in exec_com but not be documented.

REASONS: Exec_com's can call themselves by saying:

"&ec_dir>entry_point_name"

The &0 parameter, which is supposed to give the pathname of the exec_com or absin segment as typed, does not work in absentee because that pathname is expanded when the absentee request is submitted. The &0 facility will no longer be necessary when exec_com's and absin segments can call themselves using &ec_dir and &ec_name.

IMPLICATION: Incompatible change. The string &ec_dir currently is replaced by itself.
Argument Substitution

Strings of the form &i in the exec_com segment are interpreted as dummy arguments and are replaced by the corresponding argument to the exec_com command. For instance, optional_arg1 is substituted for the string &1 and optional_arg10 is substituted for &10.

The character & should be followed by a number, i, or by the string ec_name. If no corresponding optional_arg is provided, &i is replaced by the null string. The string &ec_name is replaced by the entryname portion of the exec_com pathname without the ec suffix. The string &10 is replaced by the pathname argument to exec_com, just as it was given to the command.

Argument substitution can take place in command lines, input lines or in control statements, since the replacement of arguments is done before the check for a control statement.

Control Statements

Control statements permit more variety and control in the execution of the command sequences. Currently the control statements are: &label, &goto, &attach, &detach, &input_line, &command_line, &ready, &print, &quit, &if, &then, and &else.

Control statements generally must start at the beginning of a line with no leading blanks. Exceptions to this rule are the &then and &else statements, that can appear elsewhere. Also when a control statement is part of a THEN_CLAUSE or an ELSE_CLAUSE, it does not have to start at the beginning of a line.

1. &label and &goto

These statements permit the transfer of control within an exec_com segment.

&label location identifies the place to which a goto control statement transfers control. location is any string of 32 or fewer characters identifying the label.

&goto location causes control to be transferred to the place in the exec_com segment specified by the label location. Execution then continues at the line immediately following the label.
MULTICS CHANGE REQUEST

TITLE: Change TEST and FMT to query before destroying labeled pack
AUTHOR: VanVleck

Planned for System: MR 4.0
Fixes Bug Number(s): not applicable
Documented in MTS: not applicable
Incompatible Changes: no
User/Operations-visible Interface Changes: no
Coded into (B)PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol, sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#) bos
Info Segs
Other

OBJECTIONS/COMMENTS:

SUMMARY:

Caused TEST and FMT to read the label of the volume they are asked to write on; if this is a valid Multics label, to ask

DSKA 12 IS MULTICS STORAGE SYSTEM VOLUME RPV
DO YOU WISH TO WRITE ON DSKA 12?

and to abort unless the operator answers "yes."

REASONS:

Bitter experience (05/29/76) has shown that this is a possible operator error.

IMPLICATIONS:

Less downtime.
### TITLE: Make drive number arguments to BOS be decimal

**AUTHOR:** VanVleck

**Planned for System:** MR 4.0

**Fixes Bug Number(s):** not applicable

**Documented in MIB:** not applicable

**Incompatible Change:** no

**User/Operations-visible Interface Change:** no

**Code in:** ( )PL/I ( )ALM ( )other-see below

**Performance:** ( )better ( )same ( )worse

---

**DOCUMENTATION CHANGES (specify one or more)**

- **MPM (vol, sect):** MPAM (sect)
- **MOSN (sect):** a001 MSAM (sect)
- **PLMs (AN#):**
- **Info Segs:**
- **Other:**

---

**OBJECTIONS/COMMENTS:**

---

#### SUMMARY:

Make all drive numbers input by the operator to BOS be interpreted as decimal numbers only.

This change affects IF, FLAG, ABS, SAVE, RESTOR, BOOT, CORE, DUMP, LABEL, LOADOM, PRINT, TAPED, and all callers of ARGBOS such as DUMP and PATCH.

(The contents of the CONFIG deck and the COLO, WARM, and NLABEL cards are not affected.)

#### REASONS:

Allowing drive numbers to be either decimal or octal means that the operator may make a disastrous error by omitting a period.

This problem occurred at MIT on 05/29/76.

The primary need is to make disk drive numbers be decimal. Tape drive numbers are made decimal also in order to avoid confusion.

#### IMPLICATIONS:

Less downtime.
TITLE: To install a merged fortran_io_

AUTHOR: G. Chang

SUMMARY:

To install a merged fortran_io_ to replace the current fortran_io_ and the current fast_fortran_io_.

REASONS:

The fast_fortran_io_ was intended to be an interim version in the first place.

DETAILED PROPOSAL:

A copy on write fortran_buffer_segment is also to be installed.
MULTIOS CHANGE REQUEST

TITLE: Better handling when disk drops off line.

AUTHOR: VanVleck / N. Morris

Planned for System: MR 4.1

Fixes Bug Number(s): not applicable

Incompatible Change: no

User/Operations-visible Interface Change: no

Performance: ( ) better ( ) same ( ) worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol, sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#) an61
Info Segs
Other

OBJECTIONS/COMMENTS:

---

SUMMARY: Currently if a disk drive falls into standby we retry forever, logging messages and printing every tenth. This degrades system performance severely.

PROPOSAL: If the drive "fixes itself" within 5 seconds and sends a special interrupt, retry the I/O. If 5 seconds elapse without a special interrupt, try to use the drive again. If drive is still inoperative, place in broken state. In this state, all read operations are posted as errors and all write operations are left in the disk request queue until the drive is repaired. An attempt will be made every 10 minutes to use a drive in broken state.

Change disk DIM to log detailed status after every error which calls for detailed status to be read.

REASONS: The change to device-in-standby handling will allow the system to keep going if a disk falls offline, without overloading the CPU and the SYSSERR log.

Repeating the message every 10 mins. will encourage operations to fix the problem sooner.

Printing the error messages and detailed status has been requested by FE, who would like to find out why the disk drives go standby.

IMPLICATIONS: More reliable operation.

Page 1 of 1
### SUMMARY

Any error encountered in the writing of a VTOC, which cannot be confined by the disk dim, currently crashes the system. This is because a previous policy whereby irreversible action was taken after the start of VTOC writing which would assume correct completion of the write at some later time. Since there is now a mechanism which can await correct completion of a VTOC write (covered under previous MCR), irreversible action can be postponed until correct completion of the write has been assured. Remove the code from vtoc_interrupt which crashes the system on a write failure.

### REASONS

Greater reliability. Not only do fatal VTOC write errors crash the system, but they tend to leave the VTOC buffer segment in an inconsistent state.

### IMPLICATIONS

See "DETAILED PROPOSAL".

### DETAILED PROPOSAL

Introduce the concept of a "hot" VTOC buffer, that is, one whose contents represent a known difference from the disk-resident VTOC, but for which no I/O is outstanding. The unsuccessful completion of a write operation puts a buffer in this state. Reads may be done from such a buffer, and all future calls to write it retry the operation. Demounting of the physical volume (which includes shutdown) tries once to flush all hot buffers. Any volume for which a hot buffer is found at demount time causes a volume salvage to be scheduled, as does a VTOC I/O error code returned to a caller of await_vtoc. This is because either of these cases almost ensures that records were deliberately left undeposited, and need be collected.

An implication of this strategy is that hot buffers cannot be
replaced by the VTOC buffer selection algorithm. Hence, the system can fill up with hot VTOC buffers in a limiting case. We do not propose to solve this in this pass, but await a better policy on system write errors in general, which is in the offing. The proposal as it stands is a major increase to system reliability.
TITLE: Install new bound_plio2_

AUTHOR: R. Schoeman

STATUS DATE
Written 6/4/76
Expires 12/15/76

Category (Check One)
- Lib. Maint. Tools
- Sys. Anal. Tools
- Sys. Prog. Tools

DOCUMENTATION CHANGES
- Document Specify One or More
- MFM (Vol. Sect.)
- PLMS (AN #)
- MOSN (Sect.)
- MSAM (Sect.)

Objections/Comments:
- Info Segs
- Other (Name)
- None (Reason) bugfix

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

REASON:

Fix a bug in opening stream files in PL/I I/O caused because plio2 was using iox_$modes to get line length instead of get_line_length.
MULTICS CHANGE REQUEST

TITLE: Fix descriptor segment bound field

AUTHOR: VanVleck

Planned for System: MR 4.0
Fixes Bug Number(s): not applicable
Documented in MTR: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: (X)PL/I ( )ALM ( )other-see below
Performance: ( )better (X)same ( )worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol, sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLM (AN#) anol
Info Sens
Other

OBJECTIONS/COMMENTS:

SUMMARY:

In systems 28-5 on, the descriptor segment boundary is set to 1K instead of 4K. Change it back to 4K.

REASONS:

Setting the bound field to 1K can lead to out-of-bounds faults on the descriptor segment, when the KST says that the maximum segment number is 1023. This bounds fault can crash the system.

IMPLICATIONS:

Fewer crashes and crawlouts.
**Title:** Prevent message coordinator flooding  

**Author:** Van Vleck  

**Planned for System:** not applicable  
**Fixes Bug Number(s):** not applicable  
**Documented in VTBS:** not applicable  
**Incompatible Change:** no  
**User/Operations-visible Interface Change:** no  
**Coded in:** ( ) PL/I ( ) AL ( ) Other-see below  
**Performance:** ( ) better ( ) same ( ) worse  

**DOCUMENTATION CHANGES (specify one or more):**  
MPIL (vol, sect)  
MOSM (sect)  
SA (sect)  
INPS (AMF)  
Other  

**OBJECTIONS/COMMENTS:**  

---  

**Headings are:** SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)  

**SUMMARY:**  
If a process sends messages to the message coordinator faster than they can be printed, cause the process to go into output wait status.  

**REASONS:**  
Currently if a daemon process goes into an output loop (printer drivers do this sometimes) the message coordinator receives messages faster than its terminals can print. This causes the physical device queues to fill up with unprinted messages, and the message router then takes an out-of-bounds on a queue. The out-of-bounds handler causes a message to be printed; this message gets a recursive bound fault and terminates the process, crashing the system.  

When this problem was first encountered, the printer DIM was modified to sleep for one minute if too many errors occurred. But that change did not catch all the possibilities for message loops.  

**IMPLICATIONS:**  
Fewer crashes.
DETAILED PROPOSAL:

Implement a per-source counter in the MRT which counts up when a source has sent a message and counts down when the message has been output. Modify mrd_util_write to set a flag and block the process if this count exceeds a preset number. If the counter goes to zero and the process is waiting, wake it up.
SUMMARY:

Issue VTUC I/O requests in order of vtoc index when listing directories in dc_pack.

REASONS:

Sorting these requests by vtoc index will reduce the disk arm motion caused by the dumper.

IMPLICATIONS:

Performance improvement. Items are returned in the same order as before so no interface change is evident.
**Title:** Fix problems in Network routines -- supervisor

**Author:** D. Wells

**Category:** Lib. Maint. Tools

**System:** IMP DIM

**Status:** Written 7.6.76

**Expires:** 12.15.76

**Fixes Bug Number(s):** 4.0

**Documented in:** MTB

**Coded in:** XFL/T

**Category:** Lib. Maint. Tools

**Status:** A

**Compatibility:** Other

**Document:** Specify one or more

**Code:** other

**Comments:** Info Segs

**Other:** None

**Reason:** None

---

<table>
<thead>
<tr>
<th>Objections/Comments:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Info Segs</td>
</tr>
<tr>
<td>- Other (Name)</td>
</tr>
<tr>
<td>- None (Reason)</td>
</tr>
</tbody>
</table>

---

**Use these headings: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)**

**SUMMARY:** Fix two problems in IMP DIM and NCP: 1) add fault handler to NCP assignment module to trap and handle faults and signals, 2) add check to IMP DIM to avoid a zero length move while running on the PRDS.

**REASONS:** These problems cause the system to crash.

This is an emergency installation.
**TITLE:** Delete old list command  
**AUTHOR:** T. Casey  
**STATUS**  
**DATE**  

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixes Bug Number(s)</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planned for System MR 4.0, 5.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documented in MTB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User/Operations-visible</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface change? Yes No</td>
<td>Yes</td>
<td></td>
<td></td>
<td>MPM (Vol, Sect.)</td>
</tr>
<tr>
<td>Incompatible change? Yes No</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance:</td>
<td>Better Same</td>
<td></td>
<td></td>
<td>Ring Zero</td>
</tr>
<tr>
<td>Replaces MCR</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPM (Vol, Sect.)</td>
</tr>
<tr>
<td>PLMS (AN #)</td>
</tr>
<tr>
<td>MOSN (Sect.)</td>
</tr>
<tr>
<td>MPAM (Sect.)</td>
</tr>
<tr>
<td>MSAM (Sect.)</td>
</tr>
<tr>
<td>Info Segs</td>
</tr>
<tr>
<td>Other (Name)</td>
</tr>
</tbody>
</table>

Objections/Comments:  
None Reason doc. ok

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

**SUMMARY:**

For release MR 4.0, delete all of the names (In, lt, listnames, list_totals, list_names, list_totals) from old_list.

For release MR 5.0, delete old_list.

**REASONS:**

Typing ln or lt invokes the old list command, which does useless VTOC I/O on every entry matching the given starnames, even though that information is not to be printed.

**IMPLICATIONS:**

Users who have not converted to using the new list command will be forced to. Removing the names but postponing the deletion for a while makes their conversion slightly more convenient.
## Summary of Proposal

### Fix term_ to always unsnap links.

### Reasons:

Some term_ entry points, e.g., term_$nomakeunknown, do not call link_unsnap_ and it is possible for a segment to be terminated while links to it remain. In particular, the delete command can delete in some cases without unsnapping links.
TITLE: Bug fixes for debug

AUTHOR: S. Barr

- Coded in [X] PL/I [ ] ALM [ ] other
- Planned for System MR 4.0
- Fixes Bug Number(s) 355
- Documented in MTB
- User/Operations-visible
  Interface change? [X] yes [ ] no
  Incompatible change? [X] yes [ ] no
- Performance: [X] Better [ ] Same
- Replaces MCR

Category (Check One)

Lib. Maint. Tools
Sys. Maint. Tools
Sys. Anal. Tools
Sys. Prog. Tools
DOCUMENTATION CHANGES

Objections/Comments:

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

REASONS:

1. debug occasionally prints out an incorrect error message "path now references a new segment" for the "bgr" request.
2. debug doesn't print the argument value in the case where there were no descriptors for the "apl" request.
3. If debug was entered via a mm2 because of the static handler mechanism and the user tries to quit, debug tries to return to the mm2 fault.

PROPOSAL:

Bugs described in #1 and #2 are one line fixes.

For #3 debug will follow this convention:

If the user tries to quit from debug, debug will return from the stack frame of the last time debug was called. If debug$mm2_fault was invoked via a static handler and there is no previous called frame for debug, then the "q" request will cause the condition command_abort to be signalled.

This will be implemented by having a count that will be incremented each time debug is called and decremented before debug returns, so that the presence of a previous debug invocation will be known.