To: Distribution
From: Betsy
Date: April 5, 1976
Subject: MCR's 3/16 - 31/76

Attached are the approved MCR's from March 16 - 31, 1976
SUMMARY: 1. Allow the directory to be prelinked to be specified by relative pathname.

2. If no pathname is given, have the prelink command prelink the subsystem defined by the pldt in the working directory.

3. Implement a new option to the prelinker, "-delete" or "-dl", which specifies that the prelinked system contained in the specified directory is to be deleted rather than recreated.

REASONS: Make prelinker easier to use.
**SUMMARY:**
1. Do a resetread before reading login line.
2. Do a resetread after bad login words.
3. Fix DETACH to not fault.
4. Fix test mode to work.
5. Fix fault recovery problems in as dump.
6. Fix answering service to change terminal modes (and line length) to defaults after logout or a fatal process error, and not after new proc.

**Reasons:**
1. GM has reported problems with terminals that have multi-line answerbacks. The resetread is necessary to discard any "left-over" answerback characters.

2. It is possible, by fast typing, to get the answering service to print your password if you misspell login.


5. as_dump shouldn't invoke the message coordinator when it is called by the message coordinator.

6. Modes are advertised as per-session, not per-process. They should be reset after a fatal process error in case the process was in raw input or raw output mode.

**Implications:**
1. Will not be able to type ahead login line (and password).

2-6) None.
Ver. 5
14022
MULTICS CHANGE REQUEST
JWG

TITLE: Temporary emergency fix to tty_read

AUTHOR: Robert S. Coren

Planned for System: 3.1
fixes bug number(s): not applicable
Documented in VMD: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Code in (d)RL1 ( )ALa ( )other-see below
Performance: ( )better (d)same ( )worse

LSSISMISUSE CHANGES (specify one or more)
(d)MAL1 (sect) MAL2 (sect)
(d)MAL1 (sect) MAL2 (sect)
PL1 (sect)
INFO sects
OTHER

ONE (reason): bug fix

SOCTIONS/COMMENTS:

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

SUMMARY: this bug whereby faulty prescan in tty_read was causing confusion in tty_write, which has caused at least two crashes at MLI.

REASONS: crashes are undesirable.

IMPLICATIONS: none.

DETAILED PROPOSAL: remove update of column position from tty_read prescan procedure.

NOTE: the new version of the tty_dlm planned for 3.1 will eliminate this problem since prescan will be removed. This note proposes a quick and dirty interim fix.
TITLE: Install prelink driver tables for FAST and DFAST

AUTHOR: S. Barr

- Coded in [X] FL/I [ ] ALM [ ] other-
- explain in DETAILED PROPOSAL
- Planned for System MR 3.1
- Fixes Bug Number(s)
- Documented in MTB
- User/Operations-visible
- Interface change? [ ] yes [ ] no
- Incompatible change? [ ] yes [ ] no
- Performance: [ ] Better [ ] Same [ ] Worse
- Replaces MCR


expires: 3/9/76

Document | Specify One or More

DOCUMENTATION CHANGES

Fixes Bug Number(s)

- Planned for System MR 3.1 Sys. Anal. Tools
- Documented in MTB
- User/Operations-visible
- Interface change? [ ] yes [ ] no
- Incompatible change? [ ] yes [ ] no
- Performance: [ ] Better [ ] Same [ ] Worse

Objections/Comments:

SRE doc required

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

Summary:

Install the prelinked driver tables in >tools in an analogous manner to the administration exec_coms.

Reasons:

Prelinked subsystems must be prelinked at each boot of the system. A call to do this must be added to the system start_up.ec and a directory for each subsystem to be prelinked must be created to hold the pldt and the associated segments.

Detailed Proposal:

Install fast.pldt and dfast.pldt in >tools.

Implications:

A site that wishes to run a prelinked version of FAST or DFAST subsystems will follow these steps:

1) Create a directory for the subsystem.
2) Create a link in that directory to the proper pldt with the name "pldt". (If the site wished to modify the driver table, it would copy the original segment from >tools.)
3) Add a line to part 2 of the system_start_up.ec of the form:
   prelink path
   where path is the absolute pathname of the directory to be prelinked.
**TITLE:** Fix Bugs in cv_cmf

**AUTHOR:** T. Casey

**STATUS**

**DATE**

---

**Title:**

**SUMMARY:**

Change error checking in cv_cmf so that:

1) The check for printable ascii characters in comment statements allows all printable characters, including tabs and other special characters;

2) Brief error messages (printed at the second and subsequent occurrences of the same error) will contain some part of the offending statement, where appropriate, rather than just the error number.

**REASONS:**

The current cv_cmf will not compile the cmf that is generated by the create_cmf command, and the error messages produced by the attempt to compile it consist of nothing but the error number and severity, with no indication of which statement is in error.
**TITLE:** Allow ntape\_ to handle multifile tapes

**AUTHOR:** Richard Bratt

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Document Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td>MPM (Vol, Sect.) AG93</td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td>PIMS (AN #)</td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td>MOGN (Sect.)</td>
</tr>
<tr>
<td></td>
<td>MPAM (Sect.)</td>
</tr>
<tr>
<td></td>
<td>MSAM (Sect.)</td>
</tr>
</tbody>
</table>

**Objections/Comments:** Pending changes notice, clarify user doc on passing end of real information

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

**SUMMARY:** Currently ntape\_ refuses to read or skip past a tape mark. This renders it useless in reading multi file tapes.

**PROPOSAL:** allow ntape\_ to read or skip past a tape mark.

**IMPLICATIONS:**

1. Many more tapes will be readable by ntape\_
2. Current programs which depend upon read returning error_table\_end_of_info repeatedly when a tape mark is encountered may stop working. However, since ntape\_ fails to detect end of information (sec MCR 1678), the number of programs in this class most be zero.
**Title:** Install new command `compare_command_output`  

**Author:** Steve Herbst  

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Coded in:** XPL/I □ ALM □ other-  
  - explain in DETAILED PROPOSAL  
  - Planned for System MR 4.0  
  - Fixes Bug Number(s) X  
  - Documented in MTB 355 X  
  - User/Operations-visible  
    - Interface change? □ yes □ no  
    - Incompatible change? □ yes □ no  
    - Performance: □ Better □ Same □ Worse  
    - Replaces MCR  

<table>
<thead>
<tr>
<th>Objections/Comments:</th>
<th>explain how it affects environment in user doc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>Specify One or More</td>
</tr>
<tr>
<td>Info Segs</td>
<td></td>
</tr>
<tr>
<td>Other (Name)</td>
<td></td>
</tr>
<tr>
<td>None (Reason)</td>
<td></td>
</tr>
</tbody>
</table>

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

**Summary:**  
Install the new command `compare_command_output`, used to compare the operation of two different versions of the same command.
Name: compare_command_output

This command executes two versions of another command and compares their output.

Usage:

```
compare_command_output pathA pathB
  -control_args- --ag args-
```

where:

1. pathA is the pathname of a command.
2. pathB is the pathname of a different version of the same command.

3. control_args can be:

   -after LINE Execute the command line LINE (in quotes) after executing each version of the command being tested.

   -before LINE Execute the command line LINE (in quotes) before executing each version of the command being tested.

   -brief Direct both user_output and error_output to temporary segments in the process directory, and print only the discrepancies. The default is to leave user_output and error_output attached as they are.

4. -ag args specifies the arguments to be passed to the two commands pathA and pathB. All arguments to compare_command_output following -ag are interpreted as such.

Notes:

The compare_command_output command executes the following in order:

1. The -before command line, if present.
2. The version specified by pathA.
3. The -after command line, if present.
4. The -before command line again.
5. The version specified by pathB.
6. The -after command line again.

If the -brief control argument is specified, user_output and error_output are both attached to a temporary segment for A's output, steps 1, 2, and 3 are executed, user_output and error_output are re-attached to another temporary segment for B's
output, and steps 4, 5, and 6 are executed. The compare_ascii command is invoked then to compare the contents of the two temporary segments.

To compare the same two commands with a variety of arguments, it is convenient to define an abbreviation for the beginning of the compare_command_output command line through -ag.

Example:

To compare the operation of two versions of the archive command on the same archive, keep a permanent archive somewhere and use a -before command line to make a temporary copy. Use an -after command line to print the contents of the archive after each execution.

```
compare_command_output >sss>ac ac -brief
  -before "copy real.archive temp.archive"
  -after "pr temp.archive 1;dl temp.archive"
```

Because of the -brief control argument, the user sees only the discrepancies in the printed output, labelled A and B.
Name: ntape_

This I/O module supports I/O from/to files on magnetic tape.

Entry points in the module are not called directly by users; rather, the module is accessed through the I/O system. See "Multics Input/Output System" for a general description of the I/O system, and see "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:

ntape_ reel_num -control_arg -optional_args-

where:

1. reel_num is the tape reel number. If the tape is 7-track, reel_num must contain ",7track". If the tape is 9-track, reel_num may contain ",9track" (if it contains neither, 9-track is assumed).

2. control_arg must be -raw to indicate that each physical record (block) on the tape represents one logical record.

3. optional_args may be one of the following arguments:
   -write means that the tape is to be mounted with a write ring. This argument must occur if the I/O switch is to be opened for output or input/output.
   -extend specifies extension of the file if it already exists on the tape.

Opening

The opening modes supported are sequential_input, sequential_output, and sequential_input_output. If an I/O switch attached via the ntape_ I/O module is to be opened for output or input_output, the -write argument must occur in the attach description.

Control Operation

This I/O module does not support the control operation.
**Modes Operation**

This I/O module does not support the modes operation.

**Notes:**

A.) Using the -raw control argument, the relation between logical and physical record is as follows:

1. On input, the logical record contains \( m = 4 \cdot \lceil n/36 \rceil \) bytes, where \( n \) is the number of data bits in the physical record. The first \( n \) bits of the input record are the data bits, the last \( (9 + m - n) \) bits are 0's.

2. On output, the physical record contains \( n = k \cdot \lceil (36 \cdot \lceil m/4 \rceil)/k \rceil \) data bits, where \( k + 1 \) is the number of tracks on the tape, and \( m \) is the length of the logical record in characters. The first \( 9 + m \) data bits of the physical record contain the bits of the logical record (i.e., the output buffer). The last \( (n - 9 + m) \) bits of the physical record are 0's.

B.) Tape allows a user to read and skip past tape marks. It is then the user's responsibility to detect the logical end of tape.

That is, an error message and of info is returned when a tape mark is detected but subsequent operations are allowed.
SUMMARY:

1) Modify `tape_ansi` and `tape_ibm` to return a data record even though that record was read with a parity-type error. Currently no data record is returned.

2) Provide a new control operation, "reset_error_lock", that resets the logical record I/O lock only if its value is `error_table` or `tape_error` (indicating a parity-type error), and only if the I/O switch is open for `sequential_input`. Currently, the I/O switch is permanently inhibited from further reading once a parity-type error occurs.

REASONS:

1) It is desirable to have access to the bad record, either to print it, attempt to use it, etc.

2) Many users have expressed the desire to be able to continue processing once such a bad record has been read. The specific mechanism of a one-time-only unlocking operation is proposed because continuing to process can have non-trivial results. If records are blocked, there is no way of determining which of the 0 records in a block actually contains the bad data. The record for which the tape error is returned may in fact be completely valid. If the record format is spanned, SCW’s (segment control words) may have been insidiously modified so as to invalidate subsequent blocks, etc. Nevertheless, some mechanism that is not amenable to careless use should be provided.
IMPLICATIONS:

1) `iox_read_record` calls which return `error_table_tape_error` will no longer necessarily also return non-zero record lengths.

2) `iox` level users will be able to recover from parity-type errors, completely reliably in `U`, `F`, `D`, and `V` formats, less so in `FB`, `VB`, `DB`, `S`, `SB`, `VVS`, and `VBS`.

DETAILED PROPOSAL: see attached
Error Processing

(See reset_error_lock OPERATION, below.)

If an error occurs while reading, the I/O module makes 25 attempts to backspace and reread. If an error occurs while writing, the I/O module makes 10 attempts to backspace, erase, and rewrite. Should an unrecoverable error occur while reading or writing, the I/O module "locks" the file so that no further I/O is possible. If an unrecoverable error occurs while writing file labels or tapemarks, the user is queried about preserving the defective file versus file set consistency. (See "Queries" above.) If an unrecoverable error occurs during certain phases of volume switching or label reading, the I/O switch may be closed. The overriding concern of the error recovery strategy is:

1. to maintain a consistent file set structure
2. to ensure the validity of data read or written

Close Operation

The I/O switch must be open.

Control Operation

The I/O module supports nine control operations.

reset_error_lock
hardware_status
status
volume_status
file_status
feov
close_rewind
retain_all
retain_none

In the descriptions below, info_ptr is the information pointer specified in an iox_$control entry point call.

reset_error_lock OPERATION

This operation "unlocks" the file so that further I/O is possible subsequent to a parity-type I/O error while reading. Such an error is indicated by a previous iox$_read_record or iox$_position call having returned the status code error_table$tape_error. In this case, the value of tape_file_status.event_lock is error_table$tape_error. The I/O switch must be open for sequential_input. The info_ptr should be a null pointer.

NOTE: IF RECORDS ARE BLOCKED AND/OR SPANNED, THE VALIDITY OF THOSE RECORDS READ SUBSEQUENT TO A PARITY-TYPE I/O ERROR IS NOT GUARANTEED.
**TITLE:** Solve MCS input interruption problems  
**AUTHOR:** Grady

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Status</th>
<th>Expire</th>
<th>Document</th>
<th>Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td>A</td>
<td>3/16/76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Status:** A  
- **Expire:** 3/16/76

<table>
<thead>
<tr>
<th>-Coded in</th>
<th>-Planned for System MR 4.0</th>
<th>-Fixes Bug Number(s)</th>
<th>-Documented in MTB</th>
<th>-User/Operations-visible</th>
<th>-Incompatible change?</th>
<th>-Performance:</th>
<th>-Replaces MCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑ PL/I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XXX yes</td>
<td>XXX Same</td>
<td></td>
</tr>
<tr>
<td>-explain in DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XXX no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XXX Better</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SUMMARY:** Implement the solutions to the input interruption problems described in MTB-259

**REASONS:** These problems are annoying and have been the subject of many MPRF'S

**DETAILED Proposal:**  
Coded in 355 map

**IMPLICATIONS:** command query and accept message can be changed to not issue extra new line

Objections/Comments:  
Pending changes, notification of incompatible changes

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.
Ver. 3
MULTICS CHANGE REQUEST

Title: Fix gen_sst_card for new storage system.

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, ( )same, ( )worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol, sect) MPAM (sect)

MODS (sect) MSAM (sect)
PLMs (AN#) tools
Info Segs
Other

CATEGORIES/COMMENTS:

SUMMARY: Change gen_sst_card to work for the new storage system.

REASONS: The size of the ASTE and the layout of the SST changed with system 29-C.

IMPLICATIONS: None
MULTICS CHANGE REQUEST

TITLE: Reload logical volume ID

AUTHOR: VanVleck

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

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

DOCUMENTATION CHANGES (specify one or more)

MPM (vol, sect) MPAM (sect)
MCSN (sect) aj01 MSAM (sect)
PLMS (AN#)
Info Seqs
Other

OBJECTIONS/COMMENTS:

SRB explanation needed

SUMMARY: Modify the reloader to reload the sons lvuid attribute of a directory. Add a new argument, "-no_lv" to the reloader to indicate that the logical volume ID should be ignored while reloading. Verify that all logical volume ID's used are valid before inserting them into the hardcore.

FEASIONS: When an installation does a complete reload, the result of the reload should be the same as the hierarchy when it was dumped, as far as logical volume membership. The "-no_lv" argument is necessary because otherwise attempts to reload a dump tape made at one installation onto another installation would incorrectly attempt to put the logical volume unique ID into the hardcore. When this action is done as part of a cross-system reload, the unique ID is invalid and the contents of the tape cannot be loaded.

IMPLICATIONS: An extra call to the hardcore and a call to mic will be made for each directory reloaded in the regular case; in the cross-site case an additional argument must be specified.

Since hnhcs_set_sons_lvuid doesn't set the sons lvuid if there are already segments in the directory, in order to avoid creating invalid directories with sons on more than one volume, an installation may over-ride the LVIDs on a reload tape by creating directories in advance, setting the sons lvuid as desired, creating a segment in them, and reloading with "-notrim".

Page 1 of 1
MULTICS CHANGE REQUEST

TITLE: Make I/O priority depend on scheduling priority.

AUTHOR: Van Vleck

Planned for System: MR 4.0

Fixes Bug Number(s): not applicable

Incompatible Change: ro

User/Operations-visible Interface Change: no

Tested in: PL/I

Performance: ( ) worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vcl, sect) WPAM (sect)
MSGN (sect) WSAM (sect)
PLMs (AN#) an61
Info Seqs

OBJECTIONS/COMMENTS:

Observe performance at MIT when installed.

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

SUMMARY: The disk DIM gives priority to some I/O requests and not to others. Currently all page and VTOCE reads have priority and no writes have priority. Change the disk DIM so that page reads and VTOCE reads and writes have priority if issued by a process which has its prds$level (depth in the traffic controller eligibility queue) less than the number of CPU's on the system.

REASONS: Performance experimentation indicates that this change allows the processes with the highest dispatching priority to complete their I/O faster, and so improves system throughput and response. More complicated optimizations have been proposed, such as sorting the requests according to multiprogramming depth, and these will be evaluated and considered for later installation. This change is easy and should provide much of the expected benefit.

IMPLICATIONS: In tests on the CISI machine, an improvement of about 3 to 5 percent was noticed.
**MULTICS CHANGE REQUEST**

**TITLE:** Per-drive metering in disk DIM

**AUTHOR:** Van Vleck

**Planned for System:** MR 4.0

**Fixes/Rcu Number(s):** not applicable

**Documented in MTB:** not applicable

**Incompatible Change:** no

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

**Documented in: (Y) PL/1 ( ) ALM ( ) other-see below

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

### DOCUMENTATION CHANGES (specify one or more)

<table>
<thead>
<tr>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPM (vol,sect)</td>
<td>MFA (sect)</td>
</tr>
<tr>
<td>MCRM (sect)</td>
<td>MSAM (sect)</td>
</tr>
<tr>
<td>PLMS (AN#)</td>
<td>AN61</td>
</tr>
<tr>
<td>Info Sens</td>
<td>Other</td>
</tr>
</tbody>
</table>

**OBJECTIONS/COMMENTS:**

**READINGS ARE:** SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

**SUMMARY:** Add metering to the hardcore disk DIM to measure the number of VTC'C and page reads and writes and the total seek distance on a per-drive basis. A command to print meters will be made available.

**REASONS:** When attempting to analyze the performance of the system we need to be able to confirm or deny the existence of an I/O bottleneck on a particular drive.

**IMPLICATIONS:** More wired-down storage is required; 192 words per disk subsystem.

---

**Page 1**
TITLE: Install FAST run unit manager

AUTHOR: M. Weaver

CODED IN: X PL/I ALM other

EXPLAIN IN DETAILED PROPOSAL

PLANNED FOR SYSTEM MR 3.1

FIXES BUG NUMBER(S)

DOCUMENTED IN MTB

USER/OPERATIONS-VISIBLE

INTERFACE CHANGE? X yes

INCOMPATIBLE CHANGE? X yes

PERFORMANCE:

BETTER

SAME

WORSE

REPLACES MCR

CATEGORY (CHECK ONE)

LIB. MAINT. TOOLS

SYS. ANAL. TOOLS

SYS. PROG. TOOLS

DOCUMENTATION CHANGES

DATE: Written 3/10/76

STATUS: Expires 9/10/76

SUMMARY OF PROPOSAL, REASONS FOR PROPOSAL, IMPLICATIONS, DETAILED PROPOSAL.

Summary: Install the run unit manager for FAST.

Reasons: Needed to run user programs in FAST. A general Multics run command would take too long to implement for 3.1 and may not be ideal for FAST anyway.

Detailed Proposal: The run unit manager implements most of the FAST run command (any compilation is done beforehand). In FAST, all user programs are executed within a run unit so that each program's storage can be isolated. The run unit's search rules are the main program* and working directory only. Fortran and basic programs may not be mixed within a run unit. pl/1 and alm programs may be called but they do not run in the run unit in that any static or links are handled by the system linker and remain after the run unit terminates. A single temporary scratch segment is used to store name lists, linkage sections, etc. and for language runtime use; this is truncated at the end of the run unit.

The run unit manager's steps are as follows:

1) Identify all subprograms in main program if it was compiled or part of the run.

2) Allocate all common blocks and snap all links. This step applies only to fortran programs.

3) Execute programs.

4) Recover all storage, terminating basic/fortran segments.

*If the program was compiled by the run command.
The run unit manager will be called by the FAST command processor as follows:

```
decrare fast_run_unit_manager_ entry(ptr, fixed bin(24),
   1 aligned,
   2 bit(1) unal,
   2 bit(1) unal,
   2 bit(1) unal, char(32) varying, fixed bin(35));
```

call fast_run_unit_manager_ (object_ptr, object_bc, flags, main_ename, code);

where:

1. **object_ptr**
   - points to the main program to be executed. (Input)

2. **object_bc**
   - is the bit count of the main program. (Input)

3. **flags**
   - is a structure of control info declared as follows:

```
dcl 1 flags aligned,
   2 just_compiled bit(1) unal,
   2 brief bit(1) unal,
   2 probe bit(1) unal,
   2 mbz bit(33) unal;
```

and

a. **just_compiled**
   - is ON if the object segment was just compiled as a result of this run command and its entry names should be used as the "artificial" reference names. Also its static and linkage will not be copied.

b. **brief**
   - is ON if warning messages such as unresolved links are to be inhibited. This may somehow be associated with ready_on / ready_off.

c. **probe**
   - is ON if the program is to be run under control of the debugger. (not initially implemented.)
4. mainename is the name of the main entry point. (Input)
5. code is a returned status code. (Output)

The following entry is called by the basic linker:

```
declare fast_run_unit_manager_find_entry_value entry (char(32),
    ptr, fixed bin (35));

call fast_run_unit_manager_find_entry_value (entname, entptr, code);
```

where:

1) entname is the name of the entrypoint to find. (Input)
2) entptr points to the specified entrypoint. (Output)
3) code is nonzero if entname is not found in the run unit's name space.
The indent command currently will indent the line:

\texttt{\langle NP\rangle}

to

\texttt{\langle TAB\rangle\langle NP\rangle}

where \texttt{\langle TAB\rangle} is a tab to the current indent level. Also, the line is considered non-blank, so that a comment on the next line (actually the top of the next page) is indented to column 60.

A more reasonable way to handle this case would be not to indent the \texttt{\langle NP\rangle}, and consider the line blank so that the following comment would not be indented.

**Detailed Proposal**

The indent command will change every line consisting of white space and a new-page character to the sequence \texttt{\langle NP\rangle}, which will be considered a blank line. Lines inside of quoted strings, of course, would not be affected.
MULTICS CHANGE REQUEST

TITLE: Implement iox_ control operations from command level
AUTHOR: Larry Johnson

Planned for System: 4.0
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
Replaces MCR: 1497

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

OBJECTIONS/COMMENTS:

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

Summary
Implement a new order call for the io_call command to use to perform control operations. Ultimately, all I/O modules should support this new order.

Reasons
The current io_call command is unable to implement many of the control operations provided by the I/O modules because it does not attempt to provide the info structure they require. The pointer to this structure is always set to null(). It is not feasible for io_call to construct and/or interpret these structures because there are so many different formats (at least 12 in the currently installed I/O modules, with more planned in the future).

Detailed proposal
This proposal calls for I/O modules to optionally implement an additional order call to be used by the io_call command to perform these control operations. How this order will work is explained in the documentation to be added to the SWG (See page 3, Support for the io_call control command).
Documentation changes

Since the format of the io_call command control operation will differ according to the I/O module and the specific order call, it would not be appropriate to include all the different command formats in the io_call command description. Therefore, each I/O module should include a section entitled "Implementation of the control operation of the io_call command", or something similar. The explanation of the control operation in the io_call command should be changed to direct the reader to this section of the I/O module description. Finally, an explanation of the io_call order should be added to the SWG under "Writing an I/O module".

--- MPM SUBROUTINES ---

The following is a sample showing how the tty_ I/O module order calls would be performed. Similar sections should be added to the other I/O module descriptions.

Implementation of the control operation of the io_call command

All control orders can be performed using the io_call command. The general format is:

\[ \text{io_call control switchname order -optional_arg-} \]

where:

1. order is any of the control operations previously listed.
2. optional_arg is required only for the control operations shown below.

\[ \text{io_call control switchname store_id new_id} \]

where new_id is the 4-character string to be stored as the terminal answer-back.

\[ \text{io_call control switchname set_type new_type} \]

where new_type is a decimal number representing the type number of the terminal.

--- MPM COMMANDS ---

The following should replace the section of io_call describing the control operation.

Operation: control

\[ \text{io_call control switchname order -optional_arg-} \]

where:
1. order is the control order to be performed.

2. optional_args are additional arguments required by some I/O modules for certain control orders.

This command applies only when the I/O switch is attached via an I/O module that supports the control I/O operation. The exact format of the command depends on which I/O module is being used, and which control order is being performed. Consult the I/O module description in the MPM Subroutines for more details.

--- SWG ---

The following should be added to the section on "Writing an I/O module".

Support for the io call control command

To facilitate control operations from command level (via the io_call command), an I/O module that implements the control operation should also implement the "io_call" control order. Briefly, this order works as follows: whenever the io_call command must perform a control operation, it will call iox$_control to perform the "io_call" order with a pointer to the structure described below. It is the job of the I/O module to interpret the structure and execute the real order call that was to be performed.

The format of the info structure on the io_call order is as follows (This structure will be contained in io_call_info.Incl.pl1):

```plaintext
dcl 1 io_call_info aligned,
    2 caller_name char(32),
    2 order_name char(32),
    2 nargs fixed bin,
    2 max_arglen fixed bin,
    2 args (0 refer (io_call_info.nargs))
        char (0 refer (io_call_info.max_arglen)) varying;
```

where:

1. caller_name is the name of the caller to be used in com_err_ calls.
2. order_name is the name or the order from the callers command line.
3. nargs is the number of arguments in the arg array.
4. max_arglen is the length of the longest argument.
5. args are the arguments from the command line.
The following steps describe how an I/O module could implement the `io_call` order. Other ways, of course, are possible.

1. Check to see if the order is one that requires an info structure. If not, just return `error_table_$no_operation`. The `io_call` command will recall `iox_$control` with the original order and a null info pointer.

2. If an info structure is required, it should be constructed. This may involve the interpretation of the arg array.

3. Call `iox_$control` recursively, using the info structure just constructed.

4. Finally, if the structure contains output values, these may be printed, if this is useful. For large structures, it is conceivable that the arg array could contain control arguments to select what is being printed.

The procedure has two alternatives to choose from to report an error. First, it may simply return a non-zero status code to `io_call`, in which case the following standard error would be printed:

```
io_call: Text of error. switchname
```

If this is not sufficient, the subroutine can call `sub_err` directly, passing it the caller name from the info structure. In this case, a zero status code should be returned.
<table>
<thead>
<tr>
<th><strong>VER. 3</strong></th>
<th><strong>MULTICS CHANGE REQUEST</strong></th>
<th><strong>MCR_1713</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STATUS</strong></td>
<td><strong>DATE</strong></td>
<td><strong>Written</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>03/08/76</strong></td>
</tr>
<tr>
<td><strong>AUTHOR</strong></td>
<td><strong>VanVleck</strong></td>
<td><strong>ASSIGNED</strong></td>
</tr>
<tr>
<td><strong>AUTHOR</strong></td>
<td></td>
<td><strong>EXPires</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>09/08/76</strong></td>
</tr>
<tr>
<td><strong>TITLE</strong></td>
<td>Fix bug in quota $move_quota</td>
<td></td>
</tr>
<tr>
<td><strong>AUTHOR</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Planned for System</strong></th>
<th><strong>4.0</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fixes Bug Number(s)</strong></td>
<td>not applicable</td>
</tr>
<tr>
<td><strong>Documented in MT3</strong></td>
<td>not applicable</td>
</tr>
<tr>
<td><strong>Incompatible Changes</strong></td>
<td>no</td>
</tr>
<tr>
<td><strong>User/Operations-visible Interface Changes</strong></td>
<td>no</td>
</tr>
<tr>
<td><strong>Code (0) PL/I ( 9) ALM ( 0) other-see below</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>better ( ) same ( ) worse</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>MPM (vol,sect)</th>
<th>MPAM (sect)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDSN (sect)</td>
<td>MSAM (sect)</td>
</tr>
<tr>
<td>PLI (AN#)</td>
<td>anotl</td>
</tr>
<tr>
<td>Info Sys</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>

**OBJECTIONS/COMMENTS:**

**SUMMARY:**

Change quota $move_quota to return the error code $invalid_move_quota if the parent directory does not have a terminal quota.

**REASONS:**

The current code prints a nonfatal syserr message and proceeds; as long as the (uninitialized) quota in the directory happens to contain zero the system will produce an error code later after locking and unlocking the page table lock unnecessarily. This expense is needless and if the cell ever contained garbage quota could be generated illegally.

**IMPLICATIONS:**

Eliminates peculiar syserr message.
SUMMARY:

Change delentry to be able to delete a directory whose UID in the directory header is zero (say, due to a reused address).

REASONS:

Lock fails to lock such a directory. This causes delentry to fail, and the directory cannot be deleted.

DETAILED PROPOSAL:

If the directory UID is zero, treat it as if it were a segment and delete it anyway. Log a sysserr message giving the pathname.

IMPLICATIONS:

Users no longer have to wait for the salvager to run to get rid of bad ex-directories.
TITLE: Install a FAST/DFAST version of fortran io

AUTHOR: G. Chang

- Code in X PL/I ☐ AIM ☐ other-
  explain in DETAILED PROPOSAL ☐
- Planned for System MR 3.1 ☒
- Fixes Bug Number(s):
- Documented in MTB ☐
- User/Operations-visible ☐
- Interface change? ☐ yes ☐ no
- Incompatible change? ☐ yes ☐ no
- Performance: ☐ Better ☐ Same ☒ Worse
- Replaces MCR ☒

Objections/Comments:

SUMMARY:

To install a fortran io that supports FAST/DFAST fortran io.

IMPLICATIONS:

The fortran io to be installed will only be used by FAST/DFAST users. This is a temporary arrangement. Eventually, this fortran io will be merged with the current fortran io. Distinctions between FAST, DFAST, and Multics will be made by switches in fast_related_data.

DETAILED PROPOSAL:

The differences between this fortran io and the system version is mentioned in MCR-1589 (fortran for DFAST), and will not be repeated here. This version will also contain the segments fortran stop and fortran pause. Names will be renamed fast fortran io, fast fortran stop, and fast fortran pause temporarily for this MR3.1 release.
TITLE: COBOL Maintenance Release

AUTHOR: William K. O'Neill (CEO-B)

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

Summary
A. The compiler does not generate the proper code to handle unsigned data, that is described as signed. Similarly, signed data, that is described as unsigned is not handled properly. The University of Southwestern Louisiana has requested fixes for these problems.

B. Many other difficulties have been fixed. A copy of the bug file will be available from C. Zethraeus or from me.*

Implications
In addition to fixing difficulties the compiler has been modified to use a standard prefix on external procedure names, include file names and external data names. Thus all components of the compiler have changed.

* Bug list attached.
Bugs marked as being in (Cobol Release) 1.5 will be fixed.
The following is a list of outstanding bugs in Release 1.4.

**112 DATE**** uns** For relative or indexed files having dynamic access mode, the **REWRITE** verb causes the "current record pointer" to be set to the record just rewritten rather than leaving it undisturbed. Thus, a **READ NEXT RECORD** following a **REWRITE** causes the rewritten record to be accessed rather than the next record as determined by a previous **READ RECORD** (or **START**). The same problem will occur with **DELETE** or **WRITE**.

**120 DATE**** uns** Errors occur if the data-name specified with the **RELATIVE KEY** or **RECORD KEY** clause is qualified.

**131 DATE**** 1.5** An attempt to specify a data-item declared in the **CONSTANT SECTION** as the receiving field of the **MOVE** verb is not diagnosed. This will normally result in run-time failure. Also, inconsistent and misleading diagnostics are given for other verbs for which a **CONSTANT SECTION** item is specified as the receiving field. The diag S-248 (produced now for the **ACCEPT** verb) should be produced in all cases.

**155 DATE**** uns** The **SEARCH** verb causes an entire table to be searched, even when the table contains a variable occurrence item whose length-determining field contains a less than maximum value.

**157 DATE**** 1.5** The identifier used with the **CALL** statement is not allowed to be subscripted.

**159 DATE**** 1.5** More than one contiguous receiving operands of the **ADD**, **SUBTRACT**, **DIVIDE**, and **MULTIPLY** statements cannot be subscripted. The compiler will abort with an out of segment bounds in the PD syntax phase.

**161 DATE**** uns** No warning is given when the same IFN is used for more than one **EXTERNAL** file in a COBOL program.

**162 DATE**** uns** No diagnostic is issued when **TALLY** is illegally used as an operand of a statement.

**163 DATE****** During execution of the **OPEN** statement, it is possible that the attachment has been accomplished but that the actual opening fails. In this case if the user retries the **OPEN** and (a) the **CATALOGUE-NAME** is variable and (b) it is different than it was for the original **OPEN**, the user must detach the io_switch (via io_call) before retrying. NOTE: This will be specified
in the user's guide.

Internal files (i.e., files not with EXTERNAL specified in their SELECT clause) must not have ATTACH-OPTION "syn_ (user_input | user_output)". This will cause the user_input or user_output io_switch to be closed when the CLOSE statement is executed which is not conducive for the continued life of the process. NOTE: This will be specified in the user guide.

Lower case alphabetic literals (e.g., "a", "b", ...) are not allowed to be used in the CURRENCY SIGN IS literal clause. Use of the corresponding character in a picture description is not recognized and diagnosed as a fatal error. NOTE: Upper case must be used. This will be specified in the reference manual.

A program containing the statement INITIALIZE X where X contains an array, and also has a performable paragraph or section physically located AFTER this statement causes a severity 4 error in the generator:

mc_fixup: An inconsistency exists in the fixup table ... This can be avoided by placing all such INITIALIZE statements at the end of the program.

An attempt to SET a group index data name is not diagnosed as an error. Also, no code is generated.

The subject of REDEFINES must not be separated from the object by a condition name (i.e., level entry). This causes an erroneous fatal diagnostic indicating unequal size redefines. For example, the following will fail:

```
1 a.
2 b pic 99.
3 c value is 0.
2 d pic xx redefines b.
```

In a complex condition, the combination of words AND IS NOT is not accepted. For example:

if a = c and is not > c ... For identical functionality use:

if a = c and not > c ...

The factoring if the NOT logical operator in a complex condition causes compile failure in the fixup phase of generation. For example:

if not ( a = b or c = d ) ... For identical functionality use:

if not a = b or not c = d ...

A MOVE CORRESPONDING resulting in a null match causes a fatal diagnostic. ANSI is not clear on what action should be taken in such a case. However, it would seem more useful to issue only a warning and merely generate no moves. NOTE: This will be clarified in the reference manual.

```
I

An item subordinate to an item containing the OCCURS clause will not be a candidate for a MOVE CORRESPONDING match. ANSI states that only the item actually containing the OCCURS is not to be a candidate (i.e., not the items subordinate to it).

Any "corresponding" in which the referenced level is separated from the elementary level by 2 or more levels will not process elementary levels correctly after a mismatch is found on the level immediately above the elementary level. For example:

```
1 a.
2 b.
3 c.
4 d pic x.
3 e.
4 f pic x.
3 g.
4 h pic x.
```

"move corr a to aa" will correctly move d, not move f, and incorrectly not move h.

If the DEPENDING ON identifier-1 clause is used in an "sd" entry, the record size is not stored in identifier-1 when the sd file is read.

If DFCIMAL POINT IS COMMA is specified then OBJECT IS COMMA must be specified.

The COPY...REPLACING structure causes unspecified results, such as an abort during compilation (.EX phase).

The CLOSE...REEL clause incorrectly closes the file. NOTE: This will be clarified in the reference manual.

COPY of an include file which contains a continued statement is not parsed correctly.

ADVANCING clause in the WRITE statement sends improper linefeeds to the printer. a) AFTER ADVANCING 0 advances 1 line, b) BEFORE ADVANCING incorrectly advances line after which specified number of lines are advanced.

COPY...REPLACING does not replace a numeric paragraph name.

DELETE of records on indexed data file does not delete the records correctly. NOTE: Duplicate of difficulty number 21.

Read pointer is not incremented properly when a READ is done after a DELETE. Problem can be reduced by inserting an extra READ immediately following the DELETE. NOTE: Duplicate of difficulty number 21.
READ is executed after a REWRITE. Problem can be circumvented by inserting an extra read immediately following the REWRITE. NOTE: Duplicate of Difficulty number 21.

124 35314*** 1.5 Compiler aborts in clist. The ddalloc phase calls clist to put out diagnostic 150 (unequal redefines) and gives a bad token size. The temporary solution is to fix the error in the source program so the diag is not issued.

125 35316*** 1.5 If the procedure division contains more than about 1500 source lines the compiler will abort in the replacement phase with an out of segment bounds in mc_io_gets_put.

127 35317*** 1.5 The compiler aborts in mc_addr due to a perform varying, when the item varied is subscripted. We do not know if all such subscripts will fail, or only more complex ones.

128 35318*** 1.5 Extraneous diag (possible left truncation) in the file section. The appear to be related to a move corresponding.

129 35294*** uns The message "cross reference listing not produced - source program contains too many data-names" may appear. If so the compilation will be correct, except for a cross reference listing.

130 35322*** uns The compiler will abort if the ENVIRONMENT DIVISION header is not present.

131 35326*** 1.5 A SIGN CLAUSE should not be used on a group item, or unspecified results will occur at object time.

132 35344*** 1.5 It is not possible to list source lines when using debug with COBOL object programs. Debug will ignore the request.

133 35293*** 1.5 A STRING statement where the pointer variable is overpunch sign will generate bad code (results are unpredictable) at execution time. At compile time the message "mc_register$ get unable to get pointer register" will occur, compilation will continue.

134 26958*** uns An UNSTRING of an alphanumeric into a numeric may produce invalid code if the length of the numeric is less than the length of the alphanumeric.

135 27004*** uns A READ into a file that has an occurs may store the data in the wrong location. NOTE: This is patched at USL in data division allocation.

136 27016*** uns A WRITE AFTER ADVANCING PAGE, followed by a WRITE BEFORE ADVANCING n LINES does not overprint. The second line is appended to the first.

137 26983*** uns MOVE of a scaled integer item to an alphanumeric item fails. Example: move a to b; a pic 99PP value 3000; b pic XXXX; gives 300b; should be 3000.
114 27046*** uns AOV; of a scaled integer item to an alphanumeric edited item fails. Example: move a to b; a pic 9999; b pic XXXX; gives 307b2 should be 307b0.

114 27016*** uns Data which is described with an overpunch sign (S in picture), but is actually unsigned may produce incorrect results at execution time. Similarly data described as unsigned but which has an overpunch sign may produce incorrect results.

114 27045*** uns MOVE of a COMP-7 data item to a COMP-6 data item will fail at execution time.

114 27016*** uns The key of an indexed file must be within the record. If not, a fatal diagnostic (rather than an observation) will occur.

114 27001*** 1.5 A compiler abort will occur if the number of data definitions, plus paragraph definitions, plus procedure definitions exceeds about 800. The abort is an out of bounds at mc_io_$swf_put in the replacement phase. This has been patched in the compiler driver at USL.

115 27022*** 1.5 MULTIPLY anything by data-name(x) may abort in mc_addr at compile time. Works correctly if the occurs is on the data-name. The failure will be either "attempt to reference thru null pointer" or "subscript or index error encountered". One temporary solution is to put a "giving" in the statement.

115 27013*** 1.5 A format divide (no giving) with subscripting may abort in mc_addr. See difficulty 151 for details.

115 27007*** 1.5 MULTIPLY a by b, c, d will abort in mc_addr at compile time. Similarly DIVIDE a into b, c, d. The temporary solution is to remove the multiple receiving fields.

115 27008*** 1.5 If a SORT is executed and no records have been released, a loop will occur at object time.

115 26284*** 1.5 In an edited move, if the receiving item has DB or CR in the picture clause and the sending item is zero, two zeros are generated, instead of two spaces.

116 27014*** 1.5 A move corresponding will not move some of the data items if the items are out of order.

117 36442*** 1.5 The INITIALIZE statement should not be used. It's use will result in either compiler aborts or erroneous diagnostics.

113 27015*** 1.5 An ambiguous data name in an occurs depending on will not cause a diagnostic message to be issued. But the count of fatal diagnostics will include this.

119 27017*** 1.5 Warning diagnostic 3-181 (second value not greater than first) will be issued on 88 cond-1 value "a" thru "e". This diagnostic is extraneous and should be ignored.
The compiler will abort in the data division allocation phase if a reserved word is used as the variable in an occurs depending on.

An add or subtract corresponding will produce a fatal diagnostic if the data items correspond, but are not numeric.

If bad syntax is encountered in an organization clause in the environment division, the compiler will abort with "illegal machine operation" in the id/e/synta phase.

A "go to" from a declarative to a non declarative, or to another declarative will result in a fatal diagnostic (rather than an observation).

A level 88 associated with a group item will cause a "storage condition" in the generator phase of the compiler.

(END)
SUMMARY: Install procedures to implement Multics HEALS.
Draft documentation is attached.

REASON: HEALS is a new facility not previously installed.
See:

1. Multics HEALS II Product Functional Specification,
2. Multics HEALS II Project Summary and Authorization,
   PSA 225AD
3. Product Calendar Item 2C0102
4. Multics HEALS II, Phase 2,
   MTH-265

IMPLICATIONS: HEALS reports on CPU errors and tape I/O errors will be available to site support and FED personnel.

DETAILED PROPOSAL: Install the following procedures:

- heals_report.pl1
- heals_io_reports.pl1
- heals_cpu_reports.pl1
- heals_collect_data.pl1
- heals_hran.pl1
- heals_arg_info.incl.pl1
SUMMARY OF HEALS FACILITIES

This section provides a brief description of the HEALS facilities on the Multics system.

HEALS is an acronym for "Honeywell Error Analysis and Logging System". It provides facilities for:

1. capturing and logging hardware error data,
2. sorting and analyzing the data,
3. presenting the analyzed data in a series of reports.

The purpose of HEALS is to assist central site personnel in monitoring performance of the hardware and predicting maintenance schedules. It is also useful in diagnosing transient malfunctions. HEALS facilities, therefore, are not of interest to the average user of Multics.

Multics HEALS reports are a subset of GCOS HEALS reports. The set includes those of most interest to Honeywell Field Engineering Division personnel and excludes those peculiar to GCOS systems or of interest primarily to GCOS users. The list of Multics HEALS reports is:

- I/O Error Report
- Tape Reel Error Statistics Report
- Tape Errors by Handler and Command Report
- Tape Errors by Reel-number/Unit Report
- Tape Errors by Unit/Reel-number Report
- Tape Unit Error Variance Report
- Disk Error Statistics Report
- CPU Error Report
- MOS Memory EDAC Report
- APC Statistics Report

In release MKS.1, only the CPU Error Report and the I/O Error Report for tapes are supported.

HEALS IMPLEMENTATION

Hardware error data for HEALS is logged by the syserr mechanism to the syserr log. Logging of the data is independent of other HEALS facilities.

The HEALS facilities are invoked by the command "heals_report". Arguments to this command specify the desired reports and the from/to time of the report data (see below).
The heals_report command causes the ring 4 copy of syserr_log to be updated.

SUMMARY DESCRIPTION OF HEALS REPORTS

A detailed description of the reports is contained in the manual "HEALS II Manual", Field Engineering Division, May 1975. Although this manual describes the GCOS HEALS reports, the Multics HEALS reports have the same format and content with minor exceptions (eg, the reports do not report by GCOS job and activity number). A brief summary of each report follows.

The first seven reports are concerned with I/O errors. The I/O Error report lists all I/O errors that were logged, while the other six reports select and sort the same data for convenience and emphasis. The three remaining reports deal with the processor, memory, and peripheral controller subsystems.

The I/O Error report summarizes all I/O error records found on the syserr_log for the specified time interval and details the data found on those records. It many times will be used as a final reference when more specific data is needed after first analyzing other HEALS reports. For each error, the report line contains the date/time the error was logged, the device name, the major and status of the status return words, the type of interrupt, the device command, the device address, the binary seek address for disk entries or the tape reel serial number for tape entries, the number of records for a multiple record command and the record residue, and the number of connects on this device from the time of the last system boot load.

The Tape Reel Error Statistics report lists the reel numbers of the first 512 tapes reporting errors for this reporting period. They are sorted by descending order of the total number of data alerts logged against those reel numbers. This report will indicate tape reels that may need maintenance by the tape librarian.

The Tape Errors by Handler and Command report tallies all tape errors by handler device number and tape subsystem command. This report will indicate which tape device may need additional diagnosis.

The Tape Errors by Reel-number/Unit report will indicate whether a tape reel is failing on multiple devices. This will assist in the determination of media versus device problems.

The Tape Errors by Unit/Reel-number will indicate device errors that are occurring when different tape reels are mounted on the same tape device.
The Tape Unit Error Variance report indicates which device is experiencing the most data alerts with respect to connects for the entire tape subsystem.

The Disk Error Statistics report summarizes the records for system mass storage errors that have occurred during the reporting period. The continuous binary seek address is converted to its device specific decimal equivalent in order to relate the failure to a specific physical characteristic of the device. All read, write, or seek errors will be reported.

The CPU Error report formats and prints history register dumps.

The MOS Memory EDAC report summarizes the MOS and Core storage error or error correction information.

The MPC Statistics report displays the statistical counters for Tape and Disk MPC subsystems. The display represents valuable statistics including accurate counts of device usage and certain abnormal conditions. Statistics of particular interest are counts of marginal conditions and errors successfully recovered by the firmware. Each channel and device address is displayed on this report.

HEALS USAGE

It is expected that HEALS will be used for both routine reporting of hardware errors and for specific reports on demand.

It is recommended that all HEALS reports be generated on a daily basis (or at most a weekly basis) to maintain a continuous record of hardware errors and malfunctions. This HEALS activity can be a scheduled absentee job.

At any time that specific reports are wanted for monitoring or diagnostic purposes, HEALS can be invoked by terminal command.

The ring 4 copy of syserr_log should save at least the data subsequent to the previous run of all HEALS reports. Preferably it should also save the data for the previous full report run.

HEALS INSTALLATION REQUIREMENTS

The HEALS procedures have no special requirements.

The user of HEALS must have "re" access on /system_control/audit_gate_ and must have "r" access to /system_control/perm_syserr_log in order to have the ring 4 copy of syserr_log updated.
The heals_report command produces specified reports of interest to site support and Field Engineering personnel. The reports are placed in the current working directory for subsequent perusal or printing. A report file is given a name formed from the report_name argument (see below) by appending a suffix of _report.

Usage

heals_report report_name -control_args-

where:

1. report_name is selected from the following:
   io_error selects the I/O Error Report,
   cpu_error selects the CPU Error Report,

2. control_args are selected from the following:
   -from date specifies the date and time after which errors will be reported. If this argument is not given, the default value will be the value of -to time minus 24 hours.
   -to date specifies the date and time to which errors will be reported. If this argument is not given, the default value will be the current date and time.

Notes

The dates specified after the -from, -to, and -control_args must be acceptable to the convert_date_to_binary_subroutine.

The various reports are described in detail in the section "Multics HEALS" above.
If the command line

    heals_report io_error -from 03/01/76 -to 03/02/76

is issued at 2:00 PM, a file named io_error_report and suitable
for printing will be created in the current working directory,
containing the I/O Error Report for the period from 2:00 PM,
March 1, 1976 to 2:00 PM, March 2, 1976.
TITLE: Fix 28.0 NSS bugs, Round 1

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)
- MOSN (sect)
- PLMs (AN#)
- Info Segs
- Other

OBJECTIONS/COMMENTS:

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

SUMMARY: Fix bugs in New Storage System affecting segment moving and handling of MSU451 disks.

REASONS: Current software causes shutdown failure and system crashes in certain cases.

IMPLICATIONS: Better reliability.

DETAILED PROPOSAL: Change fsout_vol to correctly output volume maps larger than 1K. Change segment_mover to correctly rethread the ASTE of a segment being moved.
**TITLE:** Change BOS Firmware Loader to Print Revision Number of Firmware  

**AUTHOR:** Noel I. Morris

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Document</th>
<th>Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>355</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOS</td>
<td>MPM (Vol, Sect.)</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>PLMS (AN #)</td>
<td></td>
</tr>
<tr>
<td>Salvager</td>
<td>MSGN (Sect.)</td>
<td></td>
</tr>
<tr>
<td>Ring Zero</td>
<td>MPAM (Sect.)</td>
<td></td>
</tr>
<tr>
<td>Ring One</td>
<td>MSAM (Sect.)</td>
<td></td>
</tr>
<tr>
<td>SysDaemon/Admin.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Runtime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Cmm/Subr.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Objections/Comments:**

- Info Segs
- Other (Name)
- None (Reason)

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

**Proposal:**

Modify the BOS FWLOAD command to print the revision number of the firmware module being loaded. This will enable FED and operations to know what versions of firmware are in use at any given time. The message

```
LOADED M500
```

will be changed to

```
LOADED M500 REV.LA
```
**TITLE:** Fix Bug in Append  
**AUTHOR:** A. Kobziar  

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>TVV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
</tr>
</tbody>
</table>

- **Coded in:** [ ] PL/1 [ ] AIM [ ] other-
- **Planned for System MR 4.0:** other-
- **Fixes Bug Number(s):** unreported
- **Documented in MTB:** 355

**User/Operations-visible:** [ ] yes [x] no
**Interface change?** [ ] yes [x] no
**Performance:** [ ] Better [x] Same
**Worse** [ ]
**Replaces MCR** [ ]

**DOCUMENTATION CHANGES**
- **Documented in:** MCR 1355

**STATUS DATE**
- **Written:** 3/12/76
- **Status:** A
- **Expires:** 9/23/76

**IMPLICATIONS:**
- The reload's running out of kst will no longer leave unconnected vtoces.

**SUMMARY:**
In creating a directory, append gets a vtoce for the new branch. If append is unable to initialize the new directory's header (makeknown fails because kst full), append deletes the branch but doesn't free the vtoce. A call to delete vtoce will be added to deal with such failures.

**Objections/Comments:**
- Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.
TITLE: Redefine Recovery Mode Directory Salvaging
AUTHOR: A. Kobziar

SUMMARY:
Define two recovery modes of directory salvaging rpv and rlv.

REASONS:
The automatic salvaging of the hierarchy is a necessary component in making crash recovery foolproof (and unattended).

IMPLICATIONS:
A shutdown failure on the root physical pack (rpv) will automatically invoke a rpv salvage, which should take about 3 min. at MIT, while a shutdown failure on any pack belonging to the root logical volume (rlv) will automatically invoke an rlv salvage, which should take about 10 min at MIT.

DETAILED PROPOSAL:
An rpv salvage will salvage all directories up to level 2 that are hierarchically accessible on the rpv, and check rpv vtoce connections (i.e. >udd>m will be salvaged). An rlv salvage will salvage all directories up to level 3 (i.e. >udd>m>amk will be salvaged) but will not check any vtoce connections. (level 3 reaches about ½ of all MIT directories.)
**Title:** Fix bug restricting absentee arg lengths  

**Author:** Steve Herbst

<table>
<thead>
<tr>
<th>Code Location</th>
<th>Category (Check One)</th>
<th>Document Specified One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL/I</td>
<td>Lib. Maint. Tools</td>
<td>Specified One or More</td>
</tr>
<tr>
<td>Other</td>
<td>Sys. Anal. Tools</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sys. Prog. Tools</td>
<td></td>
</tr>
</tbody>
</table>

**Status:** A 3/23/76

**Expires:** 3/23/76

**Date Written:** 3/15/76

---

### DOCUMENTATION CHANGES

- **Fixes Bug Number(s):**
  - Documented in MCR 355

- **User/Operations-visible Interface change:**
  - Yes

- **Performance:**
  - Better

- **Replaces MCR:**
  - MPAM (Sect.)
  - MSAM (Sect.)

---

**Objections/Comments:**

- **Info Segs**
- **Other (Name) doc ok**
- **None (Reason)**

---

**SUMMARY:**

Remove 168 character restriction on lengths of absentee arguments.
TITLE: Fix bugs in BOS device address interpretation
AUTHOR: Noel I. Morris

Objections/Comments:

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

Summary:
Several bugs have been found in the BOS utility subroutines cvadd and argmul. In certain cases, these bugs cause device addresses to be interpreted incorrectly. Problems have been found when using the dump command and when using the test command to test or clear the bulk store.

Proposal:
Analyze and fix the bugs.
**Title:** Fix Bug in BOS RESTOR Command  

**Author:** Noel I. Morris

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOCUMENTATION CHANGES</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MPM (Vol, Sect.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLMS (AN #)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOSN (Sect.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSAM (Sect.)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document Specified One or More</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BOS</td>
<td>355</td>
<td></td>
</tr>
<tr>
<td>Salvager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ring Zero</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ring One</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SysDaemon/Admn.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Runtime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Cmd/Subr.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

**Summary:**
A coding bug in the RESTOR command causes it to print incorrect information about tape contents and target disk under certain conditions.

**Proposal:**
Fix the bug.
TITLE: Fix Bugs in BOS DUMP Command

AUTHOR: Noel I. Morris

Written: 3/12/76

Status: 3/23/76

Expires: 3/23/76

Doc: DETAILED PROPOSAL

Lib. Maint. Tools

Category (Check One)

Sys. Anal. Tools

Sys. Prog. Tools

355

Document Specift One or More

MPM (Vol, Sect.)

BOS

MOSN (Sect.)

PLMS (AN #)

SysDaemon/Admin.

Ring Zero

User Cmd/Subr.

Ring One

MSAM (Sect.)

SysDaemon/Admin.

Worse

Better

Same

Replaces MCR

Objections/Comments:

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

Summary:

Several problems have been found in the BOS DUMP command:

1. Attempting to dump 256K segments causes an infinite loop.
2. Certain dumping functions do not initialize the online printer package.
3. If the DBR has the unpaged bit turned on, it appears as the character "J" in the REGS printout.

Proposal:

Fix the problems.
TITLE: Fix bug in the BOS IF command

AUTHOR: Noel L. Morris

- Code in: [ ] PL/I [ ] ALM [ ] other
- Explain in DETAILED PROPOSAL: Lib. Maint. Tools
- Planned for System MR: 4.0
- Fixes Bug Number(s): Sys. Anal. Tools
- Documented in MTB: Sys. Prog. Tools
- User/Operations-visible: Document
- Category (Check One): Specify One or More

Interface change? [ ] yes [ ] no
- Incompatible change? [ ] yes [ ] no
- Performance: [ ] Better [ ] Same
- Replaces MCR: [ ] Worse

Objections/Comments:

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

Summary:
A bug in the SHUT test of the IF command causes it to get an error.

Proposal:
Fix the bug.
TITLE: Move the "flagbox" segment

AUTHOR: Noel I. Morris

Ver. 4

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

Objections/Comments:

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

Summary:

Multics currently sets up a flagbox segment during system initialization. This segment is used to contain various data needed by BOS in order to access Multics main memory when dumping, etc. It also contains a word of 36 flag bits which can be used for communications between Multics and BOS for smooth unattended operation.

Unfortunately, the flagbox segment is placed in memory such that it is overlayed by BOS main control when BOS is entered. This requires the flag bits to be copied into the BOS toehold when BOS is entered from Multics. But, these bits will also be copied when BOS is entered from BOS --- causing them to be destroyed.

It is obvious that the flagbox segment should be part of the BOS toehold, itself. Then, no copying of data would be required. Such a mechanism was considered at one time, but it was discarded since write permission would have to be given to the BOS toehold in Multics. This was (rightly) thought to be dangerous.

Detailed Proposal:

Move the flagbox segment so that it resides in the second 16 word block of the BOS toehold. Make it an unpaged segment only 16 words long. Modify the BOS toehold to move all other data used by BOS out of the second 16 words of the toehold.

Modify the BOS appending package to pick up the SST SDW and hardcore DBR out of the toehold instead of from the saved Multics main memory image. Make the code compatible for a while so that this information will be looked for in both places.
| TITLE: | Remove general_format_parse from bound_fortran |
| STATUS | DATE |
| Written | 3/18/1976 |
| STATUS | Expiry |
| A | 2/22/76 |
| Document | Specify One or More |

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

**SUMMARY:**

The procedure general_format_parse must be available to both Fortran compilers and both Fortran I/O packages. All four bound segments are, or will be, installed in unbundled. If the binding arrangements are not changed, a prelinked version of dfast/fast will have to include bound_fortran.

**REASONS:**

One of the designed goals for dfast/fast is that it have low resource usage. bound_fortran would add a 51K segment, in order to use a 1K object segment.

**IMPLICATIONS:**

The old compiler would have to snap a link to general_format_parse instead of having the link satisfied by the binder. The old Fortran I/O package will continue to snap a link.

**DETAILED PROPOSAL:**

Make general_format_parse a standalone segment.
## Ver. 3
741022 MULTICS CHANGE REQUEST

### TITLE:
Recompile several hardcore programs

### AUTHOR:
VanVleck

### STATUS DATE
Written: 03/18/76
Expires: 09/18/76

### SUMMARY:
Recompile 40 programs on the system tape which were last compiled by the non-EIS version of PL/I. Minor changes are necessary to a few of the programs to make them compile. One program, timer_manager, requires a minor bug fix.

### REASONS:
This change reduces the number of compilers which needs to be shipped with each system release.

### IMPLICATIONS:
none

---

**Planned for System:** 4.0

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

**Incompatible Changes:** no

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

**Coded In:** (2) PL/I ( ) ALM ( ) other-see below

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

---

### DOCUMENTATION CHANGES (specify one or more)

<table>
<thead>
<tr>
<th>MPM (vol,sect)</th>
<th>MPAM (sect)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSN (sect)</td>
<td>MSAM (sect)</td>
</tr>
<tr>
<td>PLMs (AN#)</td>
<td>an61</td>
</tr>
<tr>
<td>Info Segs</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>

---

### OBJECTIONS/COMMENTS:

---

### HEADINGS AREA

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

---

---

---

---

---

Page 1 of 1
SUMMARY:

Change terminate_proc to call verify_lock unconditionally instead of only calling if pds$block_lock_count is nonzero.

REASONS:

The new version of lock does not record locks in pds$block_lock_count for "fast" locks. A process can lock such a lock and terminate, and hang the system.

IMPLICATIONS:

Fix bug.
TITLE: Fix Card Reader Hangup Problem in BOS

AUTHOR: Noel I. Morris

Summary:
When BOS detects an I/O error when reading cards, it attempts to wait for the card reader to become ready again. In certain cases, the special interrupt indicating that the reader is ready will be lost, and BOS will appear to hang.

Proposal:
Fix the card reading and special interrupt detecting mechanism in BOS.
## MULTICS CHANGE REQUEST

**MCR 1734**

<table>
<thead>
<tr>
<th>TITLE:</th>
<th>Fix set_tty for network problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTHOR:</td>
<td>Robert S. Coren</td>
</tr>
<tr>
<td>Written:</td>
<td>03/16/76</td>
</tr>
<tr>
<td>Status:</td>
<td></td>
</tr>
<tr>
<td>Expires:</td>
<td>09/16/76</td>
</tr>
<tr>
<td>CATEGORY (check one):</td>
<td></td>
</tr>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
</tr>
<tr>
<td>355</td>
<td></td>
</tr>
<tr>
<td>BOS</td>
<td></td>
</tr>
<tr>
<td>Salvager</td>
<td></td>
</tr>
<tr>
<td>Ring Zero</td>
<td></td>
</tr>
<tr>
<td>Ring One</td>
<td></td>
</tr>
<tr>
<td>SysDaemon/Admin</td>
<td></td>
</tr>
<tr>
<td>Runtime</td>
<td></td>
</tr>
<tr>
<td>User Command/Subr</td>
<td></td>
</tr>
<tr>
<td>Deferred changes</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td></td>
</tr>
<tr>
<td>OBJECTIONS/COMMENTS:</td>
<td></td>
</tr>
</tbody>
</table>

**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 | 
| Other | 

| None (reason) | no change required |

**SUMMARY:** Make set_tty command recognize network line type, and not perform certain functions for network lines.

**REASONS:** Default modes for certain terminal types are invalid for the Network.

**IMPLICATIONS:** No unexpected error messages for set_tty used over the network; inability to set tabs over the network using set_tty.

**DETAILED PROPOSAL:** If set_tty identifies the line type of the caller as being "network" (i.e., it is not any line type known to the tty_DIM), it will not set default modes when the "-terminal_type" control argument is given, nor will it perform the functions implied by the "-reset" and -tabs" control arguments.
**TITLE:** Accept options (variable) and options (non_quick) on procedure statements

**AUTHOR:** R. Barnes

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>LIB. MAINT. TOOLS</th>
<th>System Analysis Tools</th>
<th>System Program Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

- Expires [ ] 03/30/76
- Status [X] 03/24/76

**Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.**

**SUMMARY:** Change the PL/I compiler to accept options (variable) and options (non_quick) on procedure statements.

**PROPOSAL:**

1. Allowing a user to put options (variable) on an internal procedure statement, will allow him/her to call an internal procedure with a variable number of arguments just like ioa_. Since cu_arg_ptr and friends will have to be used in accessing the parameters of such a procedure, the compiler will make it non-quick.

2. Options (non_quick) allows a user to specify that a particular block must be non_quick. Options (no_quick_blocks) actually does that today, but mistakenly implies that all contained blocks are also non_quick.
TITLE: Reorganize info segment directories

AUTHOR: John Gintell

Summary:

Create a separate directory for the info segments corresponding to each "logical" library. Reorganize the info segments accordingly.

- >doc> info contains standard system
- >doc> tools contains tools
- >doc> unbundled contains unbundled software
- >doc> ils is installation maintained

change check_info_segs and help to search >doc> info, doc> unbundled and > doc> ils in that order.

doc> tools will not be searched since it does not contain info segments of interest to ordinary users.

Reasons:

The contents of the unbundled library should be synchronized with its info segments.
Info_segments for tools should not be on the "search path" of ordinary users.
TITLE: Replace all info segments in \texttt{doc\>info} to conform to new into segment conventions (new help command) and remove obsolete segments. Info segments fall in four categories:

1) commands in AG02 and AK 92
2) Some subroutines in AG 93
3) Changes in the latest release
4) General topics

Reasons:

The new help command expects a new format of info segment. The directory \texttt{doc\>info} had many obsolete info segments.

Implications:

Procedures to keep these info segments up-to-date must be established.
DETAILED PROPOSAL:

Phase 1: Remove obsolete info segments. Install info segments for all commands in AG92. (This was done for MR 3.1)

Phase 2. For 4.0 complete the rest of 1) and 4), update 3), do 2).
TITLE: Emergency fixes for modules used by FAST/DFAST

AUTHOR: M. Weaver

RAB

Objections/Comments:

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

SUMMARY:

1) Retain the new basic compiler entries in bound_ep_basic_bind,

2) Correct basic object segment generation in FAST/DFAST.

3) Add two more segdefs to fast_related_data.

REASONS: Changes needed for fortran and basic to be usable in FAST/DFAST.

DETAILED PROPOSAL:

The basic compiler sets the segname definition incorrectly. If it is entered at one of the new FAST entry points, often causing faults. The new segdefs in fast_related_data are:

1) dcl fast_related_data $terminate_run entry ext variable; /*entry to call to terminate run_unit*/ Used by fortran stop procedure; set by FAST and DFAST run unit managers to appropriate internal procedures.

2) dcl fast_related_data $fortran_io_initiated bit (1) aligned ext; if the value is "1", fortran I/O has been initialized for the run unit and set to "0" by each invocation of the run unit manager before program execution.
Fix bugs in write_user_usage_report

**STATUS**

Written: March 17, 1976

**DATE**

**EXPIRES**

09/30/76

**SUMMARY:**

Fix the following bugs:

1) Billing blows up on a project with more than 256 users;

2) in the message:
   "usage for the period from date to to date"
   from date is the later of the last time billing was run or
   the last time a new user was registered on the project;

3) in the message warning of inconsistent mischarge figures
   between profile and miscfile, the figures are reversed (each
   figure is identified as coming from the other file).

**REASONS:**

1) Projects with more than 256 users exist;

2) Incorrect messages confuse recipients of bills.

**IMPLICATIONS:** Fix for 1) decreases performance by eliminating quick blocks.

**DETAILED PROPOSAL:**

1) Declare arrays inside begin blocks, after number of projects in
   the sat and number of users on project are known; instead of
   using fixed length arrays of arbitrary size.

2) Change date-finding algorithm to use the earlier of the two
dates: this chooses a date later than the last billing date
   only for a project newly-created during the month.

3) Reverse the figures, leaving the text of the error message
   unchanged.
**TITLE:** Add &q and &r to exec_com  

**AUTHOR:** Steve Herbst  

<table>
<thead>
<tr>
<th>Objections/Comments:</th>
</tr>
</thead>
</table>

**Use these headings:**  

**Summary:** Add &q, &r, &qf and &rf features to exec_com. They work the same as in the do command.

**Reason:**

Do and exec_com implement essentially the same language with respect to parameter substitution.

**Implication:**

Incompatible change. These newly defined strings currently represent themselves.
**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 &0 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.
The strings &qi, &ri, &fi, &qfi, and &rfi also indicate argument substitution. The string &qi is replaced by the ith argument to the exec_com command with quotes doubled. The string &ri is replaced by the ith argument, requoted. Refer to the do command for a description of quote doubling and requoting. The string &fi is replaced by a string of the ith through last arguments to exec_com, separated by blanks. Likewise, &qfi is replaced by a string of the ith through last arguments with quotes doubled and &rfi is replaced by a string of the ith through last arguments, requoted.

The string &n is replaced by the number of arguments to the exec_com command. The string &ec_name is replaced by the entryname portion of the exec_com pathname without the ec suffix. The string &O is replaced by the pathname argument to exec_com just as it was typed.
**TITLE:** Fix bug in hcs_$initiate_search_rules

**AUTHOR:** Steve Herbst

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependent:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Status:**
- Planned for System MR
- Fixes Bug Number(s)
- Documented in MTP
- User/Operations-visible
- Interface change? [X] yes [ ] no
- Incompatible change? [X] yes [ ] no
- Performance: [X] Better [ ] Same [ ] Worse
- Replaces MCR

**Objections/Comments:**
- [ ] Doc ok
- [ ] Info Segs
- [ ] Other (Name)
- [X] None (Reason) doc ok

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

**Summary:** Fix bug in hcs_$initiate_search_rules preventing the root from being added to the search rules.

**Reason:** The root is a valid place to search for object segments.
**TITLE**: Fix bug in `add_name`

**AUTHOR**: Steve Herbst

---

**Category (Check One)**

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

**Status**

- Written: 3/23/76
- Expires: 02/30/76

**Objections/Comments:**

- Multics Change Request

---

**DATE**

- 3/23/76

---

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

**Summary:** Fix bug causing the `add_name` command to go into a loop when a pathname argument has bad syntax.
TITLE: Convert several hardcore data bases to be cds segments.

AUTHOR: Steve Webber

Planned for System: MR 4.1
Fixes Bug Number(s): not applicable
Documented in MRB: 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)

APM (vol, sect) MPAM (sect)
MDSN (sect) MSAM (sect)
PLMs (A#) several
Info Segs
Other

OBJECTIONS/COMMENTS:

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

SUMMARY:

Convert the following hardcore data bases to be cds segments, i.e., the source changes from alm to cds.
1. ods
2. nds
3. scs
4. sys_info
5. tc_data
6. oc_data
7. syserr_data

REASONS:

The cds language is much better suited for generating data bases than alm. Also, by creating include files, which the cds source can reference, it is much easier to generate metering and debugging tools in the user ring since we can easily know the offset of particular items in hardcore data bases.

IMPLICATIONS:

It may become tempting to use the all include file representation in hardcore code rather than the external reference method invoking the linkage mechanism. If this becomes extensive, more and more of the system will have to be recompiled when a data base changes in such a way that offsets of items change. The all include file method is almost always more efficient.
### SUMMARY

Add the procedure "get_main" to the system. It is to be called during initialization (usually during collection I initialization, but not restricted to it) to get contiguous main storage for a segment. It allows for the creation of segments such as "sst" and "tc_data" as a function of useful parameters rather than having to calculate how many pages of core are required.

### REASONS:

1. Makes for a better system administrator interface since commands such as gen_sst_card and get_tcd_card are no longer needed.

2. Makes better use of wired down core by minimizing breakage and allowing the system to have table sizes which are not arbitrarily a multiple of 1K.

3. Eases the high water mark.

### IMPLICATIONS:

Several programs have to be converted and configuration cards changed. MOSN must be updated.

### DETAILED PROPOSAL:

The get_main procedure will be called with a pointer to a segment for which storage is desired. If the segment is wired, as determined by the SLT entry for the segment, get_main acquires storage at the "perm-wired" end of the
collection I main-memory "window" (see AN70). If the segment is not wired, the storage is taken from the "naged-segs" end. get_main returns an SDW for the segment, that is generated from the address (as it determines), from the bound (as input), and with access set to read, write.

If get_main is called after init_sst has run (during collection I), the "window" no longer exists, so get_main calls oc_contiq to acquire contiguous (wired-down) core. In this case, the segment will use up integral pages from the paging pool.