TO: Distribution
FROM: M. S. Hodges
DATE: September 5, 1974
SUBJECT: Multics Change Requests

Enclosed are copies of Multics Change Requests which were approved from August 16 to August 31, 1974.
TITLE: Modifications to NCP to standardize access control and handle process termination conditions.

AUTHOR: D. Wells

SOURCE: (if external) e.g., "User", "Marketing"

CLASSIFICATION | JUSTIFICATION | Replaced by proposal MCR
--- | --- | ---
Incompatible Change | Marketing Requirement | Replaced by proposal MCR
Extension | X Conformance to Standard | Objections/Comments:
Restriction | Increased Consistency | Needs further design review.
Performance Improvement | Simplification | 
Reliability Improvement | Generalization | 
X Improvement | Bug Fix | 

REASONS: The Network Control Program does not currently properly close out connections when a process terminates.

The Network Control Program is not properly turned off when the Network Daemon process terminates.

Access to the "connect" function is determined in a non-standard manner -- i.e., a segment in the hierarchy is checked to see if the user had access to that segment. This requires that the Ring 0 NCP know about at least one pathname in the online hierarchy.

SUMMARY:

The Network Control Program will be modified to use the I/O Assignment Manager to properly close and deactivate any connections that a user had when his process terminated.

The Network Control Program will be modified to use the I/O Assignment Manager to shut down the Network when the Network Daemon process terminates.

Access to the "connect" function will be determined by a gate to be added. Access to this gate will be dynamically modified via a privileged call to the NCP. The Network Daemon process overseer will be modified to initialize this access when it initializes the Network Control Program.

IMPLICATIONS:

The number of IOAT entries will have to be increased to support this function. These entries are wired down (though the NCP use of them does not intrinsically require that they be wired down) -- thus decreasing the size of the paging pool. Because the size of the IOAT is to be determined by a configuration card, this can be managed by system maintenance personnel.
Fix Bugs Reading HALF/FULL switch in BOS and Multics

Noel I. Morris and Bernard S. Greenberg

Internal

Increased Consistency

REASONS:
Improper documentation has led to the use of the wrong RSW instruction in determining the setting of HALF/FULL switches for memory on the CPU configuration panel. This bug exists both in the BOE subroutine, getportinfo, and in the Multics initialization primitive, scas_init.

Summary:
Correct the code to use the proper RSW instruction.
### TITLE:
Make a branch for configuration deck in `system_library_1`

### AUTHOR:
Noel I. Morris

### SOURCE:
(if external) e.g., "User", "Marketing"

In ternal

<table>
<thead>
<tr>
<th>CLASSIFICATION</th>
<th>JUSTIFICATION</th>
<th>REPLACED BY PROPOSAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible</td>
<td>Change</td>
<td>Implemented in System</td>
</tr>
<tr>
<td>Extension</td>
<td>Performance to Standard</td>
<td></td>
</tr>
<tr>
<td>Restriction</td>
<td>Increased Consistency</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Simplification</td>
<td></td>
</tr>
<tr>
<td>Improvement</td>
<td>Generalization</td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td>Bug Fix</td>
</tr>
<tr>
<td>Improvement</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**REASONS:**

Many outer ring programs need access to the Multics configuration deck (e.g. on-line TandD). If a branch is created for the config deck in `system_library_1`, then the segment "config_deck" can be initiated and read without having to call into ring 0 to copy it out. This mechanism has already been used for the SLT segment and its name table.

**Summary:**

Change the SLT template include files to cause a branch to be created for "config_deck".

**Objections/Comments:**

Correct ACL should be placed in header so users don't have access to this.
MULTIICS CHANGE REQUEST

TITLE: Make a segment for the BOS toehold

AUTHOR: Noel L. Morris

SOURCE: (if external) e.g., "User", "Marketing"

Internal

CLASSIFICATION

<table>
<thead>
<tr>
<th>Incompatible</th>
<th>Prerequisite</th>
<th>Replaced by proposal MCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Standard</td>
<td>Implemented in System</td>
</tr>
<tr>
<td>Restriction</td>
<td>Performance</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td>Simplification</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Generalization</td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>Bus Fix</td>
<td></td>
</tr>
</tbody>
</table>

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

REASONS:
Multics initialization primitives currently "know" that the 1024 word block of core which contains the BOS toehold should not be assigned for paging. Programs which return to BOS also "know" the absolute location of the toehold. These wired in values can be removed from all but one program if a 1 page segment containing the BOS toehold is constructed during system initialization by bootstrap1.

Summary:
Modify bootstrap1 to construct the segment "bos_toehold" when it sets up the initial descriptor segment. Change the SLT template include files to include this new segment.

IMPLICATIONS:
Unless segments are shuffled slightly, the segment numbers of emergency_shutdown and the sst_sep will change.
TITLE: vfile_bugs

AUTHOR: M.D. MacLaren

SOURCE: (if external) e.g., "User", "Marketing"

CLASSIFICATION | JUSTIFICATION | Replaced by Proposal MCP
-----------------|--------------|--------------------------
Infocompatible    | Marketing     | Implemented in System
                  | Chores       |                          |
Extension         | Performance   |                         |
                  | Improvements  |                         |
Restriction       | Increased     |                         |
                  | Consistency   |                         |
Performance       | Simplification|                         |
Improvements      |               |                         |
Reliability       | Generalization|                         |
Irreprovement     | unreported    |                         |
                  | Bug Fix       |                         |

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

REASONS: Recent installation of vfile has compiler-caused
*bug that prevents use of sequential files. Testing
uncovered an unrelated bug in the program that results
in an error when attempt is made to backspace beyond
beginning of a segment.

SUMMARY: Replace open_seg_file in bound_vfile with a corrected
version compiled using the experimental compiler (in which
compiler bug is fixed).

IMPLICATIONS: Sequential files will have correct format (instead
of slightly incorrect format). Old files will work as well
as they do now.

NOTE: It is not feasible to change the program to get around the
compiler bug. The only alternatives to using the exl
compiler are to reinstall the old open_seg_file (complete
with bug) or compile using the old non_EIS compiler).

* Compiler bug #1203
MULTICS CHANGE REQUEST

<table>
<thead>
<tr>
<th>TITLE</th>
<th>STATUS</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make changes to Answering Service to Implement Access Isolation Mechanism</td>
<td>Written</td>
<td>10/12/74</td>
</tr>
<tr>
<td>Paul A. Green</td>
<td>Approved</td>
<td>8/30/74</td>
</tr>
<tr>
<td>SOURCE</td>
<td>Postponed</td>
<td></td>
</tr>
<tr>
<td>(if external; e.g., &quot;User&quot;, &quot;Marketing&quot;)</td>
<td>Withdrawn</td>
<td></td>
</tr>
<tr>
<td>CLASSIFICATION</td>
<td>Expires</td>
<td>2/20/75</td>
</tr>
<tr>
<td>JUSTIFICATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incompatible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marketing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement Implemented in System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extension</td>
<td>Objections/Comments:</td>
<td></td>
</tr>
<tr>
<td>to Standard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conformance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Increased</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restriction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improvement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simplification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improvement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generalization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bug Fix</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Use these headings: REASONS, SUMMARY, IMPLICATIONS, and optionally DETAILED PROPOSAL

REASONS
See MTB-078.

SUMMARY
Modify answering service include files and programs, and administrative tools, as outlined in MTB-078 (as modified by MTB-100 terminology) to provide AIM attributes for persons, projects, terminals, etc.

IMPLICATIONS
Covered by MTB-078 and a previous design review. Adds new features and complexity to the answering service.
Memo to: Multics Change Review Board
From: Paul Green
Re: Summary of Changes to Answering Service for AIM

The proposed changes will:
. record an authorization attribute (bit (72) aligned) for each entry
  in the PNT (per person), SAT (per project), PDT (per user), and CCT
  (per terminal; new table, called the Channel Control Table).
. provide tools to update and maintain these fields, under the control
  of the System Security Administrator.
. compute the access authorization of a process based on the attributes
  of the person, project, user, terminal, and login options.
. print the computed authorization of the process at login time (if it
  is greater than unclassified).
. provide options to the login and new_proc commands to specify the
  requested authorization, provide a login option to set the default
  authorization.
. provide the ability (on a per-person basis, in the PNT) to have the
  system assign passwords when the -change_password login option is
  given.
. be documented in the style of the MPM/SPS/or PLM, as appropriate.

The individual programs and include files which will be changed are
rather numerous, and so are not specified in the MCR or this note.

The above requirements are from Honeywell's proposal to the Air
Force; as time goes by we may add more changes to the above list.
TITLE: Add name tdcn_ to hcs_

AUTHOR: Bill Silver

SOURCE: (if external) e.g., "User", "Marketing"

CLASSIFICATION

<table>
<thead>
<tr>
<th>Incorrigible Change</th>
<th>Justification</th>
<th>Replaced by proposal ICP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Marketing</td>
<td>Implemented in System</td>
</tr>
<tr>
<td></td>
<td>Performance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X: Extension</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Performance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Restriction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improvement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reliability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improvement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X Generalization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bug Fix</td>
<td></td>
</tr>
</tbody>
</table>

REASONS: Adding the name tdcn_ to the gate hcs_ is a first step in the plan to remove all hcs_ entries which reference tdcn_. This is being done for reasons of security and to allow tdcn_ to be removed from ring zero once ioi_ is installed.

SUMMARY: The name tdcn_ will be added to the gate hcs_. All calls in the tape I/O modules tape_ and nstd_ which are of the form hcs_$tdcm_xxxx must be changed to calls of the form tdcn_$tdcm_xxxx.

IMPLICATIONS: When all system maintained and user programs have been changed to call tdcn_ rather than hcs_, the entry points in hcs_ which reference tdcn_ can be deleted. At this time a higher ring tdcn_ or a new gate tdcn_ will have to be installed.
**TITLE:** Deletion of set file protect on tape detach

**AUTHOR:** Bill Silver

**SOURCE:** (if external) e.g., "User", "Marketing"

---

**CLASSIFICATION** | **JUSTIFICATION** | **Replaced by proposal 'CP'**
---|---|---
Incompatible Changes | Marketing Requirements | Implemented in System
Extension | Conformance to Standard | Objections/Comments:
Restriction | Increased Consistency |
Performance Improvement | Simplification |
Reliability Improvement | Generalization |
X Improvement | unreported |
X Bug Fix |

**REASONS:** The tool currently issues a set file protect command whenever it rewinds and unloads a tape. This forces the operators to remove the tape reel from the handler in order to turn on write permit. This is an unnecessary inconvenience. Handler malfunctions sometimes compound the problem.

**SUMMARY:** Stop tdcv from issuing the set file protect command.

**IMPLICATIONS:** This operational problem will be solved.

**DETAILED PROPOSAL:** The set file protect command set up in tape_data will be changed to a reset status command.
Fix BOS to run with Reader on any channel

Fix BOS to run with Reader on any channel

N. I. Morris

Internal

Incompatible

Incompatibility

Incompatibility

Extension

Restriction

Performance

Reliability

Improve

Improvement

Increased

Consistency

Increased

Consistency

Simplification

Generalization

Bug Fix

BOS currently can only run with a card reader configured to IOM channel 16. With the increased use of URMPC's, this becomes a serious restriction.

Summary:
The card reader channel number is deposited in memory when a card is read via the IOM bootload mechanism. A simple change to the BOSLDR 5-card loader can enable this channel number to be preserved. The BOS loader program, LOADDM, can then pick up this channel number and save it in BOS common. This saved channel number can be used by BOS whenever cards are to be read.

Change the CARDS command to accept an optional argument which is a card reader channel number. This will enable changing the card reader channel during BOS operation.

Use these headings: REASONS, SUMMARY, IMPLICATIONS, DETAILED PROPOSAL (Optional)
MULTICS CHANGE REQUEST

**TITLE:** Prepare library programs for unbounded library

**AUTHOR:** Arlene J. Scherer

**SOURCE:** (e.g., "J scr", "Marketing")

**CLASSIFICATION:** 

**JUSTIFICATION:** Replace by proposal MCR

**Reasons:**

**PART ONE:** Add the unbounded library to all programs which look at the system libraries. Programs involved are:
- get_library_segment and related control tables
- submission_test
- update_key (subroutine upa_doc_task)
- the uninstalled Multics Library Descriptor (library_print).

**PART TWO:** Part Two modifies the request expressed in Part Four of MCR 812 to add a "-sys" option to submission test calls for the purpose of allowing a Subsystem or other non-standard library to be used for comparison of source and binaries. Upon further thought, a "-control_key" ("-ck") argument will better serve this need by allowing the user to specify a control_key in which his own version of the key control segments may be found, enabling him to use any arbitrary set of paths.

**SUMMARY:**

These changes need to be implemented ahead of any actual unbounded library installations. In addition, users and staff should be notified (perhaps via an ATU) of the intended establishment of the unbounded library, and what its planned contents are.
TITLE: Install compare_object

AUTHOR: Richard Barnes

SOURCE: (if external) e.g., "User", "Marketing"

CLASSIFICATION | JUSTIFICATION | Replaced by proposal "TCP"
---|---|---
Incompatible Change | Marketing Requirement | Implemented in System
Extension | Performance to Standard | Objections/Comments:
Restriction | Increased Consistency | Install in Tools
Performance Improvement | Simplification | 
Reliability Improvement | Generalization | 

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

REASONS: compare_object enables a user to determine if two object segments, possibly compiled by different versions of a compiler, are essentially identical. This enables installation personnel to ensure that a submitted source segment produced a submitted object segment with a current compiler.

SUMMARY: compare_object (cob) compares the text, link, and definition sections of two object segments. The user can also specify that certain sections should not be compared. Another entry point, compare_object_, is provided so that this comparison can be achieved from a program. compare_object_ is a function returning bit (1) aligned.
TITLE:  Implement new matching strategy in acl_commands

AUTHOR: Steve Herbst

SOURCE: (if external) e.g., "User", "Marketing"

CLASSIFICATION | JUSTIFICATION | Replaced by proposal MCP
--- | --- | ---
Incompatible | Marketing | Implemented in System
Change | Requirement |
Extension | Performance to Standard |
Restriction | Increased Consistency |
Performance Improvement | Simplification |
Reliability Improvement | Generalization |
Bug Fix |

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

REASONS: A redefinition of the matching strategy for missing components in acl names was proposed in MTB-092. The new extended_access_commands uses this new matching strategy. The regular acl commands should use it too.

SUMMARY: Change acl_commands to call find_common_acl_names.

IMPLICATIONS: Change in user interface. Users must be warned in advance.
## TITLE:
Bind together debug, trace_stack, and trace

## AUTHOR:
Barry L. Wolman

## SOURCE:
(if external) e.g., "User", "Marketing"

### CLASSIFICATION | JUSTIFICATION | Replaced by proposal "CP"
---|---|---
Incompatible Change | Marketing Requirement | Implemented in System

### Summary Changes:
- **x** Extension
- Restriction
- **x** Performance Improvement
- Reliability Improvement
- **x** Generalization
- Bug Fix

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

## REASONS:
The three debugging programs debug, trace_stack, and trace now have a number of common modules; the number of common modules will be increased if debug and trace_stack are changed to use the new argument list printing program developed for trace. This argument list printing program is an extension of the program currently used by trace_stack; it was specifically written to be usable by all three debugging programs.

## SUMMARY:
Change debug and trace_stack to call print_arg_list_, the new program mentioned above, and combine the three debuggers into a single bound segment. This work has already been done in a private version.

## IMPLICATIONS:
The output produced by trace_stack and by the .a request of debug will be slightly different. debug will be able to print the value of an argument of any PL/I scalar type. The documentation of debug will have to be updated.
**TITLE:** New ring 1 gate macros.

**AUTHOR:** P.A. Janson

**SOURCE:** (if external) e.g., "User", "Marketing"

---

**CLASSIFICATION** | **JUSTIFICATION** | **Replaced by proposal "CP"**  
---|---|---  
Incompatible Changes | Marketing Requirement | Implemented in System  
Extension | Conference to Standard | Objections/Corrections:  
Restriction | Increased Consistency |  
Performance Improver | Simplification |  
Reliability | Generalization |  
X Security Improver | Bug Fix |  

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

**REASONS:** Ring 1 gates, according to the current version of gate_macros.incl.mexp, set their linkage pointer (lp) by picking it up from the LOT of the ring they were invoked from. Firstly, this is a security violation: ring 1 gates should not trust the content of a LOT entry in an outer ring. Secondly, it is inconsistent with what hard­core gates do and any user ring gate should do.

**SUMMARY:** This MCR proposes to change a few lines of code in gate_macros.incl.mexp and to recompile ring 1 gates (deh_test_gate, installation_tools, message_segment, tape_admin) so as to have them pick their lp from the ring 1 LOT entry.

**IMPLICATIONS:** The change is fully compatible with what the current linker does. It also sets the LOT entry in ring 1 when snapping a link to a ring 1 gate. In addition the change will make ring 1 gates compatible with the design of the future user ring linker which will no longer maintain the LOT entry in the outer rings for the ring 1 gates.
**REASONS:** The following bugs exist in the installed basic:

No. 32: source program longer than 1000 lines causes fault in basic_lex
No. 33: bad code is generated for \((1 + a(i))^{\wedge} 2\)
No. 34: return without previous gosub sometimes causes fault
No. 35: math functions not setting indicators properly; this causes problems with statements such as
   
   \[
   \text{if } \sin(x) = 0 \text{ then } \cdots
   \]

No. 36: an array name appearing in more than one dim-statement is not diagnosed
No. 37: use of \(\times\) operator with fractional values can cause process termination
No. 38: instruction after instruction that causes underflow fault gets skipped due to change in behavior of FIM
No. 39: underscore not allowed in subprogram name
No. 40: illegal subprogram name in sub-statement causes fault if the sub-statement is the first statement in the program
No. 41: \(\text{char(*)} \) arg not accepted in call of basic subprogram from PL/I program
No. 42: subprogram name not properly supplied in error messages relating to incorrect call of subprogram
No. 43: on num goto \(\cdots\) not properly split into tokens
No. 44: store of control variable in for-next loop occurs after test is made instead of before as specified by manual.
No. 45: for \(i = 0\) to \(t\) step \(\cdots\) not properly split into tokens by EXL basic due to error in fix for bug 43
No. 46: references to global variables are not correct if a function is called from another function that was called from another subprogram.

**SUMMARY:** Fix the bugs.
REASONS:

1. The current version of basic uses fixed size tables for recording line numbers, data elements, and constants. These fixed size tables place a limit on the size of the program that can be compiled, and some relatively small programs can exceed a table size. Using very large sizes for these tables causes unnecessary page faults when compiling small programs.

2. In current version of basic, about 20% of compiler's time is spent in the procedure basic_, which does little more than loop calling the external procedures basic_lex and basic_statement. Making basic_lex and basic_statement be quick internal procedures of basic_ yields a 15% increase in speed of the compiler.

SUMMARY:

1. Change basic to use a storage strategy for its tables in which the tables are initially small and are located in the stack. When a table becomes full, it is copied into a pre-determined place in an external segment; if necessary, tables occurring after it in this segment are moved. Tables can continue to grow until the entire segment becomes full. This should allow programs of very large size to be compiled while enabling small programs to be compiled more efficiently.

2. Combine basic_, basic_lex, basic_statement, basic_object, basic_subprogram, and basic_error into a single program basic_. Storage formerly placed in an external segment for communication between these programs will be located in stack frame of basic_. The time required to compile the combined version is 50% greater.
than the time required to compile basic_statement, which was the module that was usually re-compiled to fix bugs.

These changes have been checked out and are currently installed in the EXL version of basic_. The anticipated reduction in page faults and increase in speed have been verified. Compiling a real 1350 line program, the compiler speed went from 21,000 lines per minute to 24,000 lines per minute; a 10,000 line "toy" program was compiled in 20 seconds, which places an upper bound of 30,000 lines per minute on speed of compiler in present form.

IMPLICATIONS: The external interface of basic_ remains unchanged, so there should be no noticeable change to any program that calls basic_.

Error Checking by indent command

T. Casey

STATUS:
Written 8-19-74
Approved 8/27/74
Selected
Postponed
Withdrawn
Expires 3/2/75

CLASSIFICATION
Incompatibility

JUSTIFICATION
Marketing

X Extension

Performance to Standard

Restriction

Increased

Performance Improvement

Simplification

Reliability Improvement

X Generalization

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

REASONS: One of the errors that indent detects is unbalanced quotes ("Program ends in a string"). Unfortunately, from the point of the missing quote to the end of the program, the indentation is performed on the contents of what should have been quoted strings changing the white space in them in some way. These changes are often inconvenient to undo. The only way to protect against this currently is to always have the indented source put in a new segment, but this also involves some inconvenience, added to every use of indent (rename, delete, etc.) to obtain a single copy of the source after verifying that it is a good copy.

SUMMARY: Change indent so that, when the original segment is to be replaced, and errors have been detected, the input segment will not be replaced. Instead, a message will be printed on error output, informing the user of this, and giving the pathname of the temporary segment containing the indented copy of the program.

IMPLICATIONS: This is an incompatible change from the present method of operation. However, since the most likely response of a user to errors during indentation is to fix the errors and indent again, this change should not cause any inconvenience.
# Implement status "names" option

**Author:** Steve Herbst  
**Source:** (if external) e.g., "User", "Marketing"

<table>
<thead>
<tr>
<th>Classification</th>
<th>Justification</th>
<th>Replaced by proposal &quot;CP&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible Change</td>
<td>Marketing Requirement</td>
<td>Implemented in System</td>
</tr>
<tr>
<td>Extension</td>
<td>Performance to Standard</td>
<td></td>
</tr>
<tr>
<td>Restriction</td>
<td>Increased Consistency</td>
<td></td>
</tr>
<tr>
<td>Performance Improvement</td>
<td>Simplification</td>
<td></td>
</tr>
<tr>
<td>Reliability Improvement</td>
<td>Generalization</td>
<td></td>
</tr>
<tr>
<td>Bug Fix</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

**Reasons:** The status command should accept "-names" as well as 
"-name" to print an entry's multiple names.

**Summary:** Recognize "-names" as a pseudonym for 
"-name".
**Title:** Install pli_operators, any_to_any, and dec_ops

**Author:** R. A. Barnes

**Source:** (if external) e.g., "User", "Marketing"

<table>
<thead>
<tr>
<th>Classification</th>
<th>Justification</th>
<th>Replaced by Proposal &quot;CP&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible Change</td>
<td>Marketing Requirement</td>
<td>Implemented in System</td>
</tr>
<tr>
<td>Extension</td>
<td>Conformance to Standard</td>
<td></td>
</tr>
<tr>
<td>Restriction</td>
<td>Increased Consistency</td>
<td></td>
</tr>
<tr>
<td>Performance Improvement</td>
<td>Simplification</td>
<td></td>
</tr>
<tr>
<td>Reliability Improvement</td>
<td>Generalization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bus Fix</td>
<td></td>
</tr>
</tbody>
</table>

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

**Reasons:**
To fix PL/I bugs: 1088, 1099, 1112, 1126, 1129, 1133, 1150, 1170, 1171, 1189, 1192, 1201, and 1202.

**Summary:** Major fixes include proper signalling of conversion when converting from character string, return (bit(*)) which used to return all zeroes, and freeing of controlled storage.

**Implications:** None, except that this installation must precede next PL/I installation.
TITLE: Add a wait for rewind completion to the detach entry for the nstd_tape_dim

AUTHOR: Susan Barr

SOURCE: (if external) e.g., "User", "Marketing"

CLASSIFICATION | JUSTIFICATION | Replaced by proposal MCP
--- | --- | ---
Incorporatable Change | Marketing Requirement | Implemented in System
Extension | Performance to Standard | Objections/Comments:
Restriction X Consistency
Performance Improvement | Clarification
Reliability Improvement | Generalization
Bug Fix

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

REASONS: If the user of the nstd_dim issues a rewind order and then tries to detach the tape, the error code device busy is returned.

DETAILED PROPOSAL: Add code to the detach entry to check if the tape is rewinding, and if so, to go blocked until completion of the rewind, and then to send the unload order.

A similar check is already made for data transfer requests and order calls.
Fix bug in mseg_access

Steve Herbst

Written 8-20-74
Approved 8/27/74

Increased Restriction Consistency
Ensures that all arguments are copied and referenced only once.

mseg_access $delete copies the input ACL to a condensed ACL structure with which to call has $delete acl entries. Per-entry error codes are placed in this condensed ACL structure. They are never copied out to the caller's ACL structure.

Copy back the per-entry error codes before returning.
### TITLE:
Change mechanism of calling initial procedure in a new process.

### AUTHOR:
Steve Webber

### SOURCE:
(if external) e.g., "User", "Marketing"

### CLASSIFICATION | JUSTIFICATION | Replaced by proposal MCP
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible Code</td>
<td>Marketing Requirement</td>
<td>Implemented in System</td>
</tr>
<tr>
<td>Extension</td>
<td>Performance to Standard</td>
<td>Objections/Comments:</td>
</tr>
<tr>
<td>X Restriction X</td>
<td>Increased Consistency</td>
<td></td>
</tr>
<tr>
<td>X Improvement</td>
<td>Simplification</td>
<td></td>
</tr>
<tr>
<td>X Quality Improvement</td>
<td>Generalization</td>
<td></td>
</tr>
<tr>
<td>X Bug Fix</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

### REASONS:
Some deficiencies exist in the current "init admin" "process overseer" mechanisms. In particular, the "direct" feature is not fully operational (the feature that allows a user to replace init admin rather than process overseer). Also, it is convenient for users to have a working directory established early in their process.

### SUMMARY:
see attached

### IMPLICATIONS:
Users will be able (if PDT, etc. say ok) to replace init admin and therefore perform their own I/O attachments. Knowledge of how to do this is transient and users so doing it should be warned.
SUMMARY:

Today the answering service will accept a process overseer from a user logging in and pass it on to the newly created process (if allowed to by the P111') in the pit. In addition, the answering service will pass on a request to have this program called directly rather than through "init_admin". (The mechanism for "direct" calling is not fully operational.)

The disadvantages with the current scheme are:

1) The "direct" feature is not fully operational,

2) The different kinds of processes (absentee, daemon, user, etc.) would probably best be served with separate "init_admin" programs, and

3) Due to a coding oversight, the init_admin_stack frame is not abandoned as it once was.

It is proposed that the above mechanism be replaced by the following:

1) init_proc (the initial program in a new process) which runs in ring 0 and sets up the KST via a call to initialize_kst and determines the init_admin_to use should be changed to
   a) get a pointer to the PIT
   b) determine from the PIT the process type and call either daemon_init_admin, absentee_init_admin, or user_init_admin unless a direct process overseer has been specified in which case it is called.

2) The program init_admin should be (logically) split into the 3 programs mentioned above.

3) The program real_init_admin should be split into the appropriate parts.

4) The replacements to init_admin should release their stack frame before calling the process overseer or at least have a small stack frame.

The reason for having user_init_admin and process_overseer two distinct programs is so that users can generate their own process overseers without being told about the internal interfaces in user_init_admin. Any direct process overseer, therefore, must have knowledge of internal interfaces which, by definition, are subject to change.