

TO: Distribution
FROM: Joan Archer
DATE: 24 July 75
RE: Multics Change Requests

Enclosed are copies of Multics Change Requests which were approved from 01 July through 15 July 75.

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

TITLE: Fix bug causing trap-before-links to fail.		STATUS	DATE	
AUTHOR: S. Webber		Written	6/25/75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) <u>unreported</u> -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status	
		Lib. Maint. Tools	Expires	
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
		355	MPM (Vol, Sect.)	
		BOS	PLMS (AN #)	
		Salvager	MOSN (Sect.)	
		<input checked="" type="checkbox"/> Ring Zero	MPAM (Sect.)	
		Ring One	MSAM (Sect.)	
		SysDaemon/Admin.	Info Segs	
	Runtime	Other (Name)		
	User Cmnd/Subr.	None (Reason) Bug Fix		
Objections/Comments:				

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

Summary: Change trap_caller_caller_ to correctly modify the (linkage fault) machine conditions before returning them to the user (for later restarting).

Reasons: Bug fix. Old PL/I and Fortran programs with trap-before-links failed to work.

Multics Change Request

TITLE: <u>Fix bug in expand_path_</u>		STATUS	DATE	
AUTHOR: <u>Steve Webber</u>		Written	<u>06.25.75</u>	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status	
	<input type="checkbox"/> Lib. Maint. Tools	<input type="checkbox"/> Sys. Anal. Tools	Expires	
	<input type="checkbox"/> Sys. Prog. Tools	<input type="checkbox"/> 355	DOCUMENTATION CHANGES	
	<input type="checkbox"/> BOS	<input type="checkbox"/> Salvager	Document	Specify One or More
	<input type="checkbox"/> Ring Zero	<input type="checkbox"/> Ring One	MPM (Vol, Sect.)	
<input type="checkbox"/> SysDaemon/Admin.	<input type="checkbox"/> Runtime	PLMS (AN #)		
<input checked="" type="checkbox"/> User Cmnd/Subr.		MOSN (Sect.)		
		MPAM (Sect.)		
		MSAM (Sect.)		
Objections/Comments:		Info Segs		
		Other (Name)		
		None (Reason)	<u>Bug fix</u>	

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

SUMMARY:

Fix bug in expand_path_ which looks at an argument before it verifies the argument is valid.

REASON:

Bug fix: causes random faults depending on stack history.

TITLE: Fix Bug in bootstrap1		STATUS	DATE	
AUTHOR: Noel I. Morris		Written	75.06.28	
-Coded in: <input type="checkbox"/> PL/I <input checked="" type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status A 07/01/75	
	Lib. Maint. Tools		Expires 01/01/75	
	Sys. Anal. Tools		DOCUMENTATION CHANGES	
	Sys. Prog. Tools		Document Specify One or More	
355		MPM (Vol, Sect.)		
BOS		PLMS (AN #)		
X Salvager		MOSN (Sect.)		
Ring Zero		MPAM (Sect.)		
Ring One		MSAM (Sect.)		
SysDaemon/Admin.		Info Segs		
Runtime		Other (Name)		
User Cmmnd/Subr.		None (Reason)		
Objections/Comments:				

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

Proposal:

Change the code in bootstrap1 used to enter BOS when certain switch settings are recognized. Currently, the code uses a canned-in SCU and TRA instruction. The code will be changed to use the instruction pair found at `<bos_toehold>|4`.

Reasons:

The BOS toehold was modified to use a RET instruction instead of a TRA instruction. Bootstrap1 should not be sensitive to the instructions used by the BOS toehold. It should use whatever instructions are placed there by BOS.

Implication:

Bootstrap1 will be able to go to BOS and return with the new BOS toehold mechanism as well as with the old.

TITLE: Delete debug code that supports version I p1		STATUS	DATE
AUTHOR: S. Barr		Written	6/20/75
		Status	A 07/01/75
		Expires	01/10/76
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL		Category (Check One)	
-Planned for System MR		<input type="checkbox"/> Lib. Maint. Tools	
-Fixes Bug Number(s) _____		<input type="checkbox"/> Sys. Anal. Tools	
-Documented in MPB _____		<input type="checkbox"/> Sys. Prog. Tools	
-User/Operations-visible		355	
Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		BOS	
-Incompatible change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no		Salvager	
-Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same		Ring Zero	
<input type="checkbox"/> Worse		Ring One	
-Replaces MCR _____		SysDaemon/Admin.	
		Runtime	
		X User Cmmd/Subr.	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		DOCUMENTATION CHANGES	
		Document Specify One or More	
		MPM (Vol, Sect.) III, Commands	
		PLMS (AN #)	
		MOSN (Sect.)	
		Info Segs	
		Other (Name)	
		None (Reason)	

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

Summary: Delete code in debug that supports obsolete versions of p1 and debug.

Reasons: Debug is used during the development and testing of programs. It is unlikely that users will have version I programs that they are debugging or even object segments with version I debug break maps. (Version II break maps were installed October 1974).

There will shortly be a new version object format (for prelinking) and a version III debug break map to fix a bug.

Proposal: Pre version I object segments and version I debug break maps will be recognized and an error message will be printed. In the case of version I p1 the program will have to be recompiled. For object segments with version I breaks, the breaks need to be reset which will cause the version I break map to be deleted.

The programs to be modified:

- db_break.p1
- db_break_map.p1
- db_sym.p1

Implications: Possible inconvenience for some users.

Multics Change Request

TITLE: Have debug support version 2 standard object segments		STATUS	DATE
AUTHOR: S. Barr		Written	6/25/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 07/01/75
		Expires	01/01/76
		DOCUMENTATION CHANGES	
		Category (Check One)	
		Lib. Maint. Tools	
		Sys. Anal. Tools	
		Sys. Prog. Tools	
		355	
		BOS	
		Salvager	Document Specify One or More
		Ring Zero	MPM (Vol, Sect.) III, Commands
		Ring One	PLMS (AN #)
		SysDaemon/Admin.	MOSN (Sect.)
		Runtime	MPAM (Sect.)
		<input checked="" type="checkbox"/> User Cmmd/Subr.	MSAM (Sect.)
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	

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

Proposal: Add code to debug to recognize version 2 standard object segments. Add a new segment ID "&i" which points to the internal static.

Modules to be changed:

- db_parse.pll
- db_assign.pll
- db_fill_snt.pll
- db_get_sym.pll
- db_line_no.pll
- db_print.pll
- db_sym.pll
- debug.pll
- db_snt.incl.pll

Implications: none

As debug decodes a data request, it parses the generalized address and generates a pointer to the data being referenced. This pointer, called the working pointer, is changed whenever the generalized address is changed. It points into either the working segment, its stack frame, or its linkage section. The actual segment depends on the most recent specification in a generalized address. The form for a generalized address is as follows:

```
[/segment name/] [offset] [segment ID] [relative offset]
```

(The brackets are not part of the debug syntax.) The segment name is either a pathname, a reference name, or a segment number, and defines what is called the working segment. The segment ID specifies which of the data bases associated with the working segment is to be used in setting the working pointer. The segment ID can be one of the following:

- &s refers to the stack frame if the working segment is a procedure segment with an active stack frame
- &l refers to an active linkage section (i.e., one with an entry in the Linkage Offset Table (LOT) for the user's ring)
- &t refers to the working segment itself
- &a refers to the source program for the working segment
- &p refers to the parameters of an active invocation of a procedure
- &i refers to an active internal static section (i.e., one with an entry in the Internal Static Offset Table (ISOT) for the user's ring)

The offset field is used as an offset within the segment referenced by the working pointer. For the working segment, this offset is relative to the base of the segment. If the working pointer points into an active stack frame, the offset is relative to the base of that frame. If the working pointer points into an active linkage section, the offset is relative to the beginning of that linkage section.

The offset can be either a number or a symbolic name. If a symbolic name is specified, a symbol table must exist for the working segment. See the pl1 command for a description of symbol table creation. If a symbolic name begins with a numeric character, the escape characters &n (for name) must precede the name, to avoid interpreting the name as a number. For example:

```
/test/&n10&t
```

might be used to specify the location associated with FORTRAN line number (i.e., label) 10 in a debug request.

- 12. b bit string
The data is printed as if it were a bit string. No more than 72 bit positions are printed in response to a single request.
- 13. g graphic
The specified number of characters are interpreted as graphic characters (this is assumed to start in typewriter mode).

Summary of Data and Control Requests

DATA REQUESTS

<u>/seg_name/</u>	<u>offset</u>	<u>seg ID</u>	<u>rel offset</u>	<u>operator</u>	<u>operands</u>
pathname	number	&t	number	,	operands
ref name	symbol	&s	register	=	input list
seg number		&l		<	function list
&n seg name		&an		>	
seg\$entry		&pn		:=	

<u>Segment ID</u>	<u>Operators</u>	<u>Registers</u>	<u>Output Modes</u>
&t text	, print	\$a	o octal
		\$q	h half-carriage octal
&s stack	= assign	\$aq	d decimal
		\$eaq	a ASCII
		\$x0	
&l linkage	< set break	.	i instruction
<i>&i internal static</i>		.	p pointer
&an source line	> transfer	\$x7	s source statement
		\$pr0	l code for line number
&pn parameter	:= call	.	n no output
		.	
		.	e floating point
		\$pr7	f floating point
		\$exp	b bit string
		\$tr	g graphic
		\$ralr	
		\$ppr	
		\$tpr	
		\$even	
		\$odd	
		\$ind	
		\$prs	
		\$regs	
		\$scu	
		\$all	

Ver. 3	MULTICS CHANGE REQUEST	MCR 1247
741022		
TITLE: Fix bug in convert_authorization_		STATUS DATE
AUTHOR: Paul Green		Written 06/20/75
		Status A07/01/75
		Expires 12/20/75
Planned for System: MR 3.0		
Fixes Bug Number(s): unreported		CATEGORY (check one)
Documented in MTB: not applicable		() Lib. Maint. Tools
Incompatible Changes: no		() Sys. Anal. Tools
User/Operations-visible Interface Change: yes		() Sys. Prog. Tools
Coded In: <input checked="" type="checkbox"/> PL/I () ALM () other-see below		() 355
Performance: () better <input checked="" type="checkbox"/> same () worse		() BOS
		() Salvager
<u>DOCUMENTATION CHANGES (specify one or more)</u>		() Ring Zero
MPM (vol,sect) MPM Subrs MPAM (sect)		() Ring One
MOSN (sect) MSAM (sect)		() SysDaemon/Admin
PLMs (AN#)		() Runtime
Info Segs		<input checked="" type="checkbox"/> User Command/Subr
Other		
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Change convert_authorization_ to strip leading blanks before checking for the keywords "system_low" and "system_high".

REASONS:

Greater consistency, as this is done now for level and category names.

IMPLICATIONS:

Greater consistency; fixes problem discovered by new card input software.

Multics Change Request

TITLE: Correcting access on hardcore gate linkage segments		STATUS	DATE
AUTHOR: A. Kobziar		Written	06.23.75
-Coded in: <input type="checkbox"/> PL/I <input type="checkbox"/> ALM <input checked="" type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.0 -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 07/01/75
		Expires	01/01/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason) invisible change	

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

SUMMARY:

Change the acIs of all gate linkage segments in the hardcore header to be read, write rather than read, execute, write. Add to each linkage segment a ringbracket 0,0,0 statement.

REASONS:

Currently, the missing ringbracket statement gives the linkage a ringbracket equal to that of the gate and thus can be called from ring 4. This is a security hole which the proposed change will correct.

TITLE: Directories for online documentation runoff source		STATUS	DATE	
AUTHOR: Bob May (Phoenix)		Written	06.16.75	
<input type="checkbox"/> PL/I <input type="checkbox"/> AIM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.0 -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status A 07/01/75	
	<input checked="" type="checkbox"/>	Lib. Maint. Tools	Expires 01/01/76	
	<input type="checkbox"/>	Sys. Anal. Tools	DOCUMENTATION CHANGES	
	<input type="checkbox"/>	Sys. Prog. Tools	Document	Specify One or More
<input type="checkbox"/>	355			
<input type="checkbox"/>	BOS	MPM (Vol, Sect.)		
<input type="checkbox"/>	Salvager	PLMS (AN #)		
<input type="checkbox"/>	Ring Zero	MOSN (Sect.)		
<input type="checkbox"/>	Ring One	MPAM (Sect.)		
<input type="checkbox"/>	SysDaemon/Admin.	MSAM (Sect.)		
<input type="checkbox"/>	Runtime	Info Segs		
<input type="checkbox"/>	User Cmmd/Subr.	Other (Name)		
Objections/Comments: NOTE: Runoff should not be used for info segments		None (Reason) internal for developers		

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

SUMMARY:

Some developers are using runoff to generate online documentation. It is desirable to keep the runoff segments in a system library for convenient access by all developers.

DETAILS:

Under >documentation, create the directory "documentation_source", with the added name "doc_source". Under doc_source, create the directories "info_segments" and "iml_info_segments", with their appropriate addnames. If other categories of info segments come into use, the corresponding source directories can be added easily. (It is assumed that the peruse text files will disappear when the new help command is installed.)

IMPLICATIONS:

This structure will keep the source directories separate from the info seg directories, so that software releases need not include them (suggested by Gary Dixon). Normally, only development sites would need this hierarchy.

TITLE: Install new KST/RNT		STATUS	DATE
AUTHOR: R. Bratt		Written	04.28.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) -Documented in MTB 154, 198 -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR	Category (Check One)	SHW	
	Lib. Maint. Tools	Status	P06/19/75 A 07/03/75
	Sys. Anal. Tools	Expires	01/08/76
	Sys. Prog. Tools	DOCUMENTATION CHANGES	
	355	Document	Specify One or More
	BOS	MPM (Vol, Sect.)	
	Salvager	PLMS (AN #)	61
	X Ring Zero	MOSN (Sect.)	
	Ring One	MPAM (Sect.)	
	SysDaemon/Admin.	MSAM (Sect.)	
Runtime	Info Segs		
User Cmnd/Subr.	Other (Name)		
	None (Reason)		
Objections/Comments:			

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

SUMMARY:

Move reference names from the KST to a new reference name table (RNT). This reference name table will subsequently be removed from ring zero. Implement a new module called ref_name_ to manage these names. Rewrite makeknown, makeunknown, and related modules for this change; to make room in the KST for per-process, per-segment access information; to prepare for "initiated mode"; to replace the current null-name facility by a per-ring count in the KSTE; and to prepare for the proposed new address space manager. No interface changes are included in this MCR except as noted.

IMPLICATIONS:

If a segment acquires more than 254 reference names it cannot be removed from a process' address space by repeated calls to terminate\$noname and terminate\$name. To remove such a segment, terminate\$seg or terminate\$file must be called. BOS dump cannot get reference names from KST (Bernie Greenberg has submitted a BOS which corrects this flaw by using the sst name table).

REASONS:

Simplify and speed up ring zero. Prepare for pre-linking system.

DESIGN:

The basic design is presented in MTB-154. This section describes only the proposed "null reference name" facility which was not contained in MTB-154. I intend to replace the concept of a "null reference name" with a usage count. When a program makes a segment known `bind_segno` increments a usage count in its KST entry. This count, which is kept on a per-ring basis, records the number of uses of this segment which are, in some sense, in progress. When the segment is made unknown by a program, `unbind_segno` decrements the appropriate ring's usage count. Thus, if each module which incorporates a segment into its address space also takes responsibility for making the segment unknown when it is finished with it, then this count truly represents the number of computations which are currently using the segment. `unbind_segno` respects this count by not physically removing a segment from a ring's address space unless its usage count is zero. This eliminates the need for null reference names.

Since core is a precious resource we have only allocated 9 bits for each ring's usage count. In addition, to avoid fixed of bit and bit of fixed annoyances, these usage counts are declared as fixed `bin(8)`. When a usage count reaches 255 `bind_segno` and `unbind_segno` will cease to increment and decrement it. Such segments may only be removed from the ring's address space by calling `unbind_segno` with a special "force" flag. This will force the usage count for the caller's ring to zero. For the reader's information a copy of the new KST include file is attached to this MCR.

```
1 1 /* BEGIN INCLUDE FILE - - - kst.incl.pl1 - - - last modified March 1975 by R. Bratt */
1 2
1 3 dcl pds$kstp ext ptr,
1 4 (kstp, kstep) ptr;
1 5
1 6 dcl 1 kst aligned based (kstp), /* KST header declaration */
1 7 2 lowseg fixed bin (17), /* lowest segment number described by kst */
1 8 2 highseg fixed bin (17), /* highest segment number described by kst */
1 9 2 highest_used_segno fixed bin (17), /* highest segment number yet used */
1 10 2 unused_1 (4) bit (36) aligned,
1 11 2 free_list bit (18) unaligned, /* relative pointer to first free kste */
1 12 2 unused_2 bit (18) unaligned,
1 13 2 uid_hash_bucket (0 : 127) bit (18) unaligned, /* hash buckets */
1 14 2 kst_entry (lowseg:highseg) aligned like kste; /* kst entries */
1 15
1 16 dcl 1 kste based (kstep) aligned, /* KST entry declaration */
1 17 2 fp bit (18) unaligned, /* forward ref pointer */
1 18 2 segno fixed bin (17) unaligned, /* segment number of this kste */
1 19 2 usage_count (0:7) fixed bin (8) unaligned, /* outstanding initiates/ring */
1 20 2 entryp ptr unaligned, /* branch pointer */
1 21 2 uid bit (36) aligned, /* unique identifier */
1 22 2 access_information unaligned,
1 23 3 dtbm bit (36), /* date time branch modified */
1 24 3 extended_access bit (36), /* extended access from the branch */
1 25 3 access bit (3), /* access from branch */
1 26 3 rb (3) bit (3), /* ring brackets from branch */
1 27 2 hdr fixed bin (3) unaligned, /* highest detectable ring */
1 28 2 flags unaligned,
1 29 3 dirsw bit (1), /* directory switch */
1 30 3 tus bit (1), /* transparent usage switch */
1 31 3 tms bit (1), /* transparent modification switch */
1 32 3 tpd bit (1), /* transparent paging device switch */
1 33 3 allow_write bit (1), /* set if initiated with write permission */
1 34 3 priv_init bit (1), /* privileged initiation */
1 35 3 audit bit (1), /* audit switch */
1 36 2 Infcount fixed bin (12) unaligned; /* inferior kste count */
```

hcs_\$initiate

hcs_\$initiate

Name: hcs_\$initiate

The hcs_\$initiate entry point, given a pathname and a reference name, causes the segment defined by the pathname to be made known and the given reference name initiated. If the reserved segment switch is on, then the segment pointer is input and the segment is made known with that segment number. In this case, the user supplies the initial segment number. If the reserved segment switch is off, a segment number is assigned and returned as a pointer.

Usage

```
declare hcs_$initiate entry (char(*), char(*), char(*), fixed bin(1),
                             fixed bin(2), ptr, fixed bin(35));

call hcs_$initiate (dir_name, entryname, ref_name, seg_sw, copy_ctl_sw,
                   seg_ptr, code);
```

where:

1. dir_name is the pathname of the containing directory. (Input)
2. entryname is the entryname of the segment. (Input)
3. ref_name is the reference name. If it is zero length, the segment is initiated with a null reference name. (Input)
4. seg_sw is the reserved segment switch. (Input)
0 if no segment number has been reserved
1 if a segment number was reserved
5. copy_ctl_sw is obsolete and should be 0. (Input)
6. seg_ptr is a pointer to the segment. (Input or Output)
Input if seg_sw is on (1)
Output if seg_sw is off (0)
7. code is a storage system status code. (Output)

Notes

The user must have nonnull access on the segment entryname in order to make it known.

If entryname cannot be made known, a null pointer is returned for seg_ptr and the returned value of code indicates the reason for failure. If entryname is already known to the user's process, code is returned as error_table_\$segknown and the seg_ptr argument contains a nonnull pointer to entryname. If ref_name has already been initiated in the current ring, the code is returned as error_table_\$namedup and the seg_ptr argument contains a valid pointer to the segment already initiated. If entryname is not already known, and no problems are encountered, seg_ptr contains a valid pointer and code is 0.

the status code error-table-\$usage-count-too-large is returned instead of error-table-\$segknown if the addition-53 of this reference name causes the AG93 count of initiated reference names for the segment to exceed a system defined ceiling.

hcs_\$initiate_count

hcs_\$initiate_count

Name: hcs_\$initiate_count

The hcs_\$initiate_count entry point, given a pathname and a reference name, causes the segment defined by the pathname to be made known and the given reference name initiated. A segment number is assigned and returned as a pointer and the bit count of the segment is returned.

Usage

```
declare hcs_$initiate_count entry (char(*), char(*), char(*),
    fixed bin(24), fixed bin(2), ptr, fixed bin(35));

call hcs_$initiate_count (dir_name, entryname, ref_name, bit_count,
    copy_ctl_sw, seg_ptr, code);
```

where:

1. dir_name is the pathname of the containing directory. (Input)
2. entryname is the entryname of the segment. (Input)
3. ref_name is the reference name. If it is zero length, the segment is initiated with a null reference name. (Input)
4. bit_count is the bit count of the segment. (Output)
5. copy_ctl_sw is obsolete and should be 0. (Input)
6. seg_ptr is a pointer to the segment. (Output)
7. code is a storage system status code. (Output)

Notes

The user must have nonnull access on the segment in order to make it known.

If entryname cannot be made known, a null pointer is returned for seg_ptr and the returned value of code indicates the reason for failure. If entryname is already known to the user's process, code is returned as error_table_\$segknown and the seg_ptr argument contains a nonnull pointer to entryname. If entryname is not already known, and no problems are encountered, seg_ptr contains a valid pointer and code is 0. If ref_name has already been initiated in the current ring, the code is returned as error_table_\$namedup and the seg_ptr argument contains a valid pointer to the segment already initiated. If the seg_ptr argument contains a nonnull pointer, the bit_count argument is set to the bit count of the segment to which seg_ptr points.

Same comment as initiate

Name: hcs_\$terminate_name

The hcs_\$terminate_name entry point terminates one reference name from a segment. If it is the only reference name for that segment, the segment is removed from the address space of the process (made unknown). For a discussion of reference names, see "Constructing and Interpreting Names" in Section I of the MPM Commands.

terminate_name ↓ *the count of initiated*
reduces

is reduced to zero,
then

Usage

```
declare hcs_$terminate_name entry (char(*), fixed bin(35));
call hcs_$terminate_name (ref_name, code);
```

where:

1. ref_name is the reference name to be terminated. (Input)
2. code is a storage system status code. (Output)

Notes

The hcs_\$terminate_noname entry point terminates a null reference name from a specified segment; the hcs_\$terminate_file and hcs_\$terminate_seg entry points terminate all reference names of a segment and make the segment unknown, given its pathname or segment number, respectively.

The term_\$single_refname entry point (see the description of the term_ subroutine) performs the same operation as the hcs_\$terminate_name entry point, unsnapping links as well. Use of the term_ subroutine is recommended.

and decrements a count of ~~the number~~ initiated reference names for the segment. If the count of initiated reference names for the given segment is at a system defined ceiling, then terminate_name will not decrement the count of initiated reference names for the given segment.

return the status code error-table-f usage-count-too-large and

Name: hcs_\$terminate_noname

The hcs_\$terminate_noname entry point terminates a null reference name from the specified segment. If this is the only reference name of the segment, the segment is removed from the address space of the process (made unknown). This entry point is used to clean up after making a segment known and initiating a single null reference name; see also the hcs_\$initiate, hcs_\$initiate_count, and hcs_\$make_seg entry points. For a discussion of reference names, see "Constructing and Interpreting Names" in Section I of the MPM Commands.

Usage

terminate_name values the counts of initiated for ~~to zero~~ to zero

```
declare hcs_$terminate_noname entry (ptr, fixed bin(35));
call hcs_$terminate_noname (seg_ptr, code);
```

where:

- 1. seg_ptr is a pointer to the segment. (Input)
- 2. code is a storage system status code. (Output)

Note

The hcs_\$terminate_name entry point terminates a specified nonnull reference name; hcs_\$terminate_file and hcs_\$terminate_seg entry points terminate all reference names of a segment and make the segment unknown, given its pathname or segment number, respectively.

Same comment as terminate_name

TITLE: Redefine meaning of the copy switch		STATUS	DATE	
AUTHOR: S. Webber		Written	07.01.75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)	Status	P 07/01/75 R 07/08/75	
	<input type="checkbox"/> Lib. Maint. Tools	Expires	01/08/76	
	<input type="checkbox"/> Sys. Anal. Tools	DOCUMENTATION CHANGES		
	<input type="checkbox"/> Sys. Prog. Tools	Document	Specify One or More	
	<input type="checkbox"/> 355	MPM (Vol, Sect.)		
	<input type="checkbox"/> BOS	PLMS (AN #) 78		
	<input type="checkbox"/> Salvager	MOSN (Sect.)		
	<input checked="" type="checkbox"/> Ring Zero	MPAM (Sect.)		
	<input type="checkbox"/> Ring One	MSAM (Sect.)		
	<input type="checkbox"/> SysDaemon/Admin.	Info Segs	X	
<input checked="" type="checkbox"/> Runtime	Other (Name)			
<input type="checkbox"/> User Cmmnd/Subr.	None (Reason)			
Objections/Comments:				

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

SUMMARY:

Redefine the meaning of the copy switch as outlined in MTB 186.

REASONS:

Prerequisite for prelinking. Cleaner supervisor interface eventually.

IMPLICATIONS:

An incompatible change with respect to:

1. error codes returned from hcs_\$delentry and hcs_\$truncate
2. signalling of no_write_permission and not_in_write_bracket.

Users and system programmers should change hcs_\$initiate [_count] calls to pass a copy control switch of lb, (don't create copy).

DETAILED PROPOSAL:

1. integrate with installation of SCT;
2. install new handler for no_write_permission;
3. change hcs_\$delentry and hcs_\$truncate.

To: Distribution
From: Steve Webber
Date: 06/23/75
Subject: Proposed Redefinition of the Copy Switch

This MTB proposes redefinition of the copy switch item in a directory branch. The prime reasons for wanting to do this are:

1. to make a more consistent and simpler mechanism available to users,
2. to simplify the supervisor, and
3. as a part of the implementation of copy_on_write.

Currently the copy switch is used by the initiate primitives to provide a pointer to a copy of a segment (which is potentially nonshareable) rather than to the segment itself. This means that the supervisor must do a good deal of work (in ring 0) implicitly (such as create a segment in the process directory, make it known, initiate a reference name, etc.). This work would better be done in the user ring either implicitly as in response to a copy_on_write fault or explicitly as when a user initiates a segment so that he gets a copy regardless of the setting of the branch item.

The proposal is to:

1. implement copy_on_write in the user ring with system software, and
2. control when a copy_on_write is to take place with the branch item "copy_switch".

The copy_on_write mechanism is simple and would work as follows:

If an attempt is made to write into a segment that the user does not have write permission to, and the copy switch is ON the following actions are taken:

- A. Create a segment in the process directory with the name "!unique...etc.copy_of..." Give the calling

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

process REW access to the segment.

- B. Make this segment known.
- C. Copy the original segment into this segment.
- D. Make the original segment unknown but reserve its segment number.
- E. Make the copy unknown.
- F. Make the copy known with the reserved segment number.

No action is taken on reference names. (It is assumed that reference names have already been dissociated from KSTE's.) Hence, the reference names which were associated with the original segment are now associated with the copy as the copy has the original segment's segment number.

It may be worth the effort to create an address space manager primitive to perform actions D, E, and F above in a single call.

The copy_on_write handler which performs the above tasks would be invoked when a "no_write_permission" (or "not_in_write_bracket") fault occurs. The signal_program will special case this before searching the stack. If the segment does not have the copy switch ON, no_write_permission (or whatever) is signalled in the usual way. If the copy switch is ON, the copy is created, etc. and the fault is restarted immediately without searching the stack.

Clearly there must be a mechanism for allowing users to take whatever action they want to--possibly to ignore copy_on_write events.

This entire change is incompatible and users will need to be told about it in advance. In addition, a consistent replacement must be provided which is as similar to what we have today as is possible. The new actions taken by the hardware primitives are proposed below:

1. hcs_\$initiate

This primitive will work exactly as today. However, system code and in particular,

- the linker, should pass a value of 1b as the copy_control_switch in the initiate calls to avoid a copy at initiate time.
2. hcs_\$initiate_count This primitive will act exactly as the hcs_\$initiate primitive does with respect to the copy switch.
 3. hcs_\$delentry_file (_seg) These primitives will look at the copy switch and treat the copy switch exactly as it treats the safety switch, i.e., an attempt to delete the segment will fail as long as the copy switch is ON.
 4. hcs_\$status (etc.) No change
 5. hcs_\$append (etc.) No change.
 6. hcs_\$add_acl_entries (etc.) No change.
 7. hcs_\$set_bc (etc.) No change.
 8. hcs_\$fs_move_file (_seg) No change.
 9. hcs_\$terminate_file No change.
 10. hcs_\$truncate_file If the copy switch is ON, take no action and return a code. If the copy switch is OFF, truncate as usual.
 11. hcs_\$truncate_seg If the copy switch is ON, cause the effect of a copy_on_write (i.e., if no_write_permission, create a copy with the same segment number) to occur and truncate the copy. If the copy switch is OFF, truncate the segment.

Note that segments created as copies of other segments will, in general, not have the copy switch ON and will be writeable. Hence, it is very unlikely for a copy_on_write fault to occur on one of these segments.

A problem arises when a program of today initiates a segment known to have the copy switch ON. This program can depend on the fact that the pointer returned to him points to a copy and hence actions such as truncate and delete will not have any effect on the original. With the new proposal, however, the returned pointer will point to the original until (if ever) an attempt is made to write into it. Hence, such programs, if they never do attempt to modify the original, will perform their cleanup actions intended for the copy on the original. This is why the truncate and delete primitives will be changed to treat the copy switch specially. An incompatible problem arises here if no modifications are performed as an error code will be returned when an attempt to delete the original is made. Note, however, that all known uses of the copy switch work with no change in behavior as modifications are always done (that is why the copy switch is ON).

It would probably be useful, as noted in MTB-169, to issue warnings when 1) the copy switch is set ON for a segment with write permission granted to some user, and 2) when write permission is granted to a segment whose copy switch is ON.

hcs_\$initiate_count

hcs_\$initiate_count

Name: hcs_\$initiate_count

The hcs_\$initiate_count entry point, given a pathname and a reference name, causes the segment defined by the pathname to be made known and the given reference name initiated. A segment number is assigned and returned as a pointer and the bit count of the segment is returned.

Usage

```
declare hcs_$initiate_count entry (char(*), char(*), char(*),
    fixed bin(24), fixed bin(2), ptr, fixed bin(35));

call hcs_$initiate_count (dir_name, entryname, ref_name, bit_count,
    copy_ctl_sw, seg_ptr, code);
```

where:

1. dir_name is the pathname of the containing directory. (Input)
2. entryname is the entryname of the segment. (Input)
3. ref_name is the reference name. If it is zero length, the segment is initiated with a null reference name. (Input)
4. bit_count is the bit count of the segment. (Output)
5. ~~copy_ctl_sw is obsolete and should be 0.~~ (Input) *Insert ①*
6. seg_ptr is a pointer to the segment. (Output)
7. code is a storage system status code. (Output)

Notes

The user must have nonnull access on the segment in order to make it known.

If entryname cannot be made known, a null pointer is returned for seg_ptr and the returned value of code indicates the reason for failure. If entryname is already known to the user's process, code is returned as error_table_\$segknown and the seg_ptr argument contains a ~~nonnull~~ pointer to entryname. (If entryname is not already known, and no problems are encountered, seg_ptr contains a valid pointer and code is 0.) If ref_name has already been initiated in the current ring, the code is returned as error_table_\$namedup and the seg_ptr argument contains a valid pointer to the segment already initiated. If the seg_ptr argument contains a nonnull pointer, the bit_count argument is set to the bit count of the segment to which seg_ptr points.

Name: hcs_\$initiate

The hcs_\$initiate entry point, given a pathname and a reference name, causes the segment defined by the pathname to be made known and the given reference name initiated. If the reserved segment switch is on, then the segment pointer is input and the segment is made known with that segment number. In this case, the user supplies the initial segment number. If the reserved segment switch is off, a segment number is assigned and returned as a pointer.

Usage

```
declare hcs_$initiate entry (char(*), char(*), char(*), fixed bin(1),
                             fixed bin(2), ptr, fixed bin(35));
```

```
call hcs_$initiate (dir_name, entryname, ref_name, seg_sw, copy_ctl_sw,
                    seg_ptr, code);
```

where:

1. dir_name is the pathname of the containing directory. (Input)
2. entryname is the entryname of the segment. (Input)
3. ref_name is the reference name. If it is zero length, the segment is initiated with a null reference name. (Input)
4. seg_sw is the reserved segment switch. (Input)
0 if no segment number has been reserved
1 if a segment number was reserved
- ~~5. copy_ctl_sw is obsolete and should be 0. (Input)~~ *Insert ①*
6. seg_ptr is a pointer to the segment. (Input or Output)
Input if seg_sw is on (1)
Output if seg_sw is off (0)
7. code is a storage system status code. (Output)

Notes

The user must have nonnull access on the segment entryname in order to make it known.

If entryname cannot be made known, a null pointer is returned for seg_ptr and the returned value of code indicates the reason for failure. If entryname is already known to the user's process, code is returned as error_table_\$segknown and the seg_ptr argument contains a nonnull pointer to entryname. If ref_name has already been initiated in the current ring, the code is returned as error_table_\$namedup and the seg_ptr argument contains a valid pointer to the segment already initiated. If entryname is not already known, and no problems are encountered, seg_ptr contains a valid pointer and code is 0.

①

5. copy_ctl_sw

specifies the action to take with respect to generating a copy of the segment. The parameter is interpreted as follows:

=0 ~~copy~~ create a copy of the specified segment (in the process directory) if the segment has its copy switch ON.

=1 do not create a copy even if the segment's copy switch is ON.

=2 create a copy even if the segment's copy switch is OFF.

7/7/75

A new feature is being added to the system which makes use of the copy switch item in a segment's branch. The feature, copy-on-write, will be a standard feature of the system and will cause a copy of a segment to be created in the process directory when, if ever, the original segment is modified and the copy switch is ON for the original segment. The system will make the copy known with the same segment number as the original. The original will be made unknown. All reference names initially associated with the original will be associated with the copy.

The use of `hcs_$initiate` and `hcs_$initiate_count` to generate a copy at initiate time will also continue to exist as a means of getting a pointer to a copy of a segment. The copy-on-write feature defers creating the copy until necessary and avoids some of the problems with the initiate interfaces.

Users who choose to use the new method need only change the `copy_ctl_sw` parameter (parameter 5) in the `hcs_$initiate` and `hcs_$initiate_count` calls so that a copy is not created at initiate time. A value of 1b will have this effect.

For those interested in the interpretation of this parameter, the following table gives results generated (the meaning of this parameter is not changing):

	Initiate Parameter	0	1	2
Branch Copy Switch				
0		no copy	no copy	copy
1		copy	no copy until mod	copy

TITLE: Implement static handlers		STATUS	DATE
AUTHOR: S. Webber		Written	06.23.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) -Documented in MTB 185 -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR	Category (Check One)		
	<input type="checkbox"/> Lib. Maint. Tools	Status	P 07/01/75 A 07/08/75
	<input type="checkbox"/> Sys. Anal. Tools	Expires	01/08/76
	<input type="checkbox"/> Sys. Prog. Tools	DOCUMENTATION CHANGES	
<input type="checkbox"/> 355	Document	Specify One or More	
<input type="checkbox"/> BOS	MPM (Vol, Sect.)		
<input type="checkbox"/> Salvager	PLMS (AN #) 51, 78		
<input type="checkbox"/> Ring Zero	MOSN (Sect.)		
<input type="checkbox"/> Ring One	MPAM (Sect.)		
<input type="checkbox"/> SysDaemon/Admin.	MSAM (Sect.)		
<input checked="" type="checkbox"/> Runtime	Info Segs pc.info sh.info		
<input type="checkbox"/> User Cmmd/Subr.	Other (Name)		
	None (Reason)		
Objections/Comments:			

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

SUMMARY:

Change real_init_admin programs to set up static handlers for alm, cput, no_write_permission, and not_in_write_bracket.

REASONS:

It makes handling several system conditions more efficient.

IMPLICATIONS:

Users who currently set up handlers for the condition "alm", "cput", "no_write_permission", and "not_in_write_bracket" will no longer get control when the event occurs, unless they take special action.

DETAILED PROPOSAL:

The proposal will be as described in the MTB except:

1. The call to the static handler made by signal_ will be identical to calls to other (standard) handlers (i.e. with 5 arguments).

2. The mapping between names of static handlers and their indices into the SCT will be kept in include files.

Users who have handlers for "cput" and "alarm" will not see these events signalled as the system static handler will not turn the continue bit ON.

Users who have handlers for "no_write_permission" and "not_in_write_bracket" will not see these events signalled if the fault occurred on a segment with the copy switch ON (when copy-on-write is installed).

Users will need to be told of these changes. There doesn't appear to be an easy way to avoid a flag day with respect to these changes but it is also very unlikely that many people will be affected.

To: Distribution
From: Steve Webber
Subject: Handling System Conditions
Date: 4/23/75

Proposed Changes to Handling System Conditions

There are several system events that occur today in a Multics process that do not fit well into the PL/I condition mechanism but were implemented as part of it for lack of a better approach. Typical examples are:

1. events that are always handled (by default) by the same procedure and are independent of stack history, etc.,
2. events that are frequent and do not want to incur the overhead of the signalling mechanism, and
3. events that are to be signalled normally but that are to be monitored regardless of the value of the "continue" parameter in a call to a handler.

Examples in the first class are:

1. alrm
2. cput
3. mme2
4. a small set handled by default_error_handler_.

Examples in the second class are:

1. linkage faults (when removed from ring 0)
2. lot faults, isot faults (for prelinking)
3. copy-on-write faults

Examples in the third class are more vague but might arise in an attempt to monitor the action of a process.

There have been many attempts to solve these problems using techniques often called "static handlers". The proposals in the past, however, have been very general and attempted to solve all problems associated with such a scheme. The proposal presented

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

here is, on the contrary, quite simple and hence does not provide a general solution. It does, however, provide an efficient, workable solution which, when its limitations are understood and handled, I think, can be useful. The scheme would work as follows:

1. A system condition table (SCT) is allocated (when needed) possibly in a combined linkage segment.
2. An index into this table will be saved as part of the machine conditions. (This index will be the same index used today to select the condition name to be signalled.)
3. The system will record and know about 50-75 events which can be signalled from ring 0. These events are precisely the ones signalled today with a few additions to accommodate new features within the system.
4. The standard `signal_` program will call a program to examine the entry in the SCT for the condition being signalled (if machine conditions are provided). If the SCT entry is nonnull, the handler specified therein is invoked. If the SCT entry is null, `signal_` proceeds to scan the stack for handlers as today.
5. Entries will be provided to get and set the value of an SCT entry. The value of an SCT entry is a pointer to a handler and may be null. There is no need for the SCT entry to contain an "entry" value because a property of static handlers is that they cannot depend on any automatic storage of a parent block.
6. The calling sequence for all SCT handlers will be simply:

```
call handler (mc_ptr, condition_name, wcptr, info_ptr,
continue);
```

If the handler sets `continue` to "0", `signal_` will not scan the stack. Otherwise, `signal_` will scan the stack as is done today. (The parameters `wcptr` and `info_ptr` are ignored.)

If a program wants to set up a static handler in this way it should probably first get the previous value of the SCT entry for the condition of interest, and then set up its own value in the SCT. The new handler has the option of calling the previous handler but this, of course, can not be guaranteed to work because of problems such as the old handler being deleted (terminated, etc) unbeknownst to the current handler. Users of this mechanism must have complete knowledge of the execution

environment under consideration.

With this brief overview, then, the following subroutines are proposed:

```
sct_manager_$get_handler (handler, index)
```

```
sct_manager_$set_handler (handler, index)
```

```
sct_manager_$call_handler (mc_ptr, condition_name, wcntr,
info_ptr, continue);
```

where:

handler	is of type entry and is the entry to call when the condition occurs.
index	is fixed bin and specifies which static handler is being set or returned.
mc_ptr	is a pointer to the machine conditions for the fault (event) being signalled.
condition_name	is the name of the event being "signalled"
wcntr	is ignored.
info_ptr	is ignored.
continue	is set to "1"b if the stack should also be searched for a handler and to "0"b if no further processing should be done.

The entry point `sct_manager_$call_handler` is called by `signal_` and is little more than a call forwarder if the SCT entry is nonnull). The program `sct_manager_` alone would know the location and format of the SCT.

It is useful to list features and qualities of a condition or event which would cause that event to be unacceptable to be handled by a static handler. The following are such cases:

1. if the necessary handler for the event requires automatic information from an ancestor block in the stack history (overflow fault, for example), or
2. if more than one program wants to know of the event, i.e., if any user program will likely have a handler for the event (such as "quit").

In contrast, handlers that require no previous stack history, require no new storage other than automatic, and have complete and sole interest in an event are possible candidates for static handlers. The following events, currently signalled, are thus likely candidates for static handlers:

```
alm  
cput  
rme2
```

New events that are good candidates for static handling are:

```
linkage_fault (when the linker is out of ring 0)  
lot_fault  
isot_fault  
copy-on-write fault (no_write_permission)
```

These events are indeed static in nature. Were the events handled in ring 0, as linkage faults are today, the handler would be so static that it would be directly called by the fault intercept module. The fact that the handler can now be removed to the user ring does not change the event in such a way that the full condition mechanism is required. The events are still system events and hence the handlers are system programs. There is no need to search the stack for a user handler.

As mentioned earlier, however, the proposed implementation would allow (hopefully knowledgeable) users to provide their own static handlers that might, for example, turn the continue bit ON so that the stack will be searched. The user-ring implementation, thus, provides more freedom -- the defaults have the effect of today.

The value of the index associated with a specific event (which the system programmer must know) is obtained from the include files `static_handler_names.incl.(pl1 alm)`.

7/7/75

A new feature is being added to the system which improves the system's performance with respect to its handling of certain user-ring system conditions. The user-interface change that will be externally visible is in the signalling of certain system events. In particular, "alarm" and "cput" will no longer be signalled, but will rather be handled directly by system code. Similarly, there will be a change in the signalling of "no_write_permission" and "not_in_write_bracket" such that these, also, will not be signalled if the segment causing the condition to be raised has its copy switch ON (see copy_on_write.info).

Today, handlers for "cput" and "alarm" are established in the process overseer of a process. This requires users wishing to write their own process overseer to know that this is necessary. This strategy is also being changed such that the "real_init_admin" program for a process will take on this task thereby relieving writers of process overseer programs from knowing about this special process (ring) initialization requirement.

Process overseers that establish timer_manager_ as the handler for "cput" and "alarm" will continue to work although this effort is no longer necessary and indeed ignored. If a process overseer establishes any other procedure as the handler for these conditions, they will no longer work in the same fashion. If this is a problem, please notify the system programming staff who can assist you in resolving your problem.

TITLE: Fix bug in absentee. Fails to set bit count correctly		STATUS	DATE
AUTHOR: S. Webber		Written	06.27.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 07.08.75
		Expires	01.08.76
		DOCUMENTATION CHANGES	
		Lib. Maint. Tools	
		Sys. Anal. Tools	
		Sys. Prog. Tools	
		355	Document Specify One or More
		BOS	
		Salvager	MPM (Vol, Sect.)
		Ring Zero	PLMS (AN #)
		Ring One	MOSN (Sect.)
		SysDaemon/Admin.	MPAM (Sect.)
		Runtime	MSAM (Sect.)
		<input checked="" type="checkbox"/> User Cmd/Subr.	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	Bug fix

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

SUMMARY:

Fix bug in absentee causing the bit count not to be set correctly and an "io_error" message being placed in the absout file.

REASON:

Bug fix.

TITLE: Change to get_entry_name		STATUS	DATE
AUTHOR: J. Broughton		Written	06.27.75
<input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL <input type="checkbox"/> Planned for System MR <input type="checkbox"/> Fixes Bug Number(s) <u>unreported</u> <input type="checkbox"/> Documented in MTB <input type="checkbox"/> User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no <input type="checkbox"/> Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no <input type="checkbox"/> Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse <input type="checkbox"/> Replaces MCR _____		RAB	
		Category (Check One)	Status <u>A 07.08.75</u>
		Lib. Maint. Tools	Expires <u>01.08.76</u>
		DOCUMENTATION CHANGES	
		355	Document Specify One or More
		BOS	MPM (Vol, Sect.)
		Salvager	PLMS (AN #)
		Ring Zero	MOSN (Sect.)
		Ring One	MPAM (Sect.)
		SysDaemon/Admin.	MSAM (Sect.)
		<input checked="" type="checkbox"/> Runtime	Info Segs
		User Cmnd/Subr.	Other (Name)
			None (Reason) <u>PLM does not yet exist</u>
Objections/Comments:			

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

SUMMARY:

Modify get_entry_name_ to distinguish between a begin block entry (which has no name) and a procedure entry (which does have a name), and return, in the former case, the string "begin block" as the name, and a new code error_table_\$begin_block. Also modify the callers of get_entry_name_ to take advantage of this information.

REASONS:

Currently, no distinction is made, and the result is indeterminate. It may return a null string and a code indicating that no name was found; it may return the name of any external symbol in the segment; or it may return garbage characters. Returning just "begin block" is not sufficient as the callers of get_entry_name_ are in many cases explicitly interested in a procedure name. Also callers may add such other information as they see fit.

DETAILED PROPOSAL:

1. Make modifications to `get_entry_name_`
2. Add new code to `error_table_`
3. Modify callers:
 - a. Change `get_block_name_` to check for new code rather than explicitly checking for a `begin` block itself. (This is necessary since with separate static, there is an additional `begin` block entry operator.)
 - b. Change `getonsource` (`$get_onloc`) to check for new code and return the name of the procedure immediately containing the `begin` block. This will produce correct results for the `onloc` builtin for the case where an error occurs in a `begin` block. The result is currently in error.
 - c. Most other callers will function adequately with the described changes; `default_error_handler_` has a number of problems with `begin` blocks in general, and will be modified at a later time.

TITLE: Install Emergency Fix to PL/I Optimizer bug		STATUS	DATE
AUTHOR: R. A. Barnes		Written	06.27.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) PL/I 1375 -Documented in MTB -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR	Category (Check One)		
	<input type="checkbox"/> Lib. Maint. Tools	DOCUMENTATION CHANGES	
	<input type="checkbox"/> Sys. Anal. Tools	Document	Specify One or More
	<input type="checkbox"/> Sys. Prog. Tools	MPM (Vol, Sect.)	
<input type="checkbox"/> 355	PLMS (AN #)		
<input type="checkbox"/> BOS	MOSN (Sect.)		
<input type="checkbox"/> Salvager	MPAM (Sect.)		
<input type="checkbox"/> Ring Zero	MSAM (Sect.)		
<input type="checkbox"/> Ring One	Info Segs		
<input type="checkbox"/> SysDaemon/Admin.	Other (Name)		
<input type="checkbox"/> Runtime	None (Reason) no interface change		
<input checked="" type="checkbox"/> User Cmmnd/Subr.			
Objections/Comments:			

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

SUMMARY:

Install a new optimizer to the PL/I compiler which fixes bug 1375 in which the optimizer fails, on some occasions, to properly handle variables whose first, but not only reference, was as argument to the addr builtin.

Ver. 3		
741022	MULTICS CHANGE REQUEST	MCR 1254
TITLE: Bug fix to clear_profile and clear_reqfile		STATUS DATE
AUTHOR: F. C. Smith		Written 06/23/75
		Status A 07/08/75
		Expires 01/08/76
Planned for System: not applicable		
Fixes Bug Number(s): not applicable		CATEGORY (check one)
Documented in MTB: not applicable		() Lib. Maint. Tools
Incompatible Change: no		() Sys. Anal. Tools
User/Operations-visible Interface Change: no		() Sys. Prog. Tools
Coded in: (<input checked="" type="checkbox"/>) PL/I () ALM () other-see below		() 355
Performance: () better (<input checked="" type="checkbox"/>) same () worse		() BOS
		() Salvager
DOCUMENTATION CHANGES (specify one or more)		() Ring Zero
MPM (vol,sect) MPAM (sect)		() Ring One
MOSN (sect) MSAM (sect)		(<input checked="" type="checkbox"/>) SysDaemon/Admin
PLMs (AN#)		() Runtime
Info Segs		() User Command/Subr
Other		
_None (reason) bug fix		
OBJECTIONS/COMMENTS:		

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

SUMMARY: Modify clear_profile and clear_reqfile to set the bit count on profile and reqfile.

REASONS: The bit count of these segments is left inconsistent with the current length after monthly billing.

Ver. 3	MULTICS CHANGE REQUEST	MCR 1255
741022		
TITLE: Bug fix to edit_proj and new_proj		STATUS DATE
		Written 06/25/75
AUTHOR: F. C. Smith		Status 07/08/76
		Expires 01/08/76
Planned for System: not applicable		CATEGORY (check one)
Fixes Bug Number(s): not applicable		() Lib. Maint. Tools
Documented in MTB: not applicable		() Sys. Anal. Tools
Incompatible Changes: no		() Sys. Prog. Tools
User/Operations-visible Interface Changes: no		() 355
Coded in: <input checked="" type="checkbox"/> PL/I () ALM () other-see below		() BOS
Performance: () better <input checked="" type="checkbox"/> same () worse		() Salvager
DOCUMENTATION CHANGES (specify one or more)		() Ring Zero
MPM (vol,sect) MPAM (sect)		() Ring One
MOSN (sect) MSAM (sect)		<input checked="" type="checkbox"/> SysDaemon/Admin
PLMs (AN#)		() Runtime
Info Segs		() User Command/Subr
Other		
_None (reason) bug fix		
OBJECTIONS/COMMENTS:		

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

SUMMARY: Modify edit_proj and new_proj so that they will put a copy of the pmf for a newly delegated project into the designated directory.

REASONS: Project administrators need to have a copy of the pmf for their projects. System administrators are currently using the supposedly obsolete "delegate" command to delegate projects.

IMPLICATIONS: System administrators will be able to delegate projects using edit_proj and new_proj.

TITLE: Experiment with new command processor		STATUS	DATE
AUTHOR: S. Herbst		Written	06.27.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		
	<input type="checkbox"/> Lib. Maint. Tools	Status	A 07/08/75
	<input type="checkbox"/> Sys. Anal. Tools	Expires	01/08/76
	<input type="checkbox"/> Sys. Prog. Tools	DOCUMENTATION CHANGES	
<input type="checkbox"/> 355	Document	Specify One or More	
<input type="checkbox"/> BOS	MPM (Vol, Sect.)		
<input type="checkbox"/> Salvager	PLMS (AN #)		
<input type="checkbox"/> Ring Zero	MOSN (Sect.)		
<input type="checkbox"/> Ring One	MPAM (Sect.)		
<input type="checkbox"/> SysDaemon/Admin.	MSAM (Sect.)		
<input type="checkbox"/> Runtime	Info Segs		
<input checked="" type="checkbox"/> User Cmnd/Subr.	Other (Name)		
	None (Reason)	none yet	
Objections/Comments:			

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

SUMMARY:

Investigate the feasibility of installing a command processor consisting of a single module that implements a well-defined full command language almost identical to that used today. This module would replace the current command processor and full command processor programs and include the ability to expand abbreviations.

REASONS:

Maintaining the current full command processor is difficult because it requires keeping track of a great number of implicit assumptions and complicated interactions between programs. Restructuring to use fast EIS builtins for string processing and to parse command lines in list form rather than string form, we should be able to design a faster and more elegant means of fulfilling the current requirements.

IMPLICATIONS:

Three changes are intended for the new implementation:

- 1) Abolishing the concept of maximum expanded command line size, which is of no use and creates one particular anomaly with respect to the sizes of active

function return strings. The new command processor will not create an expanded command line in string form and will allocate whatever list space is needed.

- 2) Changing the rule for when active functions are evaluated in a line containing several commands. The only change is that no active functions after a semicolon are evaluated until the command line before the semicolon has been processed completely.
- 3) Mismatched iteration sets will be diagnosed before any processing is done rather than, as currently, when one runs out before another.

TITLE: Fix Bugs in I/O Interfacer		STATUS	DATE
AUTHOR: Noel I. Morris		Written	75.07.01
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL		Status	A 07/08/75
-Planned for System MR		Expires	01/08/76
-Fixes Bug Number(s) _____		DOCUMENTATION CHANGES	
-Documented in MTB _____		Document	Specify One or More
-User/Operations-visible		MPM (Vol, Sect.)	
Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		PLMS (AN #)	
-Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		MOSN (Sect.)	
-Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same		MPAM (Sect.)	
<input type="checkbox"/> Worse		MSAM (Sect.)	
-Replaces MCR _____		Info Segs	
Objections/Comments:		Other (Name)	
		None (Reason)	

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

Summary:

Several entries in the I/O Interfacer are passed fixed binary offsets and lengths. If these numbers are negative, certain boundary violation conditions will not be detected.

Proposal:

Change the modules ioi_connect, ioi_set_status, and ioi_workspace to reject negative arguments.

Implications:

Any users who make calls to the I/O Interfacer with negative values in order to hoodwink the boundary checking mechanisms will have to revise their code.

TITLE: Improve delete 's description of failure in deleting subtree		STATUS	DATE
AUTHOR: S. Herbst		Written	07.01.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> AIM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 07/08/75
		Expires	01/08/76
		DOCUMENTATION CHANGES	
Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input checked="" type="checkbox"/> User Cmmd/Subr.		Document	Specify One or More
Objections/Comments:		MPM (Vol, Sect.)	III, 2
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Change delete_, when:

1. deleting a directory;
2. and query switch is on (command environment);
3. and some segment cannot be deleted in a directory;

to print an error message for every segment that cannot be deleted in the given directory before returning an error code.

TITLE: Un-document iox_\$open append argument		STATUS	DATE
AUTHOR: S. Herbst		Written	07.01.75
<input type="checkbox"/> Coded in <input checked="" type="checkbox"/> PL/I <input checked="" type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL <input type="checkbox"/> Planned for System MR <input type="checkbox"/> Fixes Bug Number(s) _____ <input type="checkbox"/> Documented in MTB _____ <input type="checkbox"/> User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no <input type="checkbox"/> Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse <input type="checkbox"/> Replaces MCR _____		Status	A 07/08/75
		Expires	01/08/76
		DOCUMENTATION CHANGES	
		Category (Check One)	Document Specify One or More
		<input type="checkbox"/> Lib. Maint. Tools	
		<input type="checkbox"/> Sys. Anal. Tools	
		<input type="checkbox"/> Sys. Prog. Tools	
		<input type="checkbox"/> 355	
		<input type="checkbox"/> BOS	
		<input type="checkbox"/> Salvager	MPM (Vol, Sect.)
		<input type="checkbox"/> Ring Zero	PLMS (AN #)
		<input type="checkbox"/> Ring One	MOSN (Sect.)
		<input type="checkbox"/> SysDaemon/Admin.	MPAM (Sect.)
		<input type="checkbox"/> Runtime	MSAM (Sect.)
		<input checked="" type="checkbox"/> User Cmnd/Subr.	
Objections/Comments:		<input type="checkbox"/> Info Segs	
		<input type="checkbox"/> Other (Name)	
		<input type="checkbox"/> None (Reason) no changes	

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

SUMMARY:

Eliminate mention of the append (extend) argument to iox_\$open from comments in the various iocb include files.

REASONS:

This argument is no longer used. Comments should be changed to say "not_used" in place of "append".

TITLE: Install Priority Scheduler		STATUS	DATE
AUTHOR: R. Mullen		Written	07.01.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input checked="" type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.0 -Fixes Bug Number(s) _____ -Documented in MTB 163 -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 07/08/75
		Expires	01/08/76
Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input checked="" type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input type="checkbox"/> User Cmmnd/Subr.		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #) 73	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Install priority scheduler. By default the initializer will go in one work_class, all other processes in another.

REASONS:

This modification to the priority scheduler allows increased administrative control over use of cpu resources.

IMPLICATIONS:

- a. Because this scheduler requires an additional 160 words of storage in tc_data, the "tcd" config card may need to be changed at various sites. If a site's tcd card is currently correct, it is sufficient to reduce the number of APTE's allocated by 4.
- b. Scheduler overhead may increase .01% per workclass defined. Other, equally minor, savings have been achieved.
- c. Paging overhead may increase if >8 workclasses are defined. It may decrease if this scheduler is used to limit service to processes which are known to page heavily.
- d. Response to trivial interactions is claimed to be identical to the current scheduler. This is supported by both analysis of the code involved and such objective metering as is possible.

DETAILED PROPOSAL:

As MTB-163, but with addition of unconditional first-come first-served running of just interacted processes before going to new priority scheduler code.

TITLE: Install new version of vfile_ I/O module		STATUS	DATE	
AUTHOR: M. Asherman		Written	07.01.75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input checked="" type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR <u>3.0</u> -Fixes Bug Number(s) <u>unreported</u> -Documented in MTB <u>061</u> -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status	
		Lib. Maint. Tools	Expires	
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
		355	MPM (Vol, Sect.)*, vfile I/O modu	le
	BOS	PLMS (AN #)		
	Salvager	MOSN (Sect.)		
	Ring Zero	MPAM (Sect.)		
	Ring One	MSAM (Sect.)		
	SysDaemon/Admin.	Info Segs		
	Runtime	Other (Name)		
	<input checked="" type="checkbox"/> User Cmnd/Subr.	None (Reason)		
Objections/Comments:				
* iox_ spec's revised for positioning properly in indexed files.				

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

SUMMARY:

This version incorporates the following changes in the vfile_ I/O module.

- 1) recovery from interrupted openings (crashes) for indexed files as an automatic feature.
- 2) inter-process sharing of indexed files opened for modification is supported with a single file lock.
- 3) the control operation accepts an order "seek_head" that finds the first record whose key has a specified relation to a given key (required by COBOL).
- 4) several unreported bugs have been fixed - the most serious caused self-destruction of large indices.
- 5) substantial performance improvements for most operations on indexed files.

DETAILED PROPOSAL:

See MPM documentation attached.

PROPOSED SPECS FOR vfile_

I/O Module

Name: vfile_

This I/O module supports the I/O from/to files in the storage system. All logical file types are supported.

Entries in this module are not called directly by users; rather, the module is accessed through the I/O system. See the MPM section, the Multics I/O system, for a general description of the I/O system, and see the MPM section, file I/O, for a discussion of files.

Attach Description

The attach description has the following form:

vfile_ pathname -option_1-...-option_n-

1. pathname is the absolute or relative pathname of the file.
2. option_i can be chosen from the following list of options.

-extend specifies extension of the file if it already exists. This option is only valid with openings for output or input_output.

-share -wait_time- allows an indexed file to be open in more than one process at the same time, even though not all openings are for input. (See Multiple Openings below). This option is only valid with openings for direct_input, direct_update, or direct_output with -extend. The wait_time, if specified, is the maximum time in seconds that this process will wait to perform an operation on the file. A value of minus one means the process may wait indefinitely. If no wait_time is given, a default value is used.

To form the attach description actually used in the attachment, the pathname is expanded to obtain an absolute pathname.

Opening and Access Requirements

All opening modes are supported. For an existing file, the mode must be compatible with the file type. (See the MPM Section, File I/O). The mode must be compatible with any options in the attach description.

If the opening is for input only and without the -share option, only "r" access is required on the file. In all other cases "rw" access is required on the file.

Rewrite Operation

If the file is a sequential file, the new record must be the same length as the replaced record. If not, the code returned is error_table_\$long_record or error_table_\$short_record.

Delete Operation

If the file is a sequential file, the record is logically deleted, but the space it occupies is not recovered.

Modes Operation

This operation is not supported.

Control Operation

The order "seek_head" is accepted when the I/O switch is open for keyed_sequential_input or keyed_sequential_update. For this order the info_ptr argument must point to a structure of the following form:

```
dc1 1  info based (info_ptr),
     2  relation_type fixed,
     2  n fixed,
     2  search_key char (0 refer (n));
```

The operation locates the first record with a key whose head has the specified relation with the given search_key. The next record position and (for keyed_sequential_update) the current record position are set to the record. If no such record exists, the code, error_table_\$no_record is returned.

The head of a record's key is the first n characters of the key, the key being extended by blanks if it has fewer than n characters. The allowed values for info.relation_type are:

```
0    head = search_key
1    head >= search_key
2    head > se rch_key
```

Multiple Openings

It is possible to have or attempt to have multiple openings of the same file, that is to have two or more open I/O switches attached to the same file. These switches might be in the same process or in different processes. With respect to the effects of multiple openings, the various opening modes can be divided into four classes (explained below). Multiple openings in which the opening modes are in more than one class are invalid, as are multiple openings within certain classes. The vfile_ module prevents some cases of multiple opening, the code error_table_\$file_busy being returned by the open operation. In cases where an invalid multiple opening does occur, I/O operations will cause unpredictable errors in the processes involved, and the contents of the files may be damaged.

The classes of multiple openings are:

1. Openings for input without the -share option.

Any number of openings in this class are allowed. The existence of an opening in this class never causes damage to the file. When this class of opening is attempted, the existence of all class 2 and 3 openings and some class 4 openings will be detected for structured files.

2. Openings for output or input_output without file extension.

Only one opening is allowed. The existence of another opening is never detected when this class of opening is attempted. The file is simply replaced by an empty file of the appropriate type. If the file was already open with an opening of any class except (1), the contents of the new file will probably be damaged.

3. Openings for update without the -share option and for output or input_output without the -share option and with file extension.

Only one opening of this class is allowed. For structured files, multiple openings within the class are detected. An invalid multiple opening involving an opening of this class and other openings of class 4, may be detected. If not, the only effect is that the class 3 opening locks the file for the entire opening.

4. Openings with the -share option.

(This applies to `direct_input`, `direct_update`, and `direct_output` with `-extend` only). Any number of openings of this type are allowed. When a process performs an operation on the file, the file is locked. Other processes attempting an operation while the file is locked will wait up to the limit specified by the `wait_time` option in the `attach_description`. If the operation is not carried out because of the `wait_time` limit, the code `error_table_$file_busy` is returned.

There are two system status codes that pertain only to class 4 openings: `error_table_$asynch_deletion` and `error_table_$asynch_insertion`. The first is returned by the `read_record`, `read_length`, and `rewrite_rec rd` operations when a record located by a `seek_key` operation has been deleted (by an operation in some other opening). The second is returned by `write_record` when a record with the key for insertion (defined by a `seek_key` operation) has already been inserted (by some other opening).

Interrupted Openings

If a process opens a file and terminates without closing the file, the file may be left in an intermediate state that prohibits normal I/O operations on the file. The exception is openings for input only. The details depend on the particular type of file as follows.

1. Unstructured files.

In general, the bit count of the file's last segment will not be properly set. This condition is not detected at subsequent openings, and part of the file's contents may be overwritten or ignored.

2. Sequential file.

In general, certain descriptors in the file and the bit count of the files' last segment will not be properly set. This condition is detected at a subsequent open, and the code `error_table_$file_busy` is returned.

3. Indexed files.

In general, the bit counts of the files' segments will not be properly set, and the files' contents will be in a complex intermediate state (e.g. a record will be deleted but not its key in the index). This situation is detected at a subsequent open or at the beginning of the next operation, if the file is already open with the `-share` option. Unless the opening is for input without the `-share` option, the file is automatically adjusted. If this situation is detected by an opening for input without the `-share` option, the code `error_table_$file_busy` is returned. Opening the file for update will properly adjust the file.

When an indexed file is adjusted, the interrupted operation (`write_record`, `rewrite_record`, or `delete_record`), if any, is completed. However for `rewrite_record`, it may happen that the bytes of the record are potentially incorrect. (Everything else will be correct). In this case an error message is printed on the console. The user can rewrite or delete the record as required. The completion of an interrupted write operation may also produce an incorrect record, in which case the defective record and its key are automatically deleted from the file.

Inconsistent Files

The code `error_table$bad_file` (console message: "File is not a structured file or is inconsistent") may be returned by operation on structured files. It means that an inconsistency has been detected in the file. Possible causes are:

1. The file is not a structured file of the required type.
2. A program accidentally modified some words in the file.

Multics Change Request

TITLE: Changing error codes returned by dir_control_error		STATUS	DATE
AUTHOR: A. Kobziar		Written	07/02/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR <u>1215</u>		Status	A 07/15/75
		Expires	01/15/76
		DOCUMENTATION CHANGES	
Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input checked="" type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input checked="" type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input type="checkbox"/> User Cmmd/Subr.		Document	Specify One or More
Objections/Comments:		MPM (Vol, Sect.)	
		PLMS (AN #)	61
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Change dir_control_error\$ to differentiate between access denial due to wrong mode or access isolation restriction by returning error_table\$ai_restricted if the process' access authorization is not equal to the directory's access class. Change the message associated with error_table_\$incorrect_access from "incorrect access to directory" to "incorrect access mode to directory".

REASONS:

Programs (such as del_dir_tree) which assume that an error return of incorrect_access can always be cured by setting access for the user will now work properly on upgraded directories.

TITLE: Add two mailbox wakeup entries		STATUS	DATE
AUTHOR: S. Herbst		Written	07.07.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MFB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 07/15/75
		Expires	01/15/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	69
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Install two mailbox facility entry points needed to implement secure wakeup messages, in particular, to make send_mail_work. These entry points, mailbox_\$wakeup_add_index and mailbox_\$accept_wakeups_index are described on the following pages.

REASONS:

Secure wakeups require that a user's event channel id be stored in his mailbox, protected by extended access. Since access may change, another user's right to use the channel must be interpreted each time and the channel id must not be released to ring4. It is convenient to let a single ring1 primitive send the wakeup, add the message, and report any errors doing either.

DETAILED PROPOSAL:

Two entry points to be added to the gate mailbox_ and the module mbx_mseg_, also minor changes to mseg_ and mseg_add_ to make some access isolation checks needed to send a wakeup to another process.

Entry: mailbox_\$wakeup_add_index

This entry points adds a message to a specified mailbox and sends a wakeup to the owner of the mailbox.

Usage:

```
dcl mailbox_$wakeup_add_index entry(fixed bin,ptr,
                                     fixed bin,bit(36),fixed bin(71),fixed bin(35));

call mailbox_$wakeup_add_index (mbx_index, msg_ptr,
                                msg_bitcnt, switches, id, code);
```

where:

1. mbx_index is the index of a mailbox. (Input)
2. msg_ptr is a pointer to the message to be added. (Input)
3. msg_bitcnt is the length of the message in bits. (Input)
4. switches are: (Input)

normal_wakeup is ON if a normal wakeup is to be sent.

urgent_wakeup is ON if an urgent wakeup is to be sent.

always_add is ON if the message is to be added to the mailbox regardless of whether a wakeup could be sent.

6. code is a standard status code, for example:

```
error_table_$action_not_performed
  when sending a normal wakeup if the recipient is
  accepting urgent but not normal wakeups. This
  error code effectively means, "Try urgent."
error_table_$moderr
  if insufficient access to add a message.
error_table_$wakeup_denied
  if unable to send an urgent wakeup.
error_table_$invalid_channel
  if the recipient is not logged in or has not
  initialized for accepting message wakeups.
error_table_$no_info
  if the sender of a wakeup is not allowed to know
  what has taken place because the recipient has
  higher AIM authorization than him.
```


Entry: mailbox_\$accept_wakeups_index

This entry point manipulates information in the mailbox header to allow and defer normal and urgent message wakeups.

Usage:

```
dcl mailbox_$accept_wakeups_index entry(fixed bin,  
                                         fixed bin(71),bit(36),fixed bin(35));
```

```
call mailbox_$accept_wakeups_index (mbx_index,  
                                     channel_id, switches, code);
```

where:

1. `mbx_index` is the index of a mailbox. (Input)
2. `channel_id` is the id of an event-call channel created by the caller. (Input)
3. switches are: (Input)

`normal_wakeup` is ON if normal wakeups are to be allowed.

`urgent_wakeup` is ON if urgent wakeups are to be allowed. If `normal_wakeup` is on, `urgent_wakeup` must be on.

4. `code` is a standard status code, probably zero.

TITLE: Fix bug in enter_abs_request		STATUS	DATE
AUTHOR: S. Herbst		Written	07.07.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		
	Lib. Maint. Tools		DOCUMENTATION CHANGES
	Sys. Anal. Tools		
	Sys. Prog. Tools		
	355		
BOS			
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	no change

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

SUMMARY:

Change enter_abs_request to print an error message when the specified about file is a directory.

REASONS:

Currently, no message is printed in this case and no request is submitted.

Multics Change Request

TITLE: Fix worst case "flush -ck " bugs in page control.		STATUS	DATE
AUTHOR: B. Greenberg		Written	7/7/75
-Coded in: <input type="checkbox"/> PL/I <input checked="" type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.0 -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Expires
	Lib. Maint. Tools		A 07/15/75
	Sys. Anal. Tools		DOCUMENTATION CHANGES
	Sys. Prog. Tools		355
	355		Document
BOS		MPM (Vol, Sect.)	
<input checked="" type="checkbox"/> Ring Zero		PLMS (AN #)	
Ring One		MOSN (Sect.)	
SysDaemon/Admin.		MPAM (Sect.)	
Runtime		MSAM (Sect.)	
User Cmmd/Subr.		Info Segs	
Objections/Comments:		Other (Name)	
		None (Reason)	

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

(Retroactive MCR for emergency fixes constituting MSS 25.8a, MSS 25.9d)

SUMMARY: Flush, invoked with a -check option, causes a plethora of "not yet on paging device" pages to appear. Two bugs relative to this have manifested themselves.

1. find_core does not recognize when all of core is in this anomalous state.
2. When an attempt is made to migrate a "not yet on paging device" page to the paging device, and no paging device records are allocatable, the page remains "not yet on the paging device", and cannot be replaced from core.

REASONS: When all core becomes "not yet on paging device", Multics runs out of core.

IMPLICATIONS: Reliability.

DETAILED PROPOSAL:

To fix (1), cause find_core to invoke claim_mod_core after a fixed number of modified or "not yet on paging device" pages have been skipped.

To fix (2), modify write_page to turn off the "not yet on paging device" flag in a PTW if an attempt to allocate paging device for that page fails. This will cause the page to be faulted in from disk if referenced again.

Multics Change Request

TITLE: Fix Core Garbage Collector not to Collect AUTHOR: Out of Service Pages B. Greenberg		STATUS Written/ 7/7/75 Status A 07/15/75 Expires 01/15/76	DATE
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR asap -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input checked="" type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input type="checkbox"/> User Cmmd/Subr.	DOCUMENTATION CHANGES Document Specify One or More MPM (Vol, Sect.) PLMS (AN #) MOSN (Sect.) MPAM (Sect.) MSAM (Sect.)	
Objections/Comments:		Info Segs Other (Name) None (Reason)	

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

SUMMARY: Modify collect_free_core, the core garbage collector which runs in initialization, not to collect out-of-service pages.

REASONS: The page control protocol changes of system 25-5 make out-of-service pages look garbage-collectable to the current collect_free_core. It is rather curious that this has not caused any failures until recently, as it is a fatal bug.

IMPLICATIONS: Reliability.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1270
TITLE: Fix qedx to handle 256K segments.		STATUS: DATE
AUTHOR: VanVleck		Written: 06/22/75
		Status: A 07/15/75
		Expires: 12/22/75
Planned for System: not applicable		
Fixes Bug Number(s): not applicable		CATEGORY (check one)
Documented in MTB: not applicable		() Lib. Maint. Tools
Incompatible Change: no		() Sys. Anal. Tools
User/Operations-visible Interface Change: no		() Sys. Prog. Tools
Coded in: <input checked="" type="checkbox"/> PL/I () ALM () other-see below		() 355
Performance: () better <input checked="" type="checkbox"/> same () worse		() BOS
		() Salvager
DOCUMENTATION CHANGES (specify one or more)		() Ring Zero
MPM (vol,sect) qedx MPAM (sect)		() Ring One
MOSN (sect) MSAM (sect)		() SysDaemon/Admin
PLMs (AN#)		() Runtime
Info Segs		<input checked="" type="checkbox"/> User Command/Subr
Other		
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Re-declare variables in qedx so that it is able to edit 256K segments.

REASONS:

The current qedx sometimes gets out-of-bounds errors and sometimes types thousands of NUL characters when invoked on a 256K segment.

IMPLICATIONS:

Fix bugs.

Multics Change Request

TITLE: Change mexp to accept "&l" control sequence		STATUS	DATE
AUTHOR: S. Webber		Written	07.08.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 07/15/75
		Expires	01/15/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
			MPM (Vol, Sect.)
			PLMS (AN #) 51
			MOSN (Sect.)
			MPAM (Sect.)
			MSAM (Sect.)
			Info Segs
			Other (Name)
			None (Reason)
Objections/Comments:			

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

SUMMARY:

Add a new control sequence to mexp, &ln, which expands into a decimal number equal to the length of the nth parameter to the macro in which it occurs.

REASON:

This is a useful extension.

8. dup causes the text up to the next dupend found in the text to be duplicated n times where n is the decimal value of the (first) parameter to the pseudo-operation.
9. &i is expanded to be the particular parameter in an iterated list for which the current iteration expansion is being done (see below).
10. &x is expanded into the decimal integer corresponding to the argument position of the iteration argument for which the current iteration is being done (see "Examples" below).
11. &An is expanded to be the n+1'st argument to the mexp command.
12. ifarg if ifarg occurs in the context of an opcode or pseudo-operation it causes conditional expansion of the text up to the next ifend depending on whether or not the first parameter to the pseudo-operation is one of the arguments to the mexp command (other than the source name).

13. &n

is expanded into the decimal integer corresponding to the number of characters in the n'th argument.

If a parameter is not specified for a particular parameter position, a zero length string is used for expansion.

The argument &0 expands to be the first label on the statement invoking a macro.

Any parentheses around a parameter are stripped off upon expansion. Parentheses used in this manner are treated as quoting characters.

Blanks cannot appear in a macro parameter list unless within a parenthesized parameter.

Iteration

The iteration feature is invoked by passing a parenthesized list of parameters in the parameter position for the specified iteration. The parameter number for an iteration sequence immediately follows the &(of its definition. (If no parameter number is specified, 1 is assumed.) Iterated arguments are scanned in the same manner as macro arguments and hence quoting can be done with the use of parentheses.

TITLE: Install new plio2_recio_		STATUS	DATE
AUTHOR: R. Schoeman		Written	07.03.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) <u>1378</u> -Documented in MTB -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Expires
	<input type="checkbox"/> Lib. Maint. Tools	DOCUMENTATION CHANGES	
	<input type="checkbox"/> Sys. Anal. Tools		Document Specify One or More
	<input checked="" type="checkbox"/> Sys. Prog. Tools		MPM (Vol, Sect.)
<input type="checkbox"/> 355		PLMS (AN #)	
<input type="checkbox"/> BOS		MOSN (Sect.)	
<input type="checkbox"/> Salvager		MPAM (Sect.)	
<input type="checkbox"/> Ring Zero		MSAM (Sect.)	
<input type="checkbox"/> Ring One			
<input type="checkbox"/> SysDaemon/Admin.			
<input checked="" type="checkbox"/> Runtime			
<input type="checkbox"/> User Cmnd/Subr.			
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	Bug fix

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

SUMMARY:

Install a slightly changed plio2_recio_.

REASONS:

Prior to this change, if during execution of a rewrite or delete statement with a key the key was not found, the "transmit" condition, instead of the correct "key" condition, was raised. This change fixes that bug.

TITLE: Eliminate TN1200 padding for raw mode		STATUS	DATE
AUTHOR: S. Herbst		Written	07.08.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 07/15/75
		Expires	01/15/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	57
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Change tty_ to not pad Terminet 1200 output when running in raw mode.

REASON:

Raw mode should not edit the user's output stream in any way.

Ver. 3	MULTICS CHANGE REQUEST	MCR	1274
741022			
TITLE: Fix bug in the library command ldelete.		STATUS	DATE
AUTHOR: P. Kelley		Written	07/02/75
		Status	A 07/15/75
		Expires	01/02/76
Planned for System:	not applicable	CATEGORY (check one)	
Fixes Bug Number(s):	not applicable	<input checked="" type="checkbox"/>	Lib. Maint. Tools
Documented in MTB:	not applicable	<input type="checkbox"/>	Sys. Anal. Tools
Incompatible Changes:	no	<input type="checkbox"/>	Sys. Prog. Tools
User/Operations-visible Interface Change:	no	<input type="checkbox"/>	355
Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other-see below		<input type="checkbox"/>	BOS
Performance: <input checked="" type="checkbox"/> better <input type="checkbox"/> same <input type="checkbox"/> worse		<input type="checkbox"/>	Salvager
<u>DOCUMENTATION CHANGES (specify one or more)</u>		<input type="checkbox"/>	Ring Zero
MPM (vol,sect)	MPAM (sect)	<input type="checkbox"/>	Ring One
MOSN (sect)	MSAM (sect)	<input type="checkbox"/>	SysDaemon/Admin
PLMs (AN#)	AN80 - (no changes necessary)	<input type="checkbox"/>	Runtime
Info Segs		<input type="checkbox"/>	User Command/Subr
Other			
OBJECTIONS/COMMENTS:			

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

Summary:

The library command ldelete currently fails to perform its job, i.e. deleting library segments. This proposal will modify ldelete to turn off a segment's safety switch before attempting to delete a segment, thus enabling the command to function properly.

Ver. 3	MULTICS CHANGE REQUEST	MCR	1275
741022			
TITLE: Modify the online updating tools to copy words instead of bits.		STATUS	DATE
AUTHOR: P. Kelley		Written	07/07/75
		Status	A 07/15/75
		Expires	01/07/76
Planned for System: not applicable		CATEGORY (check one)	
Fixes Bug Number(s): not applicable		<input checked="" type="checkbox"/> Lib. Maint. Tools	
Documented in MTB: not applicable		<input type="checkbox"/> Sys. Anal. Tools	
Incompatible Change: no		<input type="checkbox"/> Sys. Prog. Tools	
User/Operations-visible Interface Change: no		<input type="checkbox"/> 355	
Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other-see below		<input type="checkbox"/> BOS	
Performance: <input checked="" type="checkbox"/> better <input type="checkbox"/> same <input type="checkbox"/> worse		<input type="checkbox"/> Salvager	
DOCUMENTATION CHANGES (specify one or more)		<input type="checkbox"/> Ring Zero	
MPM (vol,sect) MPAM (sect)		<input type="checkbox"/> Ring One	
MOSN (sect) MSAM (sect)		<input type="checkbox"/> SysDaemon/Admin	
PLMs (AN#) AN80 - (no changes necessary)		<input type="checkbox"/> Runtime	
Info Segs		<input type="checkbox"/> User Command/Subr	
Other			
OBJECTIONS/COMMENTS:			

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

Summary:

Due to a current hardware bug (#062) segments sometimes aren't copied correctly by the updating tools. The procedure upd_copy_seg_task_ will be modified to use the "mir" instruction instead of the "csi" instruction to perform the copy. There will also be coded in a check after the copy to make sure that the transfer was successful.

Multics Change Request

TITLE: Priority Scheduler Answering Service		STATUS	DATE
AUTHOR: T. Casey		Written	07/08/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input checked="" type="checkbox"/> ALM <input checked="" type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.0 -Fixes Bug Number(s) _____ -Documented in MTB 193 -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		DOCUMENTATION CHANGES Document Specify One or More MPM (Vol, Sect.) PLMS (AN #) 66 MOSN (Sect.) MPAM (Sect.) MSAM (Sect.) 4 Info Segs Other (Name) None (Reason)
	Lib. Maint. Tools		
	Sys. Anal. Tools		
	Sys. Prog. Tools		
	355		
	BOS		
	Salvager		
	Ring Zero		
	Ring One		
	<input checked="" type="checkbox"/> SysDaemon/Admin.		
	Runtime		
	User Cmnd/Subr.		
Objections/Comments:			

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

SUMMARY:

Modify answering service and associated administrative tools as needed to interface with the priority scheduler.

REASONS:

Marketing requirement.

DETAILED PROPOSAL:

As described in MTB 193, with the following extension:

It will be possible for the system administrator to specify one or more load control groups to be used by absentee jobs (with the work class being implied by the group). Each such group will be assigned to one or more of the absentee queues, and there must be one and only one group for each queue.

TITLE: Fix bug in stack_frame_exit_		STATUS	DATE
AUTHOR: J. M. Broughton		Written	07.08.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) unreported -Documented in MTB -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR		Status	A 07/15/75
		Expires	01/15/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason) PLM does not yet exist	

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

SUMMARY:

Modify stack_frame_exit_ to correctly handle the case of a frame which exits to a begin block.

REASONS:

This procedure currently assumes that a frame is exited by a normal call to a procedure. In such a case, the last location exited in the frame can be determined from the return pointer. If a frame "calls" a begin block, the return pointer is not set, and as a result may designate the location at which the last call was made or may be totally undefined. This causes confusing information to be returned by default_error_handler_, trace_stack_, etc.

DETAILED PROPOSAL:

Modify stack_frame_exit_ to detect the case described above, and return, as the location at which the frame was exited, a pointer to the location preceding the begin block. Also add a flag to the situation structure returned to indicate the case.

Multics Change Request

TITLE: Improvements to interpret_op_ptr_		STATUS	DATE	
AUTHOR: J. M. Broughton		Written	07.08.75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)	Status	A 07/15/75	
		Lib. Maint. Tools	Expires	01/15/76
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
		355		
	BOS	MPM (Vol, Sect.)		
	Salvager	PLMS (AN #)		
	Ring Zero	MOSN (Sect.)		
	Ring One	MPAM (Sect.)		
	SysDaemon/Admin.	MSAM (Sect.)		
	<input checked="" type="checkbox"/> Runtime	Info Segs		
	User Cmnd/Subr.	Other (Name)		
Objections/Comments:		None (Reason)	no change	

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

SUMMARY:

1. Make various changes to enable interpret_op_ptr_ to detect additional forms of calls to operators and return better information. New cases to be handled; begin block operators, trace entry operators, entry variable call operators, basic operators.
2. Remove explicit calls to hcs_\$make_ptr to get address of segdef's in pll_operators_.

REASONS:

1. To provide better information when faults occur while executing in these operators, or when calls are made out through these operators.
2. To improve performance slightly.

DETAILED PROPOSAL:

1. Add additional segdef's to bracket sections of code in pll_operators_ to detect new cases, and move begin block operators to same section as entry operators.
2. Add segdef's to basic_operators_ to determine what action(s) are being performed. place call operator return point in fixed location in stack for standard and extended precision basic and recompile basic_runtime_(s) to reflect change.
3. Change interpret_op_ptr_ to recognize new cases.

Multics Change Request

TITLE: Fix bugs in get_link_ptr_		STATUS	DATE	
AUTHOR: J. M. Broughton		Written	07.08.75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) <u>unreported</u> -Documented in MTB -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)	Status	A 07/15/75	
		Lib. Maint. Tools	Expires	01/15/76
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
		355	MPM (Vol, Sect.)	
	BOS	PLMS (AN #)		
	Salvager	MOSN (Sect.)		
	Ring Zero	MPAM (Sect.)		
	Ring One	MSAM (Sect.)		
	SysDaemon/Admin.			
	X Runtime			
	User Cmmnd/Subr.			
Objections/Comments:		Info Segs		
		Other (Name)		
		None (Reason) no change		

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

SUMMARY:

Modify get_link_ptr_ to fix two bugs:

1. a fault when called to find a link reference near the beginning of an object segment.
2. failure to recognize the compiler identifier "PL/I" in addition to "v2pll".

REASONS:

The procedure get_link_ptr_ is called to find a call through a link closely preceding some specified location and return the name of the routine called as specified by the link.

Bug (1) is caused by the procedure's looking back for a link reference farther than there is object segment to look at.

Bug (2) prevents the procedure from recognizing calls from recently compiled PL/I programs, as the compiler of an object segment is used to select the code sequences searched for.

DETAILED PROPOSAL:

Make changes indicated to fix bugs. Incidental change: convert to use version 2 object_info structure.

Multics Change Request

TITLE: Fix meter gate and spg to work with combined linkage for gates		STATUS	DATE
AUTHOR: S. Webber		Written	07.09.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> AIM <input type="checkbox"/> other- explain in DETAILED PROPOSAL		Status	A 07/15/75
		Expires	01/15/76
-Planned for System MR		DOCUMENTATION CHANGES	
-Fixes Bug Number(s)		Document	Specify One or More
-Documented in MTB		MPM (Vol, Sect.)	
-User/Operations-visible		PLMS (AN #) 52	
Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no		MOSN (Sect.)	
-Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		MPAM (Sect.)	
-Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same		MSAM (Sect.)	
<input type="checkbox"/> Worse		Info Segs	
-Replaces MCR		Other (Name)	
Objections/Comments:		None (Reason)	

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

SUMMARY:

1. Fix meter_gate to find gate linkage sections by looking at the lot instead of assuming foo.link.
2. Same for spg.
3. Add reset feature to meter_gate.

REASONS:

1. Allows us to combine gate linkage sections.
2. Reset feature is very useful.

IMPLICATIONS:

ring_zero_meter_limits_ASCII should be changed to allow "lot" and "active_sup_linkage" to be readable (if meter_gate is to continue to be a general tool).

meter_gate

meter_gate

Name: meter_gate, mg

This command is used to interpret and print per-system metering information for entries in specified hardcore gates.

Usage

meter_gate gate_name -control_arg^S_A -entryname-

where:

1. gate_name is the name of the gate segment to be examined; i.e., hcs_, phcs_, etc.
2. control_arg^S can be selected from the following list.
 - time, -tm causes the output to be sorted on the total time spent in each entry.
 - average, -av causes the output to be sorted on the average time spent in each entry.
 - call, -cl causes the output to be sorted on total calls to each entry.
 - page, -pg causes the output to be sorted on the average number of page faults in each entry.

~~insert~~
If ^{none of the above} a control_arg^S is not specified, the output is not sorted.
3. entryname is the name of a single entry in the specified gate. Only the information for that entry is printed. If entryname is not specified, information for all entries is printed.

Notes

The output header consists of the time ^{since the last reset (or if no reset has ever been given that)} the system was brought up, the current time and the total charge time ~~(total_opa_time - idle_time)~~. Also printed is the total number of calls to the gate, the amount of time spent in the entries that were called and the percentage of total charged time that was spent in the entries that were called.

meter_gate

meter_gate

The following is a brief description of the variables printed out by meter_gate.

<u>Item</u>	<u>Meaning</u>
calls	is the total number of times the gate entry point was called.
pcent	is the percentage of total charge time spent in the called segment.
avg	is the average virtual time in milliseconds spent in the called segment.
pfault	is the average number of page faults incurred during a call to a segment through the specified entry.
entry name	is the name of an entry point to the gate.

Entries not called during the reporting interval are not printed.

*** INSERT

-report_reset, -rr performs the reset function after printing the report

-reset, -rs performs the reset function and suppresses printing the report.

TITLE: Update get_entry_arg_descs_		STATUS	DATE
AUTHOR: M. Weaver		Written	07.09.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB 187 -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 07/15/76
		Expires	01/15/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	79
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Update get_entry_arg_descs_ to use version 2 object_info structure and to handle the new entry sequence changes. Add a new entry point, \$text_only (same arguments), that will return descriptors only for entry sequences that are described entirely in the text.

REASONS:

Entry parameter descriptor pointers are being moved from the definitions to the text section. The new entry point is needed by the command processor, which doesn't want the overhead of looking at definitions. Even if much of the code is distinct from that of the main entry point, it will be good to have the parameter descriptor finding code in a single module.

DETAILED PROPOSAL:

Keep the entry validation code in the main entry point. This means that even if the "entry" pointer points to a new style entry sequence, object_info_ will still be called and the definition checked.

The entry \$text_only will not validate the entry but just check the flags at addrel (entry_ptr, -1). This should be safe since \$text_only will generally be called with the output of hcs_\$make_ptr.

TITLE: Remove 8-page slew at end of salvage		STATUS	DATE	
AUTHOR: B. May (Phx)		Written	07.09.75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> AIM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status	
		Lib. Maint. Tools	Expires	
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
		355		
		BOS		
		<input checked="" type="checkbox"/> Salvager	MPM (Vol, Sect.)	
		Ring Zero	PLMS (AN #)	
		Ring One	MOSN (Sect.)	
		SysDaemon/Admin.	MPAM (Sect.)	
	Runtime	MSAM (Sect.)		
	User Cmmnd/Subr.			
Objections/Comments:		Info Segs		
		Other (Name)		
		None (Reason)		

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

SUMMARY:

Currently, the salvager uses an 8-cycle do loop to slew the salvage report completely into the stacker. This should be changed to slew just once.

REASONS:

For most salvager runs, immediate access to this report is not necessary. Thus, the 8-cycle slew is unnecessarily wasteful of paper.

A secondary reason arose from the use of lightweight paper on the PRM1200/1600 printers. The printer stacker mechanism very frequently jams the paper for the long slew.

IMPLICATIONS:

Much paper will be saved.

Operators must manually skip out the paper when it is necessary to examine the salvager output.

This change was discussed with Andy Kobziar. He has no objections to the passage of this MCR.