

TO: Distribution

FROM: Ross E. Klinger

DATE: December 3, 1973

SUBJECT: Proposed new commands

Attached are the MPM write-ups and sample output for two new commands: snoop (sn), and print_snoop_data (psd). These commands provide the means by which the segment references of large, multi-procedure processes can be metered. They are an aid in determining the flow of execution within such a process, and more important, in which procedures a process is spending its time. The systems programmer and subsystem designer ought to benefit especially from locating the "hotspots" in a process. In particular, the Initializer and IO Daemons are likely candidates for "snooping".

The snoop and psd commands augment the tracing facilities provided by the commands trace and page_trace, and the PL/I compile option -profile. The latter provides too fine a trace resolution for effective per process metering: trace is more of a debugging than metering tool; and page_trace cannot directly, or completely, meter the actual flow of a process.

- 1) Segment references are detected by snoop irrespective of page or segment faults. References to wired pages, or to pages which are largely core-resident due to frequent usage, can therefore be detected.
- 2) snoop uses a virtual CPU timer to control its sampling, so that actual non-ring 0 execution time is metered.
- 3) snoop provides for up to 32,767 samples, or approximately 9 hours of non-ring 0 CPU time at a sample rate of 1 second. page_trace is limited to approximately 350 samples.

The following are planned additions to print_snoop_data:

- brief to suppress the detailed trace.
- ignore s1,s2.... to exclude the specified segment numbers s1, s2, etc. when generating the histograms. Segments such as pl1_operators_ are so frequently referenced that they tend to unbalance the histogram resolution.

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

A more extensive modification could eventually provide finer details of a segment reference, such as which page of a segment was referenced. snoop can currently resolve down to an offset within a component of a bound segment.

If you wish to try these commands, they currently reside in >udd>pdo>Klinger>public . In any case, I welcome your comments and suggestions. Send written comments to:

Ross E. Klinger
MIT Room 39-464

or mail comments to:

Klinger.P00

```

-----
| snoop |
-----

```

12/03/73
Page 1

Name: snoop, sn

The snoop command periodically samples the machine registers in order to determine which segments a process is referencing. Three output segments are produced, which are interpretable by the print_snoop_data command. (See the MPM write-up of the print_snoop_data command.)

Usage snoop -control_group-

Control arguments may be chosen from one of the following two control groups:

1) arguments which initiate sampling

-time n specifies the rate in milliseconds at which
-tm n the process will be sampled. n must be a
positive integer. The default is n = 1000;
i.e., the process will be sampled once every
second.

-segment name specifies the names to be given the three
-sm name output segments. name may be either an
absolute or relative pathname. name may end
with the suffix .snoop ; if it does not,
.snoop will be appended. The output segments
will be named as follows:

```

(entry portion of) name.snoop
(entry portion of) name.snoopx
(entry portion of) name.snoope

```

If name is an absolute pathname, the output segments will be placed in the appropriate directory. If name is a relative pathname, the output segments will be placed in the user's working directory. The default will cause the output segments to be placed in the user's working directory, with entry names as follows:

```

mm/dd/yy__hhmm.m_zzz_www.snoop
mm/dd/yy__hhmm.m_zzz_www.snoopx
mm/dd/yy__hhmm.m_zzz_www.snoope

```

snoop

(See the MPM write-up of the date_time_ subroutine for an explanation of the above code.)

2) the argument which terminates sampling

-reset specifies that the process is no longer to be sampled.
-rs

Notes

The machine registers can be sampled only when the process is running in a ring other than ring 0. Were a process to use, for example, a total of 100 seconds of processor time, and snoop, running at a sample rate of $n = 1000$, were to record only 23 samples, it would indicate that 77 seconds of processor time were spent in ring 0.

Under certain conditions, the contents of one of the machine registers sampled - the Temporary Segment Register (TSR) - may be invalid. This invalidity will be noted, but does not necessarily indicate that the process is in error.

At the maximum sample rate, 1 millisecond, execution time may be increased by as much as 50%. Using a 1 second sample rate, the increase in execution time is negligible.

If the process being sampled should be terminated without an invocation of snoop with the -reset option, interpretable output segments will still be produced; however, both the off-time and the last recorded sample will be invalid.

Command
12/03/73

Name: print_snoop_data, psd

The print_snoop_data command interprets the three data segments produced by the snoop command, and produces a printable output segment which contains the following information: a detailed trace of segment references; a segment number to pathname dictionary; and histograms of the Procedure Segment Register (PSR) and Temporary Segment Register (TSR) segment reference distributions. (See the MPM write-up of the snoop command.)

Usage print_snoop_data name

name specifies the names of the data segments to be interpreted, as well as the name of the output segment to be produced. name may be either an absolute or relative pathname. name may end with the suffix .snoop; if it does not, .snoop will be appended.

If name is an absolute pathname, the specified directory will be searched for three segments with entry names as follows:

(entry portion of) name.snoop
(entry portion of) name.snoobox
(entry portion of) name.snooobe

The output segment will be placed in the same directory with the entry name:

(entry portion of) name.snooop

If name is a relative pathname, the user's working directory will be searched for the data segments, and the output segment will be placed in the working directory, named as above.

Notes

1) print_snoop_data is able to detect a re-used segment number. The appearance of a parenthesized integer preceding a segment number indicates such a re-usage.

```
|-----|
|  print_snoop_data  |
|-----|
```

Page 2

```
23416542 >udd>user>bound_alpha_16542
(1) 23412104 >udd>user>max351512
(2) 23416160 >system_library_languages>assign_16160
```

The appearance of three such lines in the detailed trace would imply the following:

a) a reference was made to location 6542 in bound_alpha_ . The particular component of bound_alpha_ being referenced could not be determined. bound_alpha_ was assigned segment number 234.

b) a reference was made to location 512 in max35 . max35 is a component of a bound segment whose name can be determined from the segment number to pathname dictionary. The segment bound_alpha_ has been terminated, and when the segment of which max35 is a component was initiated, it was assigned segment number 234.

c) a reference was made to location 6160 in assign_ . The segment of which max35 is a component has been terminated, and when assign_ was initiated, it was assigned segment number 234.

2) The appearance of a segment number suffix (i.e., .1 , .2 , etc.) indicates a component of a bound segment.

```
310 >system_library_standard>bound_tl_term_
310.1 tssi_
310.2 translator_info_
```

The appearance of the above lines in the segment number to pathname dictionary would indicate that tssi_ was the first component of bound_tl_term_ to be referenced, and that translator_info_ was the second component of bound_tl_term_ to be referenced.

1223	PS		23013164	>system_library_1>bound_sss_wire	3164
1223	TSR		211112	>112	
1224	PSR		22412774	>system_library_1>pll_operators_12774	
1224	TSR	(1)	32413235	>process_dir_dir>!BHQ8HDgb888888>cg_scratch_2_13235	
1225	PSR		23013204	>system_library_1>bound_sss_wired_13204	
1225	TSR		23013203	>system_library_1>bound_sss_wired_13203	
1226	PSR		23013206	>system_library_1>bound_sss_wired_13206	
1226	TSR		23013203	>system_library_1>bound_sss_wired_13203	
1227	PSR	(2)	31116642	create164	
1227	TSR		222113751	>process_dir_dir>!BHQ8HDgb888888>combined_linkage_4.00113751	
1228	PSR	(2)	31116750	create172	
1228	TSR		4012346	dir_seg12346	
1229	PSR		23013206	>system_library_1>bound_sss_wired_13206	
1229	TSR		40112	dir_seg112	
1230	PSR	(2)	31116772	create1214	
1230	TSR	(2)	31117151	create1373	
1231	PSR		23013204	>system_library_1>bound_sss_wired_13204	
1231	TSR		23013203	>system_library_1>bound_sss_wired_13203	
1232	PSR		25316160	11st1410	
1232	TSR		25316156	11st1406	
1233	PSR		25316222	11st1452	
1233	TSR		25316221	11st1451	
1234	PSR		23013204	>system_library_1>bound_sss_wired_13204	
1234	TSR		23013203	>system_library_1>bound_sss_wired_13203	
1235	PSR		2531112	delete1112	
1235	TSR		231116771	>user_dir_dir116771	
1236	PSR		22417076	>system_library_1>pll_operators_17076	
1236	TSR		23013203	>system_library_1>bound_sss_wired_13203	
1237	PSR		25311054	delete11054	
1237	TSR		25311051	delete11051	

SAMPLE OUTPUT

DETAILED TRACE

300.9 c_a
 300.10 expmac
 300.11 macro_table
 300.12 adjust_ref_count
 300.13 instruction_info
 300.14 compile_statement
 300.15 compile_tree
 300.16 prepare_operand
 300.17 load
 300.18 m_a
 300.19 jump_op
 300.20 call_op
 300.21 stack_temp
 300.22 base_man
 300.23 compile_exp
 300.24 set_indicators
 300.25 move_data
 300.26 generate_constant
 300.27 eval_exp
 300.28 pointer_builtins
 300.29 need_temp
 301 >system_library_1>sit
 302 >system_library_1>name_table
 303 >system_library_auth_maint>bound_bllp_
 303.1 olip
 303.2 general_ready
 304 >system_library_standard>bound_io_commands_
 304.1 read_llst_
 304.2 file_output
 304.3 dprint
 304.4 dprint_
 305 >system_library_standard>bound_conversion_rtns_
 305.1 convert_date_to_binary_
 306 INVALID NON-RING 0 SEGMENT NUMBER
 (1) 306 >system_library_languages>pl1_version_
 307 INVALID NON-RING 0 SEGMENT NUMBER
 (1) 307 >system_library_languages>pl1_error_messages_
 310 >system_library_standard>bound_fl_term_
 310.1 fssl_
 310.2 translator_info_
 310.3 find_include_file_
 310.4 term_
 311 INVALID NON-RING 0 SEGMENT NUMBER
 (1) 311 >user_dir_dir>P00>Kllinger>snoopwork>psd.llst
 (2) 311 >system_library_standard>bound_fscom2_
 311.1 create
 311.2 copy
 311.3 copy_seg_
 311.4 equal_
 311.5 rename
 312 INVALID NON-RING 0 SEGMENT NUMBER
 (1) 312 >library_dir_dir
 313 >library_dir_dir>include
 314 >library_dir_dir>include>object_info.incl.pl1
 (1) 314 >system_library_standard>bound_access_commands_
 314.1 acl_commands
 314.2 check_path_name_
 314.3 setcac1
 315 >library_dir_dir>include>component_info.incl.pl1

SAMPLE OUTPUT

DICTIONARY

PSR REFERENCE HISTOGRAM
(Resolution factor = 5)

SEGMENT

217.1 xx
217.2 xx
217.3 x
217.4 x
217.8 x
217.9 x
223 xxxxxxx
224 xxx
227.1 x
230 xxx
243.1 xxx
243.2 x
253.1 x
253.2 x
253.3 x
253.6 xx
253.7 x
253.8 x
254.1 x
255.1 x
255.3 x
256 x
272.1 xxxxxx
273.1 xx
273.2 x
273.3 x
273.4 xxxxxxxxxxxxxxx
273.5 xx
273.6 x
273.7 x
273.8 x
273.9 x
273.10 x
273.11 x
273.12 x
273.14 xx
274.1 x
274.2 x
274.4 xxxxxxx
274.5 x
274.6 xx
274.7 x
274.8 x
274.9 xxx
274.10 xx
274.12 xxxxxxx
274.13 x
274.14 x
274.15 x
274.16 x
274.17 x
274.18 xxxxxxx
274.19 x
274.20 xx

9

SAMPLE OUTPUT

REFERENCE HISTOGRAM