

Inter-Office Memorandum

To	Mesa Users	Date	October 27, 1980
From	Jim Sandman	Location	Palo Alto
Subject	Control Transfer Counting Tool	Organization	SDD/SS/Mesa

XEROX

Filed on: [Iris]<Mesa>Doc>XferCounter.bravo (and .press)

A tool for studying the behavior of Mesa programs is described below. It counts the number of control transfers (**XFERs**) to a module and records the time spent executing in a module. It can also be used to gather information on the flow of control between groups of modules. An **XFER** is the general control transfer mechanism in Mesa. The following are all **XFERs**: procedure call, return from a procedure, traps, and process switches.

The system is implemented as a set of commands that can be executed from the Mesa Debugger, a routine that intercepts all **XFERs** and collects statistics about them, and a routine that intercepts conditional breakpoints for turning the **XFER** monitoring on and off. Existing Debugger commands are used to specify where **XFER** monitoring is enabled, and additional commands are provided for controlling the counting of **XFERs** and outputting the results.

This tool is intended to provide a global view of the behavior of a system. With this tool, a user can identify modules that warrant closer study with other tools such as the Performance Monitor.

Components

CountTool is the component of the tool that is loaded with client programs built on top of Alto/Mesa. This configuration contains one module: **Counter**; it contains the **XFER** trap handler and a breakpoint handler. **CountTool** must be loaded and started in the system it will monitor. This may be done by including **CountTool** in the client configuration whose control module imports and starts **XferCountDefs.Counter** or by executing the following command in the Alto Executive:

```
>Mesa CountTool Client
```

CountPackage is the component that is loaded into the Mesa Debugger. It implements the basic commands required to enable **XFER** monitoring and to output measurement results.

CountPackage must be loaded into the Debugger before its commands can be executed. The easiest way is to load it when installing the Debugger by executing the following command in the Alto Executive:

```
>XDebug CountPackage/1
```

The **CountPackage** creates a window through which all interaction with the tool takes place.

Operation

There are two modes of operation, plain and matrix. Plain mode (the default) simply records the time spend in a module and the number of **XFERs** to that module. Matrix mode is used to gather information on flow of control between groups of modules. Each module is a member of one of sixteen groups. A sixteen by sixteen matrix of counts and times is maintained by the Counter. The rows of the matrix are the groups of the source of the **XFER**, the `from` group. The columns of the matrix are the groups of the destination of the **XFER**, the `to` group.

In plain mode when **XFER** monitoring is enabled and a **XFER** occurs, the trap handler calculates the time since the last **XFER** and adds that to the cumulative time for the current module. It then calculates which module is the destination of the **XFER** and makes that the current module, incrementing its count. In matrix mode when **XFER** monitoring is enabled and a **XFER** occurs, the trap handler updates the appropriate element of the matrix. In both modes, the **XFER** handler then completes the **XFER**, and the client program continues.

The state of **XFER** monitoring can be controlled by two methods. The first is by setting a conditional break to be handled by the tool's breakpoint handler. The second is by calling the procedures **XferCountDefs.StartCounting** and **XferCountDefs.StopCounting**.

When the break handler intercepts a breakpoint, it checks to see if the breakpoint is conditional. If not, the break handler proceeds to the Debugger. If it is, the state of **XFER** monitoring is changed and program execution is resumed. A condition of zero turns on **XFER** monitoring; a condition of one toggles the state of **XFER** monitoring; a condition of two turns off **XFER** monitoring. Any other condition has no effect.

The procedures **XferCountDefs.StartCounting** and **XferCountDefs.StopCounting** provide an alternative method of enabling **XFER** monitoring. These procedures may be called from statements in the client program, or they may be called from the Debugger's interpreter.

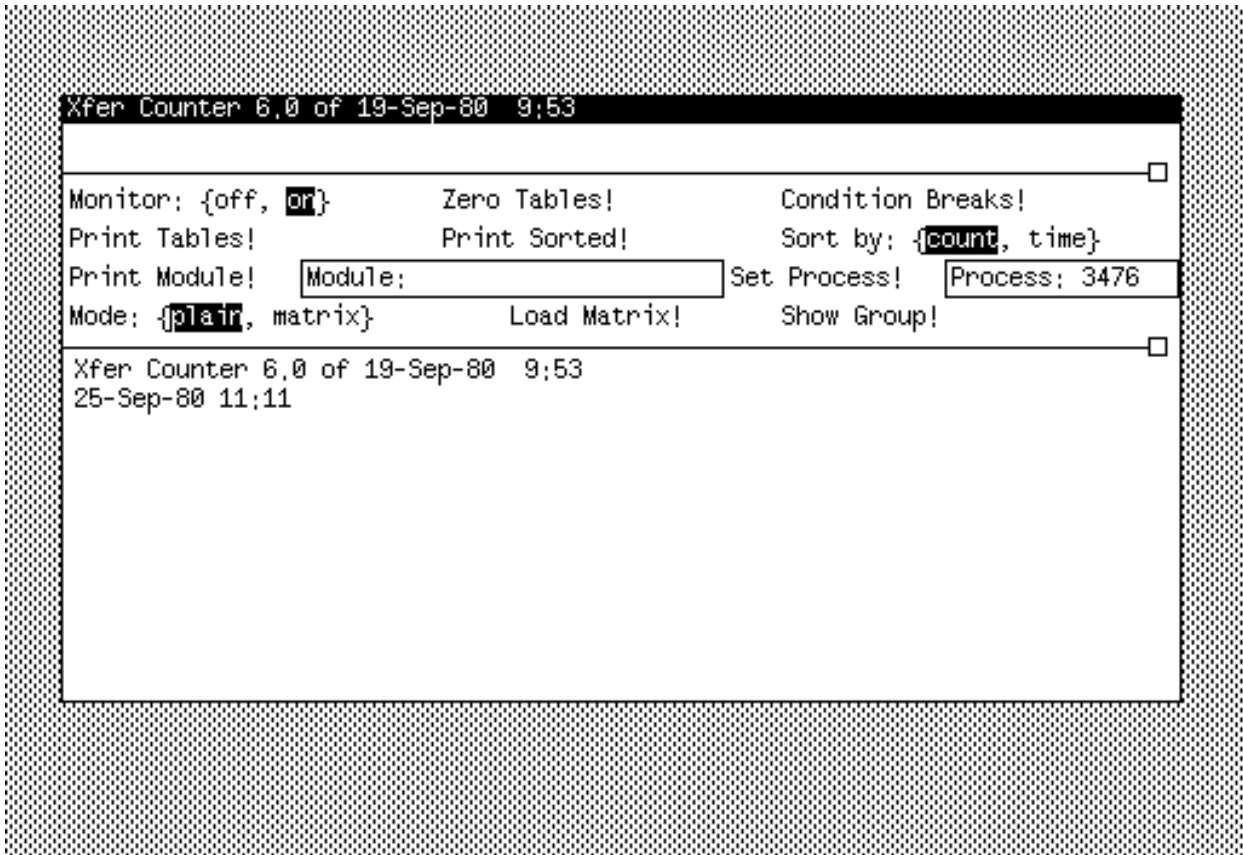
Since multiple processes may interact with each other, there is the concept of the *tracked process*. If the tracked process is not NIL, only those **XFERs** that are encountered during execution of the tracked process are counted; all others are simply resumed. If the tracked process is NIL, then all processes are tracked.

The group information for matrix mode is entered into the Counter by reading an edited version of the output from the debugger's `Display GlobalFrameTable` command. Appending the group number to the line for a modules will assign the module to that group. If no group number is specified, the module is assigned to the group of the previous line. Modules not assigned to any group are members of group zero. For example:

```
StringsB, G:173134B, gfi:33B 1      group 1
StringsA, G:173140B, gfi:32B      group 1
StreamsC, G:173144B, gfi:31B 2    group 2
StreamsB, G:173150B, gfi:30B      group 2
StreamsA, G:173154B, gfi:27B      group 2
SegmentsB, G:173160B, gfi:26B 3   group 3
SegmentsA, G:173164B, gfi:25B     group 3
OurProcess, G:173170B, gfi:24B 4   group 4
NonResident, G:173210B, gfi:23B   group 4
Modules, G:173214B, gfi:22B       group 4
Miscellaneous, G:173220B, gfi:21B group 4
MesaInit, G:173224B, gfi:20B 0    group 0
MesaDebug, G:173234B, gfi:17B    group 0
```

Window and Commands

Interaction with the **CountPackage** is done through its window. There are three subwindows: the message subwindow, the form subwindow, and the log subwindow. Error messages and warnings are displayed in the message subwindow. Commands are invoked in the form subwindow. All output is displayed in the log subwindow and written on `Count.log`. An illustration of the window during a sample session is shown below.



Monitor: {off, on}

turns off/on the tools breakpoint handler. All conditional breakpoints will affect the state of **XFER** monitoring when the monitor is on, and will behave as normal conditional breakpoints when it is off.

Zero Tables!

zeros out all counts and times.

Condition Breaks!

makes all non-conditional breakpoints conditional by adding the condition "1" to them.

Print Tables!

displays all the statistics for each module in order of increasing global frame table index (gfi) for plain mode. In matrix mode, it displays the statistics for each nonzero element of the matrix. The output format of times is `sec.msec:usec`. This command may be aborted by typing **^DEL**.

Print Sorted!

displays all the statistics for each module in order of decreasing time or decreasing number of **XFERs** depending on the value of `Sort by`. This command may be aborted by typing **^DEL**. Not allowed in matrix mode.

Sort by: {count, time}

when set to `count`, the `Print Sorted` command displays table entries in order of decreasing number of **XFERs**, otherwise it displays in order of decreasing time.

Print Module!

displays the statistics for the module specified by `Module`. Not allowed in matrix mode.

Module:

specifys the module to the `Print Module` command. It is either the module's global frame table index (gfi), its global frame address (g), or its module name (if the current configuration contains the desired module).

Set Process!

specifies that only those **XFERs** executed by the specified process are to be counted. The default case is to track all processes.

Process:

used by the `Set Process` command. It contains an octal `ProcessHandle` as obtained from the Debugger's `List Processes` command. If `Process` is empty, all processes are tracked.

Mode: {plain, matrix}

when set to `plain` (default), the Counter records transfers between modules. When set to `matrix`, the Counter records transfers from one group to another.

Load Matrix!

using the current selection as a file name, reads the file to input group information.

Show Group!

using the current selection as a group number, prints the names of the modules belonging to that group. This command may be aborted by typing **^DEL**.

Limitations

1. Execution speed: Xfer monitoring slows down the executions of a program considerably, since extra processing is done on every **XFER**. As a result, interrupt processes that are triggered by clocks (e.g., the keyboard process) will run relatively more frequently.
2. Idle loop accounting: When no process is running, the Mesa Emulator runs in its idle loop waiting for a process to become ready. This idle time is charged to the process that was last running.
3. Time base: The time base available on the Alto is a twenty-six bit counter, where the basic unit of time is 38 microseconds. Thus the counter turns over every 40 minutes, and no individual time greater than 40 minutes is meaningful. Total times are thirty-two bit numbers and will overflow after 340 minutes.
4. Overhead calculation: Due to implementation restrictions and timer granularity, some of the overhead of processing an **XFER** trap is incorrectly assigned to the client program instead of the **CountTool**. As a result, times must be interpreted as only a relative measure of the time spent in a module.
5. Counter sizes: Counts are thirty-two bit numbers. The maximum total count is 4,294,967,295 **XFERs**!
6. Memory requirements: The **CountTool** requires seven pages of resident memory: two for its code and five for its frames and tables. This may affect the performance of some systems that use a lot of memory, especially on the Alto.
7. Worry mode: The **CountTool** operates in worry mode; as a consequence, you may find that you cannot `Quit` from the Debugger after your session. Use the `Kill` command instead.

Getting Started

The steps required for using the count tool are outlined below.

1. obtain the `.bcd` files for **CountTool** and **CountPackage**.
2. install the **CountPackage** in the Debugger.
3. start your program with the **CountTool** included.
4. enter the Debugger and set conditional breakpoints to enable monitoring as desired.
5. turn the break handler on by setting the `Monitor` parameter to `on`.
6. proceed with program execution.
7. return to the Debugger via an interrupt or an unconditional breakpoint.
8. display results with the `Print` commands.

Sample Session

The following annotated listing of Debug.log and Count.log should give a fair idea of the use of the count tool. It counts the **XFERS** executed when loading a module.

From Debug.log:

```
Alto/Mesa Debugger 6.0m of 5-Sep-80 12:02
25-Sep-80 9:56
```

You called?

```
>Set Root configuration: MesaExec
-- set breakpoints to count XFERS involved with loading.
>Set Module context: MesaExec
>Break Entry procedure: LoadNew Breakpoint #1.
>Break Xit procedure: LoadNew Breakpoint #2.
-- the Condition Breaks command will make these conditional.
-- Condition Breaks and set the process.
>Proceed [Confirm]
You called?
-- look at the results.
```

```
>--Test.map -- file containing group information.
-- set mode to matrix and load group information using Load Matrix command.
>Proceed [Confirm]
You called?
-- look at the matrix.
>Kill session [Confirm]
```

From Count.log:

```
Xfer Counter 6.0 of 19-Sep-80 9:53
25-Sep-80 10:48
```

```
Track process: 3647B -- ignore keyboard and interrupt key
Conditionalized breaks
```

```
-- Output of Print Tables command with mode = plain
```

```
Total Xfers      4,088
Total Time       1.329:842
```

Gfi	Frame	Module	#Xfers	%Xfers	Time	%Time
1B	174164B	Resident	12	.29	6:286	.47
3B	174030B	DiskIO	869	21.25	583:996	43.91
4B	174000B	Swapper	530	12.96	96:050	7.22
5B	173344B	MDSRegion	538	13.16	200:367	15.06
7B	173314B	BFS	1	.02	76	.00
10B	173304B	Directory	80	1.95	38:900	2.92
11B	173270B	DiskKD	2	.04	533	.04
13B	173260B	Files	140	3.42	15:392	1.15
15B	173254B	FSP	100	2.44	15:468	1.16
16B	173240B	LoadState	97	2.37	105:384	7.92
22B	173214B	Modules	96	2.34	35:471	2.66
23B	173210B	NonResident	3	.07	457	.03
25B	173164B	SegmentsA	74	1.81	17:945	1.34
26B	173160B	SegmentsB	89	2.17	13:487	1.01
27B	173154B	StreamsA	55	1.34	10:629	.79
30B	173150B	StreamsB	44	1.07	8:115	.61

Control Transfer Counting Tool

31B	173144B	StreamsC	55	1.34	10:629	.79
32B	173140B	StringsA	89	2.17	9:525	.71
33B	173134B	StringsB	19	.46	3:924	.29
35B	173124B	AlFont	176	4.30	21:107	1.58
36B	173104B	AltoLoader	237	5.79	29:070	2.18
40B	173100B	BcdOperations	153	3.74	16:421	1.23
45B	172720B	LoaderCore	516	12.62	74:256	5.58
47B	172660B	StreamIO	23	.56	2:057	.15
50B	172650B	SystemDisplay	82	2.00	13:830	1.03
54B	170274B	MesaExec	8	.19	457	.03
Ignored Xfers					98	
Ignored Time					309:943	

-- Output of Print Sorted command with Sorted by = count

Total Xfers		4,088					
Total Time		1.329:842					
Gfi	Frame	Module	#Xfers	%Xfers	Time	%Time	

3B	174030B	DiskIO	869	21.25	583:996	43.91	
5B	173344B	MDSRegion	538	13.16	200:367	15.06	
4B	174000B	Swapper	530	12.96	96:050	7.22	
45B	172720B	LoaderCore	516	12.62	74:256	5.58	
36B	173104B	AltoLoader	237	5.79	29:070	2.18	
35B	173124B	AlFont	176	4.30	21:107	1.58	
40B	173100B	BcdOperations	153	3.74	16:421	1.23	
13B	173260B	Files	140	3.42	15:392	1.15	
15B	173254B	FSP	100	2.44	15:468	1.16	
16B	173240B	LoadState	97	2.37	105:384	7.92	
22B	173214B	Modules	96	2.34	35:471	2.66	
26B	173160B	SegmentsB	89	2.17	13:487	1.01	
32B	173140B	StringsA	89	2.17	9:525	.71	
50B	172650B	SystemDisplay	82	2.00	13:830	1.03	
10B	173304B	Directory	80	1.95	38:900	2.92	
25B	173164B	SegmentsA	74	1.81	17:945	1.34	
27B	173154B	StreamsA	55	1.34	10:629	.79	
31B	173144B	StreamsC	55	1.34	10:629	.79	
30B	173150B	StreamsB	44	1.07	8:115	.61	
47B	172660B	StreamIO	23	.56	2:057	.15	
33B	173134B	StringsB	19	.46	3:924	.29	
1B	174164B	Resident	12	.29	6:286	.47	
54B	170274B	MesaExec	8	.19	457	.03	
23B	173210B	NonResident	3	.07	457	.03	
11B	173270B	DiskKD	2	.04	533	.04	
7B	173314B	BFS	1	.02	76	.00	
Ignored Xfers					98		
Ignored Time					309:943		

-- Output of Print Sorted command with Sorted by = time

Total Xfers		4,088					
Total Time		1.329:842					
Gfi	Frame	Module	#Xfers	%Xfers	Time	%Time	

3B	174030B	DiskIO	869	21.25	583:996	43.91	
5B	173344B	MDSRegion	538	13.16	200:367	15.06	
16B	173240B	LoadState	97	2.37	105:384	7.92	
4B	174000B	Swapper	530	12.96	96:050	7.22	
45B	172720B	LoaderCore	516	12.62	74:256	5.58	

10B 173304B Directory	80	1.95	38:900	2.92
22B 173214B Modules	96	2.34	35:471	2.66
36B 173104B AltoLoader	237	5.79	29:070	2.18
35B 173124B AlFont	176	4.30	21:107	1.58
25B 173164B SegmentsA	74	1.81	17:945	1.34
40B 173100B BcdOperations	153	3.74	16:421	1.23
15B 173254B FSP	100	2.44	15:468	1.16
13B 173260B Files	140	3.42	15:392	1.15
50B 172650B SystemDisplay	82	2.00	13:830	1.03
26B 173160B SegmentsB	89	2.17	13:487	1.01
31B 173144B StreamsC	55	1.34	10:629	.79
27B 173154B StreamsA	55	1.34	10:629	.79
32B 173140B StringsA	89	2.17	9:525	.71
30B 173150B StreamsB	44	1.07	8:115	.61
1B 174164B Resident	12	.29	6:286	.47
33B 173134B StringsB	19	.46	3:924	.29
47B 172660B StreamIO	23	.56	2:057	.15
11B 173270B DiskKD	2	.04	533	.04
23B 173210B NonResident	3	.07	457	.03
54B 170274B MesaExec	8	.19	457	.03
7B 173314B BFS	1	.02	76	.00
Ignored Xfers	98			
Ignored Time	309:943			

Matrix loaded

-- Output of Print Tables command with mode = matrix

Total Xfers	3,834				
Total Time	871:004				
From -> To	#Xfers	%Xfers	Time	%Time	
-----	-----	-----	-----	-----	
1 -> 1	2	.05	114	.01	
1 -> 2	1	.02	114	.01	
1 -> 3	4	.10	266	.03	
2 -> 1	1	.02	419	.04	
2 -> 2	542	14.13	110:756	12.71	
2 -> 4	126	3.28	17:487	2.00	
2 -> 5	87	2.26	11:201	1.28	
2 -> 6	7	.18	342	.03	
2 -> 10	53	1.38	17:221	1.97	
3 -> 1	4	.10	228	.02	
3 -> 3	179	4.66	37:719	4.33	
3 -> 4	64	1.66	5:829	.66	
3 -> 6	13	.33	2:247	.25	
3 -> 10	1	.02	38	.00	
4 -> 2	124	3.23	19:240	2.20	
4 -> 3	61	1.59	6:705	.76	
4 -> 4	1,105	28.82	264:795	30.40	
4 -> 6	87	2.26	16:725	1.92	
4 -> 7	9	.23	1:790	.20	
4 -> 9	8	.20	5:486	.62	
5 -> 2	82	2.13	13:563	1.55	
6 -> 2	7	.18	495	.05	
6 -> 3	14	.36	2:057	.23	
6 -> 4	87	2.26	27:089	3.11	
6 -> 6	207	5.39	54:330	6.23	
6 -> 7	36	.93	7:658	.87	
6 -> 10	12	.31	1:066	.12	

Control Transfer Counting Tool

9

7 -> 4	9	.23	2:552	.29
7 -> 6	37	.96	5:067	.58
7 -> 7	759	19.79	221:932	25.48
9 -> 4	11	.28	647	.07
10 -> 2	59	1.53	9:944	1.14
10 -> 3	1	.02	152	.01
10 -> 6	12	.31	3:543	.40
10 -> 10	23	.59	2:171	.24
Ignored Xfers	396			
Ignored Time	673:569			