



Debugging with Linaro DDTx



Research Computing
UNIVERSITY OF COLORADO **BOULDER**



Website: www.rc.colorado.edu

Documentation: <https://curc.readthedocs.io>

Helpdesk: rc-help@colorado.edu

Survey: <http://tinyurl.com/curc-survey18>

Meet the User Support Team



Layla
Freeborn



Brandon
Reyes



Andy
Monaghan



Michael
Schneider



John
Reiland



Dylan
Gottlieb



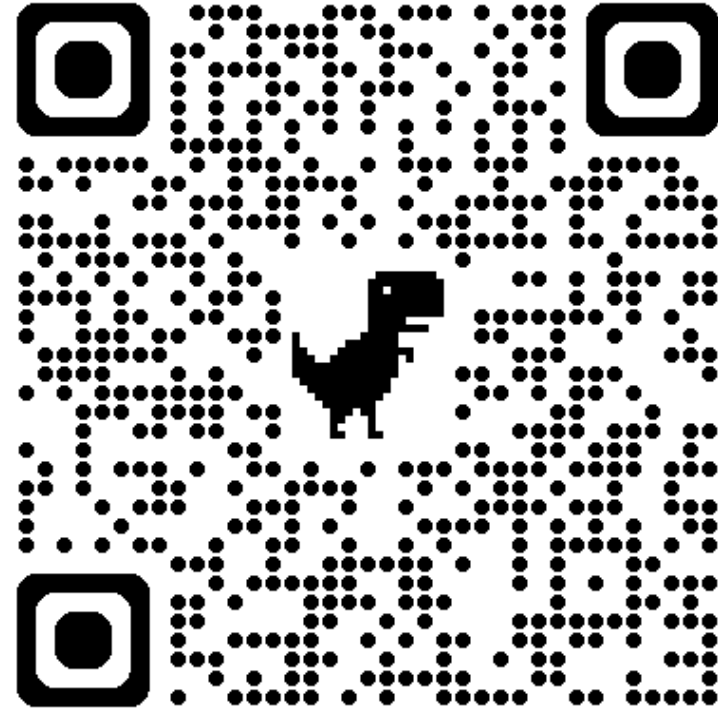
Mohal
Khandelwal



Ragan
Lee

Slides

[https://github.com/ResearchComputing/
debugging_with_linaro_ddt_rc](https://github.com/ResearchComputing/debugging_with_linaro_ddt_rc)



Debugging with VS Code

Instructor:

Research Computing

- Website: www.rc.colorado.edu
- Helpdesk: rc-help@colorado.edu

Slides:

https://github.com/ResearchComputing/debugging_with_linaro_ddt_rc

Survey:

<http://tinyurl.com/curc-survey18>

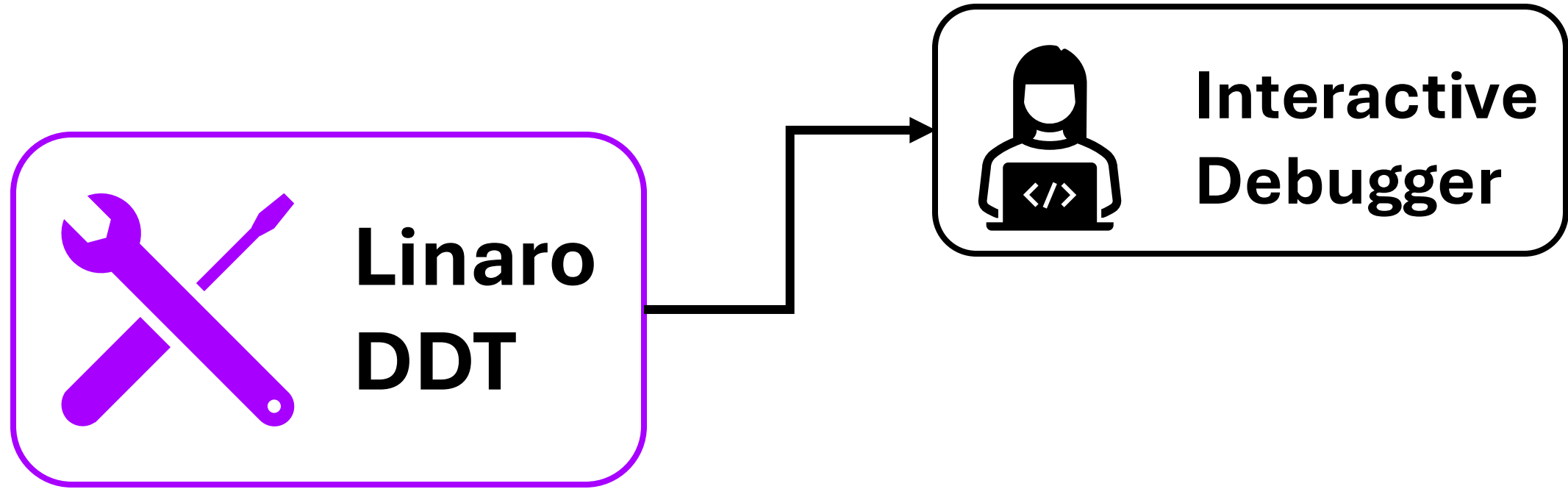


Overview

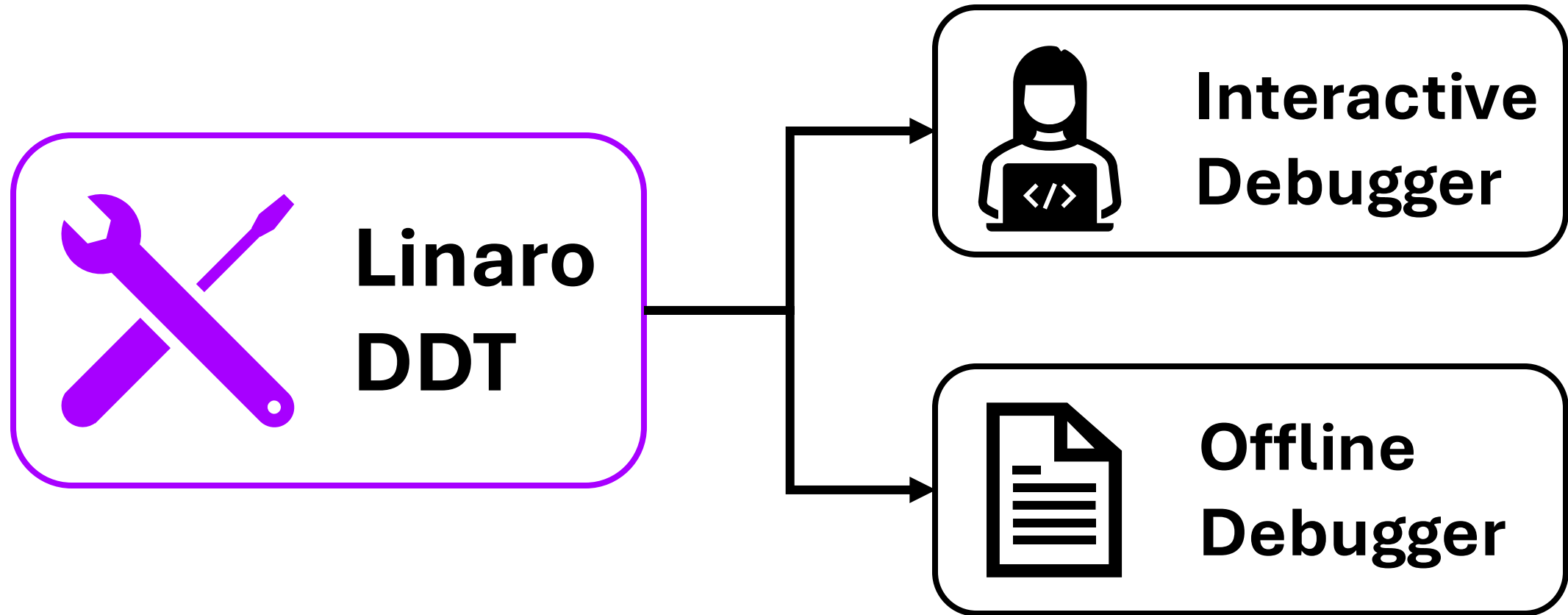


Linaro
DDT

Overview



Overview



Preparing for Linaro DDT

1. Load modules

```
ml <compiler> <mpi-library> linaroforge
```

Preparing for Linaro DDT

1. Load modules

```
ml intel/2022.1.2 impi linaroforge
```

2. Compile for debugging

```
CFLAGS = -O0 -g
```

Preparing for Linaro DDT

1. Load modules

```
ml intel/2022.1.2 impi linaroforge
```

2. Compile for debugging

```
CFLAGS = -O0 -g
```

3. Start DDT session

```
ddt -n <Num-Cores> ./mmult_c
```


Preparing for Linaro DDT

1. Load modules

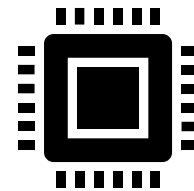
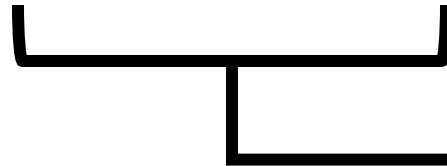
```
ml intel/2022.1.2 impi linaroforge
```

2. Compile for debugging

```
CFLAGS = -O0 -g
```

3. Start DDT session

```
ddt -n <Num-Cores> ./mmult_c
```



X16(Max)

Run

Application: /gpfs/alpine1/scratch/misc7570/examples/mmult_c Details

Application: /gpfs/alpine1/scratch/misc7570/examples/mmult_c

Arguments:

☐ stdin file:

Working Directory:

☒ **MPI:** 8 processes, Intel MPI (MPMD) Details

Number of Processes: 8

Implementation: Intel MPI (MPMD) Change...

mpiexec.hydra arguments

☐ **OpenMP** Details

☐ **CUDA** Details

☐ **ROCm** Details...

☐ **Intel Xe** Details...

☐ **Memory Debugging** Details...

☐ **Submit to Queue** Configure... Parameters...

Environment Variables: none Details

Plugins: none Details

Help Options Run Cancel

Program Location & Settings

Run

Application: /gpfs/alpine1/scratch/misc7570/examples/mmult_c Details

Application: /gpfs/alpine1/scratch/misc7570/examples/mmult_c Folder icon

Arguments: Dropdown arrow

☐ stdin file: Dropdown arrow Folder icon

Working Directory: Dropdown arrow Folder icon

☒ **MPI:** 8 processes, Intel MPI (MPMD) Details

Number of Processes: 8 Dropdown arrow

Implementation: Intel MPI (MPMD) Change...

mpiexec.hydra arguments Dropdown arrow

☐ **OpenMP** Details

☐ **CUDA** Details

☐ **ROCm** Details...

☐ **Intel Xe** Details...

☐ **Memory Debugging** Details...

☐ **Submit to Queue** Configure... Parameters...

Environment Variables: none Details

Plugins: none Details

Help Options Run Cancel

Program Location & Settings

MPI Settings

Application: /gpfs/alpine1/scratch/misc7570/examples/mmult_c Details

Application: /gpfs/alpine1/scratch/misc7570/examples/mmult_c Folder icon

Arguments: Dropdown arrow

☐ stdin file: Dropdown arrow Folder icon

Working Directory: Dropdown arrow Folder icon

☒ **MPI:** 8 processes, Intel MPI (MPMD) Details

Number of Processes: 8 Dropdown arrow

Implementation: Intel MPI (MPMD) Change...

mpiexec.hydra arguments Dropdown arrow

☐ **OpenMP** Details

☐ **CUDA** Details

☐ **ROCm** Details...

☐ **Intel Xe** Details...

☐ **Memory Debugging** Details...

☐ **Submit to Queue** Configure... Parameters...

Environment Variables: none Details

Plugins: none Details

Help Options Run Cancel

Program Location & Settings

MPI Settings

Enable Memory Debugging & Profiling

Run

Application: /gpfs/alpine1/scratch/misc7570/examples/mmult_c Details

Application: /gpfs/alpine1/scratch/misc7570/examples/mmult_c

Arguments:

☐ stdin file:

Working Directory:

☒ **MPI: 8 processes, Intel MPI (MPMD)** Details

Number of Processes: 8

Implementation: Intel MPI (MPMD) Change...

mpiexec.hydra arguments

☐ OpenMP Details

☐ CUDA Details

☐ ROCm Details...

☐ Intel Xe Details...

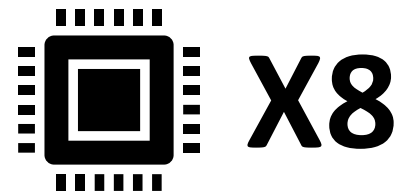
☐ Memory Debugging Details

☐ Submit to Queue Configure... Parameters

Environment Variables: none

Plugins: none

Help Options Run Cancel



Listening for your program

Waiting for all processes to be ready...

Processes connected: 8/8

Processes ready: 0/8

Cancel

Output >>

Linaro DDT

The screenshot displays the Linaro DDT (Linaro Forge 24.0.4) interface. The main window shows the source code of a C program named `mmult.c`. The code is as follows:

```
82
83
84 int main(int argc, char *argv[])
85 {
86     int mr, nproc, sz, slice;
87     double *mat_a, *mat_b, *mat_c;
88     char filename[32];
89     int remainder;
90     MPI_Status st;
91
92     MPI_Init (&argc, &argv);
93     MPI_Comm_rank(MPI_COMM_WORLD, &mr); // my rank
94     MPI_Comm_size(MPI_COMM_WORLD, &nproc); // number of processors
95
96     if (mr == 0)
97     {
98         printf(WORKED_EXAMPLE_NOTICE "\n\n");
99     }
100
101     if(argc > 3 || (argc > 2 && strcmp(argv[1], "-h") != 0) )
102     {
```

The interface includes a menu bar (File, Edit, View, Control, Tools, Window, Help), a toolbar with various icons, and a status bar at the bottom. The left sidebar shows the project structure with "Application Code" and "Sources" folders. The "Sources" folder contains the file `mmult.c`. The right sidebar shows the "Locals" panel, which lists the current variables and their values:

Name	Value
<return value>	0
argc	1
argv	0x7ffc46e5c4f8
mr	32764
nproc	0
sz	0
slice	0
mat_a	0x15517542bb18
mat_b	0x0
mat_c	0x6800420048002200
filename	"[310]261[377F]374[177]"
remainder	65536
st	

The bottom panel shows the "Stacks (All)" view, which lists the current stack frames:

Processes	Threads	Function
8	8	main (mmult.c:92)
8	8	ucs_async_thread_func

Linaro DDT - Linaro Forge 24.0.4

File Edit View Control Tools Window Help

Current Group: All Focus on current: ☒ Group ☐ Process ☐ Thread

All 0 1 2 3 4 5 6 7

Create Group

Project Files

Search (Ctrl+K)

Application Code

Sources

mmult.c

main(int argc, char * argv)

minit(int sz, double * A)

mmult(int sz, int nslices, double * A, double * B)

mwrite(int sz, double * A, double * B)

External Code

```
82
83
84 int main(int argc, char *argv[])
85 {
86     int mr, nproc, sz, slice;
87     double *mat_a, *mat_b, *mat_c;
88     char filename[32];
89     int remainder;
90     MPI_Status st;
91
92     MPI_Init (&argc, &argv);
93     MPI_Comm_rank(MPI_COMM_WORLD, &mr); // my rank
94     MPI_Comm_size(MPI_COMM_WORLD, &nproc); // number of processors
95
96     if (mr == 0)
97     {
98         printf(WORKED_EXAMPLE_NOTICE "\n\n");
99     }
100
101     if (argc > 3 || (argc > 2 && strcmp(argv[1], "-h") != 0) )
102     {
```

Locals

Current Line(s)

Current Stack

Locals

Name	Value
<return value>	0
argc	1
argv	0x7ffc46e5c4f8
mr	32764
nproc	0
sz	0
slice	0
mat_a	0x17542bb18
mat_b	0x0420048002200
mat_c	0x0000000000000000
filename	0x0000000000000000
remainder	65536
st	0x0000000000000000

Input/Output Breakpoints Watchpoints Stacks (All) Tracepoints Tracepoint Output Logbook

Stacks (All)

Processes	Threads	Function
8	8	main (mmult.c:92)
8	8	ucs_async_thread_func

Evaluate

Name	Value
------	-------

Ready

Debugger Controls

Debugger Controls

Current Group: All | Focus on current: ☒ Group ☐ Process ☐ Thread ☐ Step Threads Together

Create Group

Project Files

Search (Ctrl+K)

Application Code

Sources

mmult.c

main(int argc, char * argv)

main(int sz, double * A)

mmult(int sz, int nslices, double * A, double * B)

mwrite(int sz, double * A, double * B)

External Code

```
82
83
84 int main(int argc, char *argv[])
85 {
86     int mr, nproc, sz, slice;
87     double *mat_a, *mat_b, *mat_c;
88     char filename[32];
89     int remainder;
90     MPI_Status st;
91
92     MPI_Init (&argc, &argv);
93     MPI_Comm_rank(MPI_COMM_WORLD, &mr); // my rank
94     MPI_Comm_size(MPI_COMM_WORLD, &nproc); // number of processors
95
96     if (mr == 0)
97     {
98         printf(WORKED_EXAMPLE_NOTICE "\n\n");
99     }
100
101     if(argc > 3 || (argc > 2 && strcmp(argv[1], "-h") != 0) )
102     {
```

Locals

Name	Value
<return value>	0
argc	1
argv	0x7ffc46e5c4f8
mr	32764
nproc	0
sz	0
slice	0
mat_a	0x15517542bb18
mat_b	0x0
mat_c	0x6800420048002200
filename	"[310]261[377F]374[177]"
remainder	65536
st	

Input/Output | Breakpoints | Watchpoints | Stacks (All) | Tracepoints | Tracepoint Output | Logbook

Stacks (All)

Processes	Threads	Function
8	8	main (mmult.c:92)
8	8	ucs_async_thread_func

Evaluate

Name	Value
------	-------

Ready

Debugger
Controls

Code
Editor

Debugger Controls

Code Editor

Debugger Interface: Linaro DDT - Linaro Forge 24.0.4

Current Group: All | Focus on current: ☒ Group ☐ Process ☐ Thread ☐ Step Threads Together

Project Files

Search (Ctrl+K)

Application Code

Sources

mmult.c

main(int argc, char * argv)

minit(int sz, double * A)

mmult(int sz, int nslices, double * A)

mwrite(int sz, double * A)

External Code

mmult.c

```
82
83
84 int main(int argc, char *argv[])
85 {
86     int mr, nproc, sz, slice;
87     double *mat_a, *mat_b, *mat_c;
88     char filename[32];
89     int remainder;
90     MPI_Status st;
91
92     MPI_Init (&argc, &argv);
93     MPI_Comm_rank(MPI_COMM_WORLD, &mr); // my rank
94     MPI_Comm_size(MPI_COMM_WORLD, &nproc); // number of processors
95
96     if (mr == 0)
97     {
98         printf(WORKED_EXAMPLE_NOTICE "\n\n");
99     }
100
101     if(argc > 3 || (argc > 2 && strcmp(argv[1], "-h") != 0) )
102     {
```

Locals

Name	Value
<return value>	0
argc	1
argv	0x7ffc46e5c4f8
mr	32764
nproc	0
sz	0
slice	0
mat_a	0x15517542bb18
mat_b	0x0
mat_c	0x6800420048002200
filename	"[310]261[377F]374[177]"
remainder	65536
st	

Current Stack

Stacks (All)

Processes	Threads	Function
8	8	main (mmult.c:92)
8	8	ucs_async_thread_func

Evaluate

Name	Value
------	-------

Ready

Debugger
Controls

Code
Editor

“Point”
Controls

The screenshot displays the Linaro DDT - Linaro Forge 24.0.4 interface. The top menu bar includes File, Edit, View, Control, Tools, Window, and Help. Below the menu is a toolbar with various icons for running, stepping, and debugging. The main window is divided into several panels:

- Project Files:** A tree view on the left showing the project structure, including Application Code, Sources, and External Code. The file `mmult.c` is selected.
- Code Editor:** The central area showing the source code of `mmult.c`. The code includes headers, variable declarations, and MPI-related functions. The current line of execution is highlighted at line 92.
- Debugger Controls:** A panel on the right side of the code editor, containing tabs for Locals, Current Line(s), and Current Stack. The Locals tab is active, showing a list of variables and their values.
- Stacks (All):** A panel at the bottom left showing the call stack. It lists processes, threads, and functions. The current function is `main (mmult.c:92)`.
- Evaluate:** A panel at the bottom right for evaluating expressions, with a table for Name and Value.

The interface is divided into three main sections by red lines: the top section for debugger controls, the middle section for the code editor, and the bottom section for the "Point" controls (stacks and evaluate).

Debugger
Controls

Code
Editor

“Point”
Controls

Debugger Controls

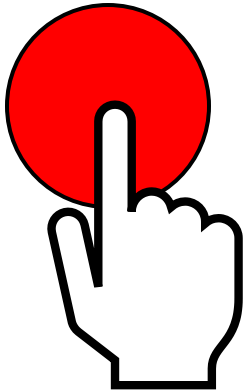
Code Editor

“Point” Controls

Variables, Data

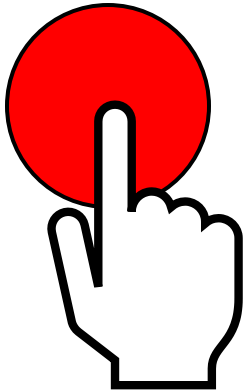
The 3 “points” of Debugging

Breakpoint

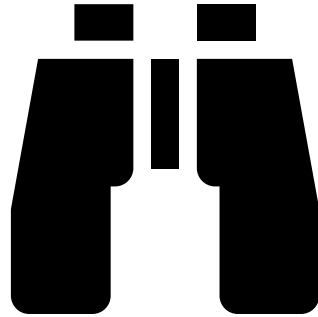


The 3 “points” of Debugging

Breakpoint

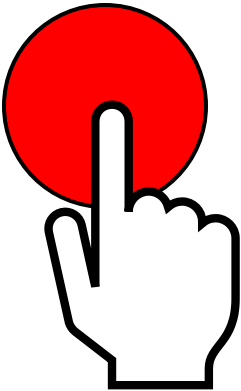


Watchpoint

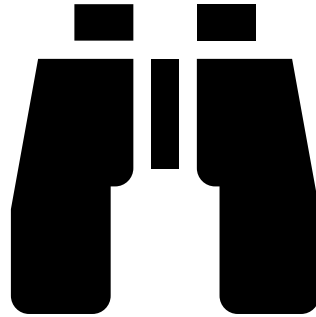


The 3 “points” of Debugging

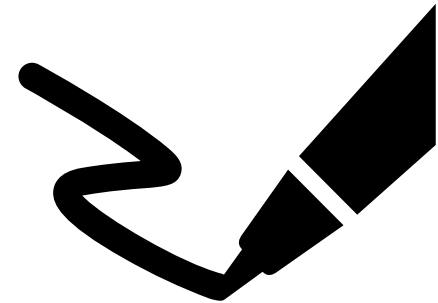
Breakpoint



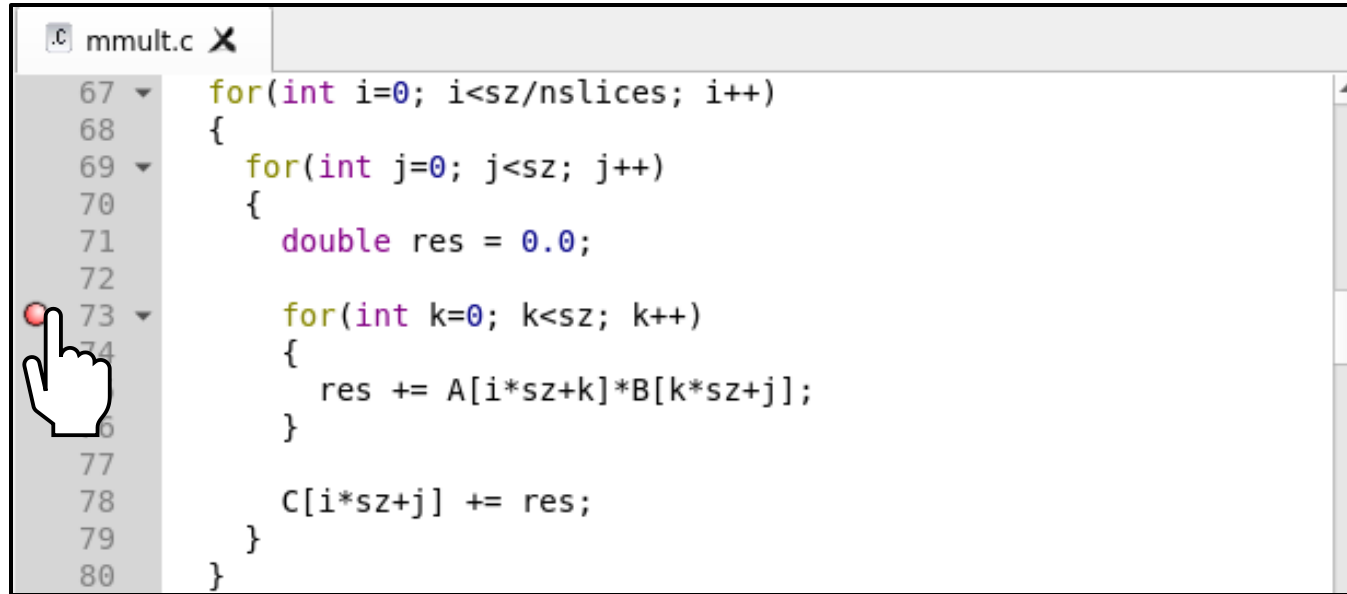
Watchpoint



Tracepoint



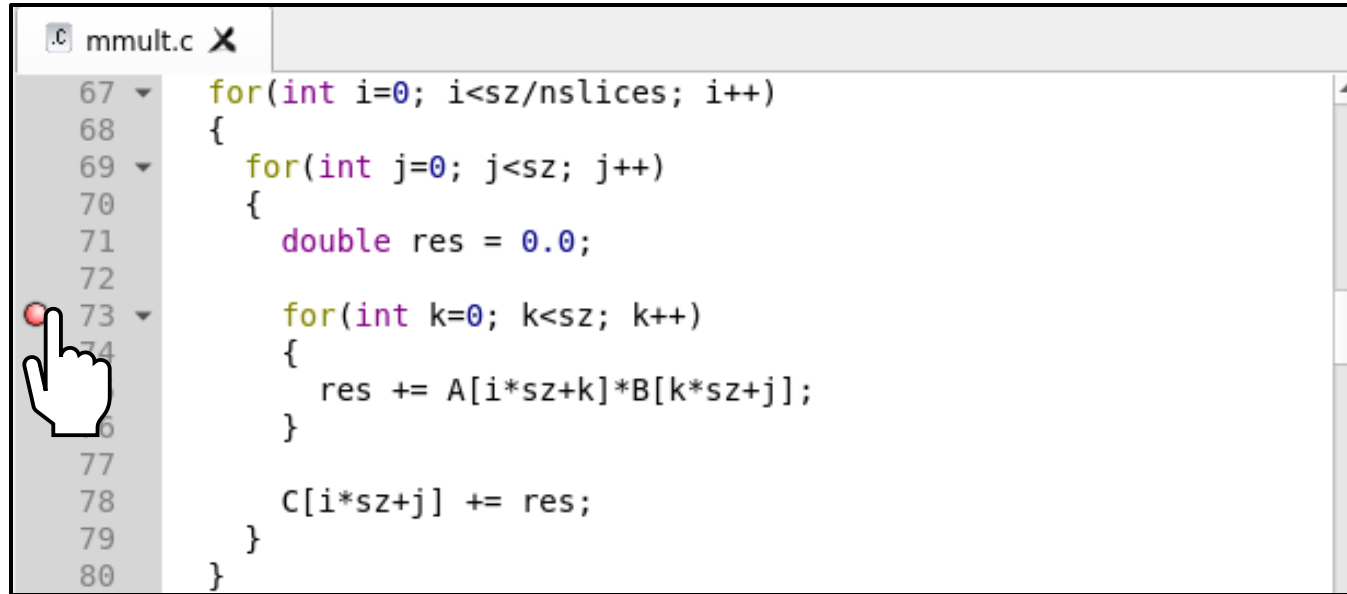
Breakpoints



The image shows a code editor window titled "mmult.c". The code is a C program with three nested for loops. A red circle, representing a breakpoint, is set on line 73, which is the start of the innermost loop. A mouse cursor is pointing at this red circle. The code is as follows:

```
67 for(int i=0; i<sz/nslices; i++)
68 {
69     for(int j=0; j<sz; j++)
70     {
71         double res = 0.0;
72
73         for(int k=0; k<sz; k++)
74         {
75             res += A[i*sz+k]*B[k*sz+j];
76         }
77
78         C[i*sz+j] += res;
79     }
80 }
```

Breakpoints

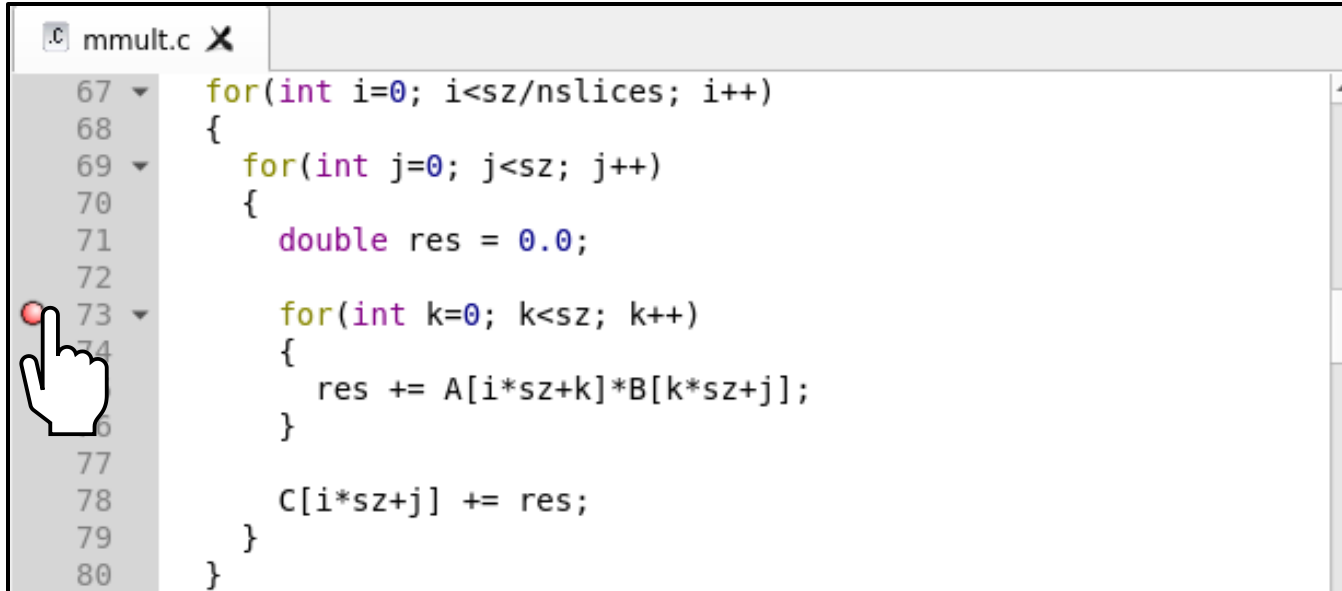


The screenshot shows a code editor window titled "mmult.c". The code is a C program for matrix multiplication. A red circle breakpoint is set on line 73, which is the start of the innermost loop. A mouse cursor is pointing at the breakpoint.

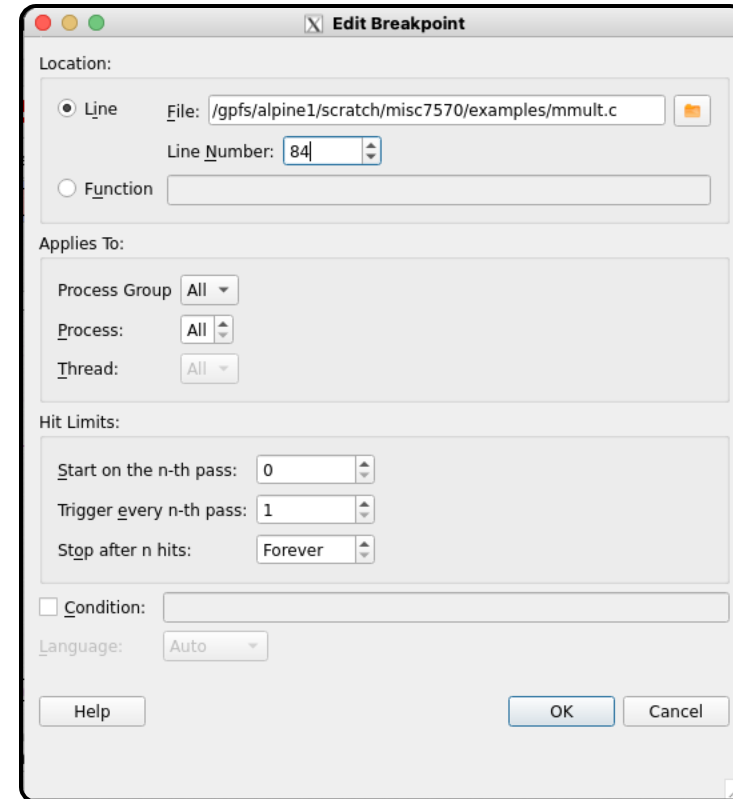
```
67 for(int i=0; i<sz/nslices; i++)
68 {
69     for(int j=0; j<sz; j++)
70     {
71         double res = 0.0;
72
73         for(int k=0; k<sz; k++)
74         {
75             res += A[i*sz+k]*B[k*sz+j];
76         }
77
78         C[i*sz+j] += res;
79     }
80 }
```

Input/Output Breakpoints Watchpoints Stacks (All) Tracepoints Tracepoint Output Logbook											
Breakpoints											
	Processes	Threads	File	Line	Actual Line	Function	Condition	Start After	Trigger Every	Stop After	Full
<input checked="" type="checkbox"/>	All	all	mmult.c	73	73	mmult		0	1	Forever	/gp
<input checked="" type="checkbox"/>	All	all	mmult.c	84	92	main		0	1	Forever	/gp

Breakpoints



```
.C mmult.c X
67  for(int i=0; i<sz/nslices; i++)
68  {
69      for(int j=0; j<sz; j++)
70      {
71          double res = 0.0;
72
73          for(int k=0; k<sz; k++)
74          {
75              res += A[i*sz+k]*B[k*sz+j];
76          }
77
78          C[i*sz+j] += res;
79      }
80  }
```



Location:

☒ Line File: /gpfs/alpine1/scratch/misc7570/examples/mmult.c
Line Number: 84

☐ Function

Applies To:

Process Group: All
Process: All
Thread: All

Hit Limits:

Start on the n-th pass: 0
Trigger every n-th pass: 1
Stop after n hits: Forever

☐ Condition:

Language: Auto

Help OK Cancel

Input/Output	Breakpoints	Watchpoints	Stacks (All)	Tracepoints	Tracepoint Output	Logbook					
Breakpoints 🔍 ✖											
	Processes	Threads	File	Line	Actual Line	Function	Condition	Start After	Trigger Every	Stop After	Full
<input checked="" type="checkbox"/>	All	all	mmult.c	73	73	mmult		0	1	Forever	/gp
<input checked="" type="checkbox"/>	All	all	mmult.c	84	92	main		0	1	Forever	/gp

Breakpoints

Applies To:

Process Group All ▼

Process: All ▲▼

Thread: All ▼

Edit Breakpoint

Location:

☒ Line File: /gpfs/alpine1/scratch/misc7570/examples/mmult.c 📁

Line Number: 84 ▲▼

☐ Function

Applies To:

Process Group All ▼

Process: All ▲▼

Thread: All ▼

Hit Limits:

Start on the n-th pass: 0 ▲▼

Trigger every n-th pass: 1 ▲▼

Stop after n hits: Forever ▲▼

☐ Condition:

Language: Auto ▼

Help OK Cancel

Breakpoints

Hit Limits:

Start on the n-th pass: 0

Trigger every n-th pass: 1

Stop after n hits: Forever

Edit Breakpoint

Location:

☒ Line File: /gpfs/alpine1/scratch/misc7570/examples/mmult.c
Line Number: 84

☐ Function

Applies To:

Process Group: All

Process: All

Thread: All

Hit Limits:

Start on the n-th pass: 0

Trigger every n-th pass: 1

Stop after n hits: Forever

☐ Condition:

Language: Auto

Help OK Cancel

Breakpoints

☐ Condition:

Edit Breakpoint

Location:

☒ Line File:

Line Number:

☐ Function

Applies To:

Process Group

Process:

Thread:

Hit Limits:

Start on the n-th pass:

Trigger every n-th pass:

Stop after n hits:

☐ Condition:

Language:

Variable Sparkline

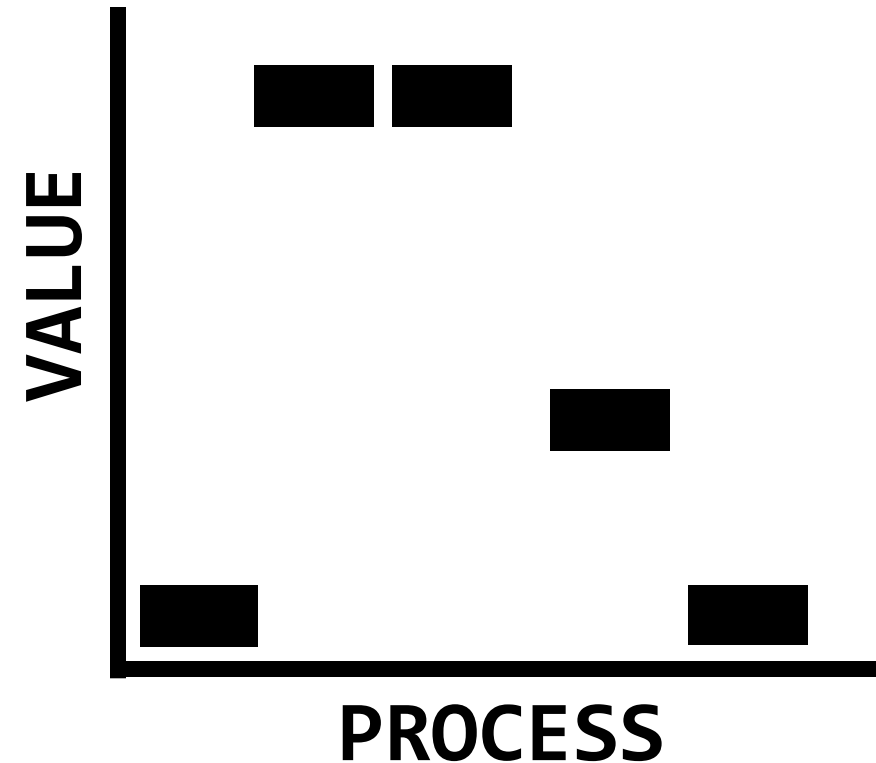
Locals	Current Line(s)	Current Stack
Locals		
Name	Value	
<return value>	0	
argc	1	
▸ argv	0x7ffe6f640048	
mr	32766	
nproc	0	
sz	0	
slice	0	
▸ mat_a	0x14fe7d7f9b18	
▸ mat_b	0x0	
▸ mat_c	0x6800420048002200	
filename	" 3101~o 376 177"	
remainder	65536	
▸ st		

Variable Sparkline

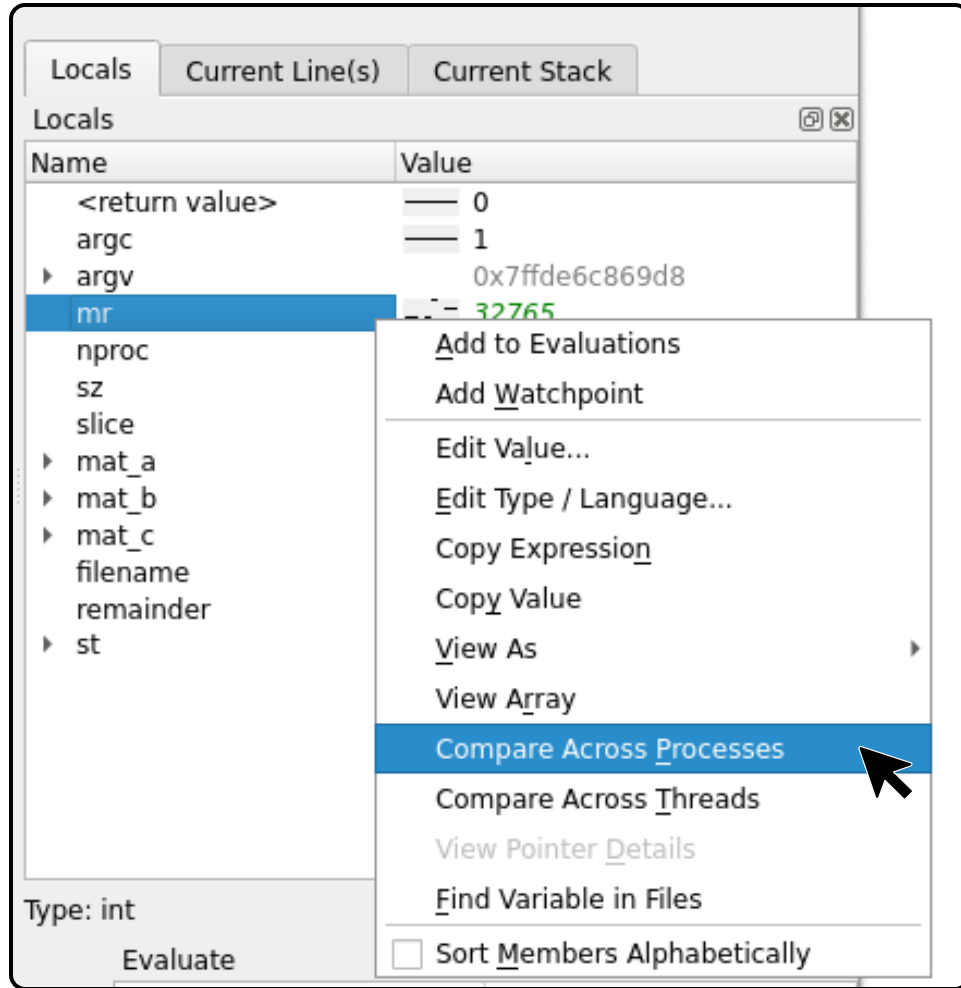
Locals	
Current Line(s)	
Current Stack	
Locals	
Name	Value
<return value>	0
argc	1
▸ argv	0x7ffe6f640048
mr	32766
nproc	0
sz	0
slice	0
▸ mat_a	0x14fe7d7f9b18
▸ mat_b	0x0
▸ mat_c	0x6800420048002200
filename	" 3101~o 376 177"
remainder	65536
▸ st	

Variable Sparkline

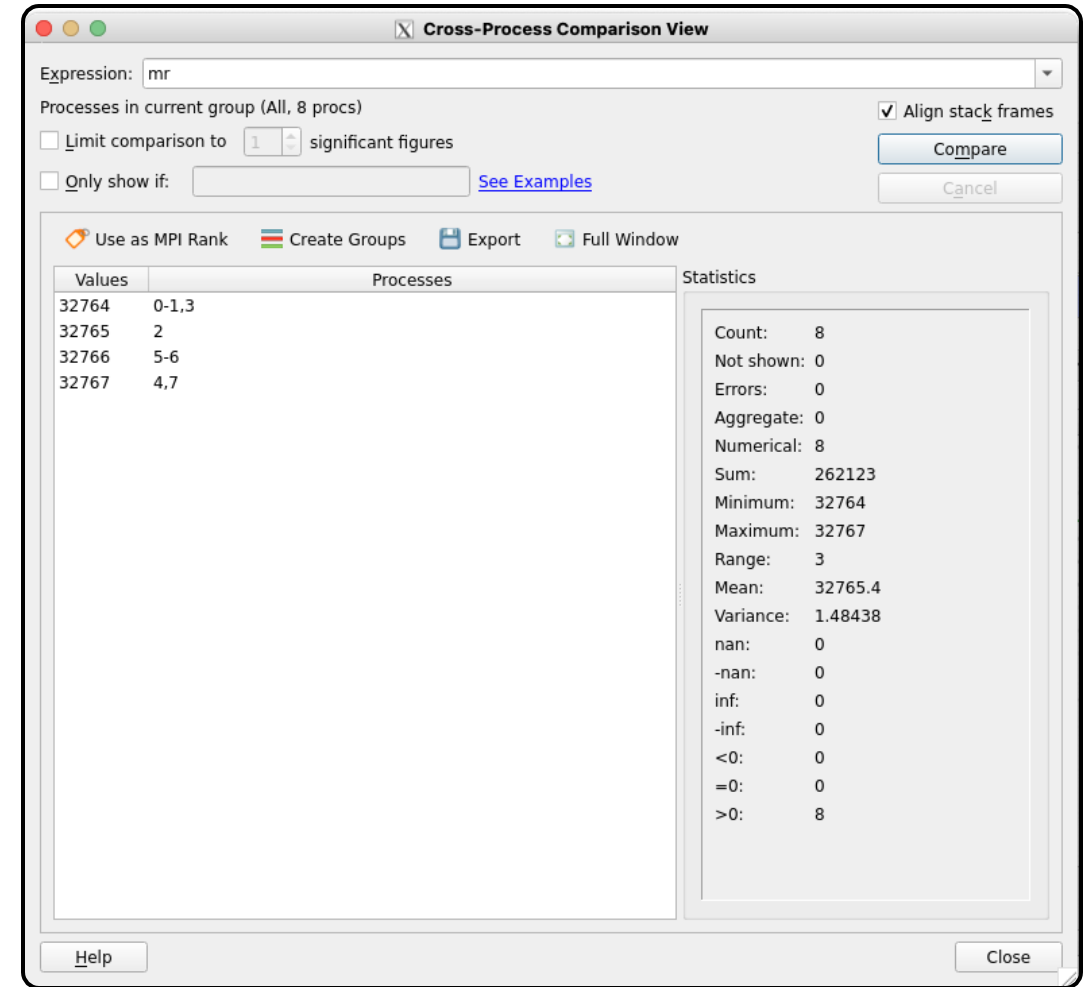
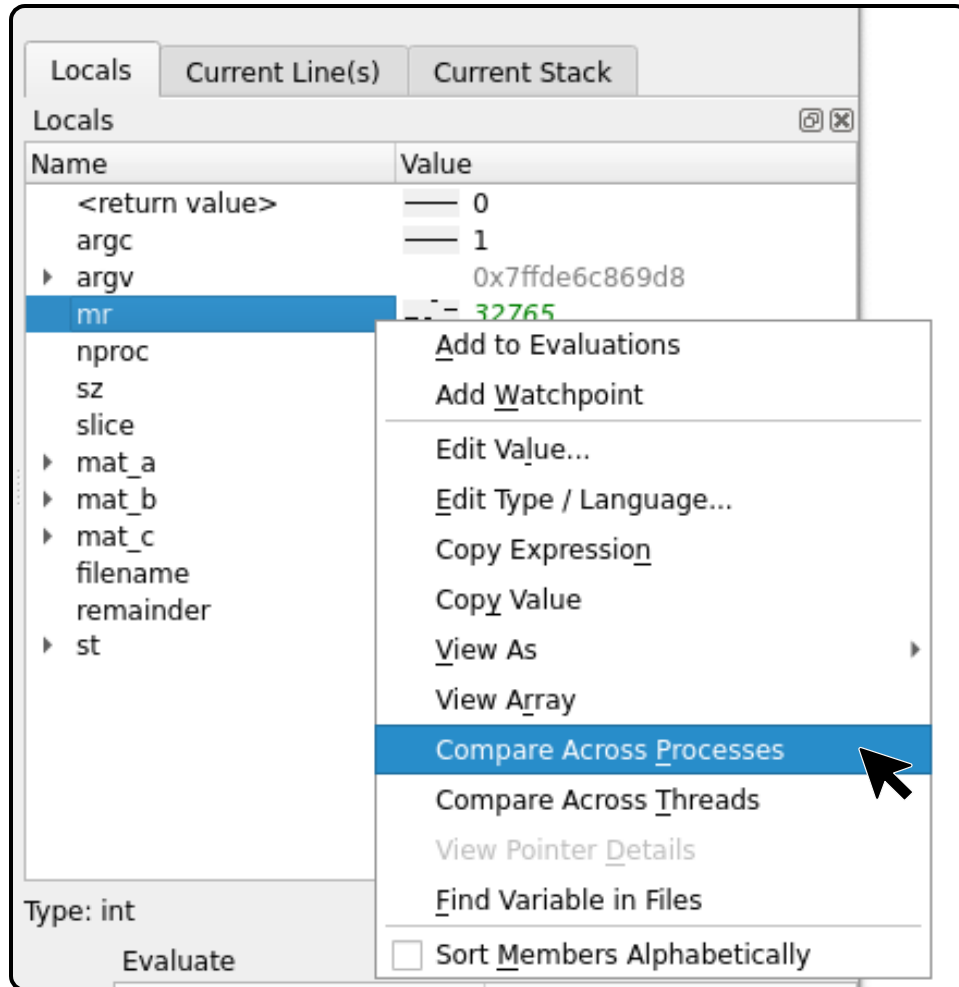
Locals	
Current Line(s)	
Current Stack	
Locals	
Name	Value
<return value>	0
argc	1
▸ argv	0x7ffe6f640048
mr	32766
nproc	0
sz	0
slice	0
▸ mat_a	0x14fe7d7f9b18
▸ mat_b	0x0
▸ mat_c	0x6800420048002200
filename	" 3101~o 376 177"
remainder	65536
▸ st	



Variable Comparison

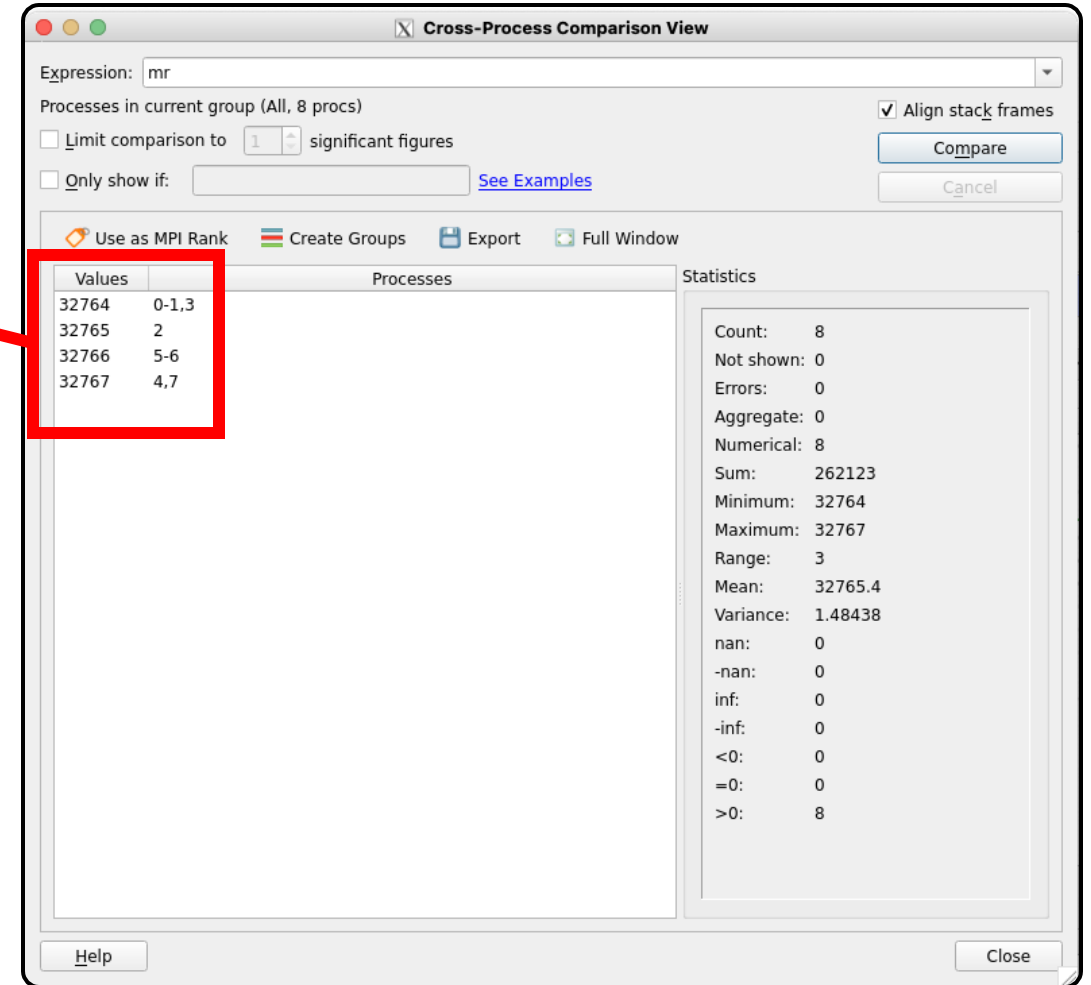


Variable Comparison



Variable Comparison

Values	
32764	0-1,3
32765	2
32766	5-6
32767	4,7



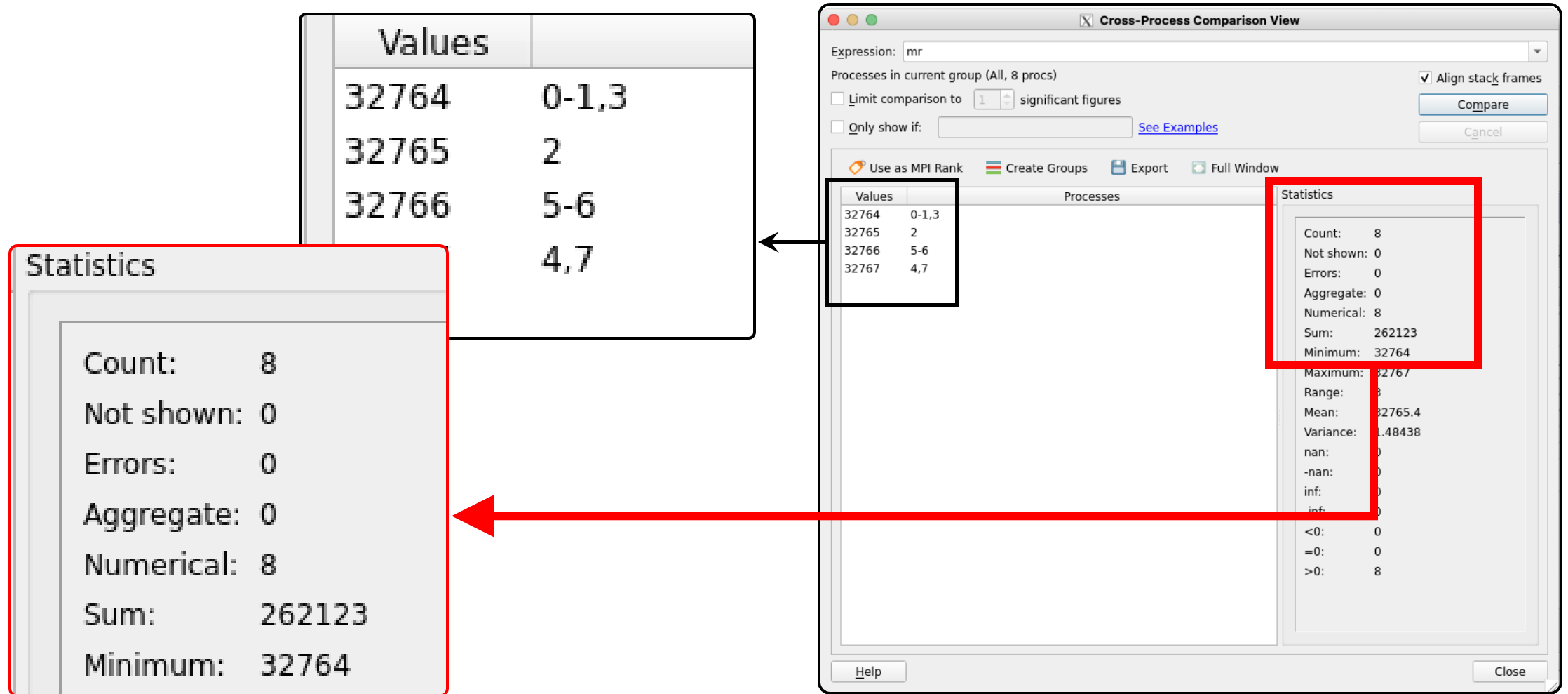
The screenshot shows a 'Cross-Process Comparison View' window. At the top, the 'Expression' is set to 'mr'. Below this, it states 'Processes in current group (All, 8 procs)'. There are checkboxes for 'Limit comparison to 1 significant figures' and 'Only show if:'. A 'Compare' button is visible. Below the controls, there are icons for 'Use as MPI Rank', 'Create Groups', 'Export', and 'Full Window'. The main area contains a table with 'Values' and 'Processes' columns. A red box highlights the first four rows of this table, which correspond to the data in the table on the left. To the right of the table is a 'Statistics' panel with various metrics.

Values	Processes
32764	0-1,3
32765	2
32766	5-6
32767	4,7

Statistics

- Count: 8
- Not shown: 0
- Errors: 0
- Aggregate: 0
- Numerical: 8
- Sum: 262123
- Minimum: 32764
- Maximum: 32767
- Range: 3
- Mean: 32765.4
- Variance: 1.48438
- nan: 0
- nan: 0
- inf: 0
- inf: 0
- <0: 0
- =0: 0
- >0: 8

Variable Comparison



View MD Arrays

Multi-Dimensional Array Viewer

Array Expression:

Distributed Array Dimensions: [How do I view distributed arrays?](#)

☐ Staggered Array [What does this do?](#) ☒ Align Stack Frames

Range of \$i: From: To: Display:

Range of \$j: From: To: Display:

☐ Auto-update

☐ Only show if: [See Examples](#)

Data Table

	j											
i	0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	4	5	6	7	8	9	10	11	12
1	2	4	6	8	10	12	14	16	18	20	22	24
2	3	6	9	12	15	18	21	24	27	30	33	36
3	4	8	12	16	20	24	28	32	36	40	44	48
4	5	10	15	20	25	30	35	40	45	50	55	60
5	6	12	18	24	30	36	42	48	54	60	66	72
6	7	14	21	28	35	42	49	56	63	70	77	84
7	8	16	24	32	40	48	56	64	72	80	88	96
8	9	18	27	36	45	54	63	72	81	90	99	108

View MD Arrays

Multi-Dimensional Array Viewer

Array Expression: `tables[$i][$j]` Evaluate Cancel

Distributed Array Dimensions: None [How do I view distributed arrays?](#)

☐ Staggered Array [What does this do?](#) ☒ Align Stack Frames ☐ Auto-update

Range of \$i: From: To: Display: Rows

Range of \$j: From: To: Display: Columns

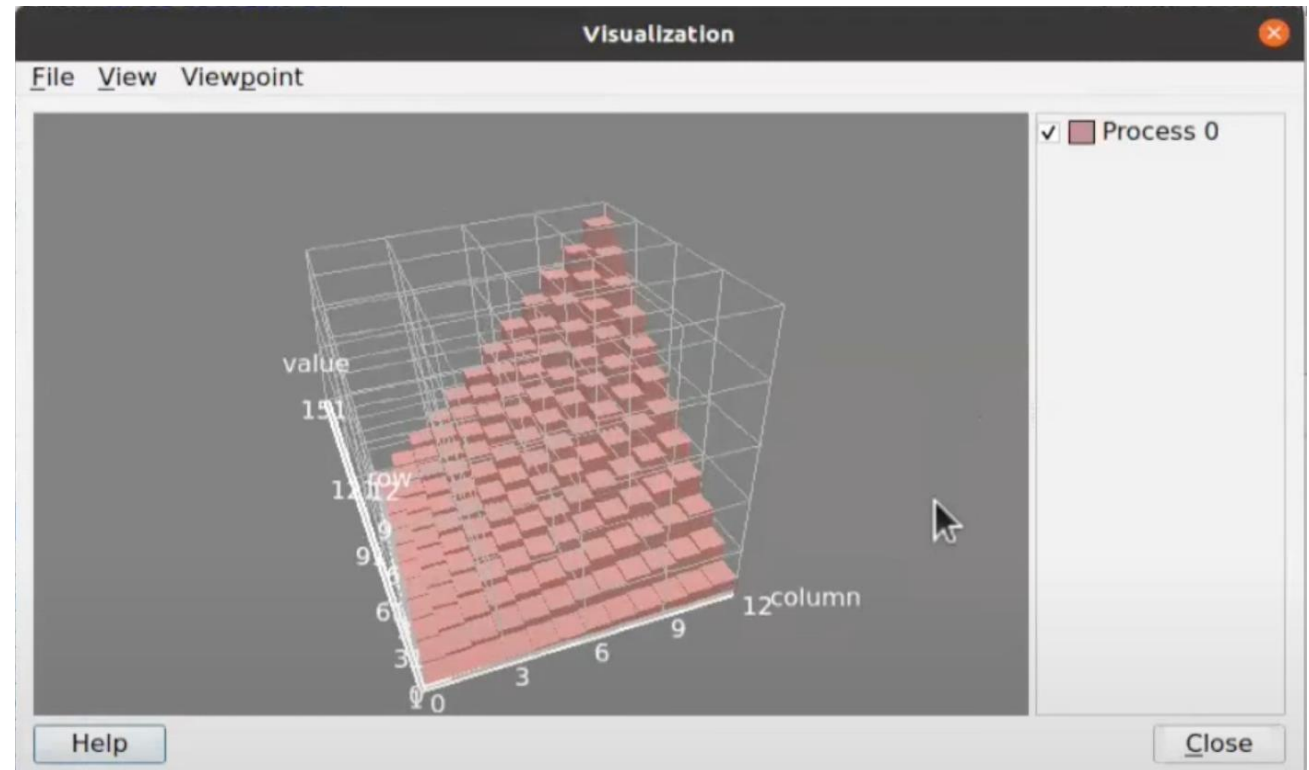
☐ Only show if: [See Examples](#)

Data Table Statistics

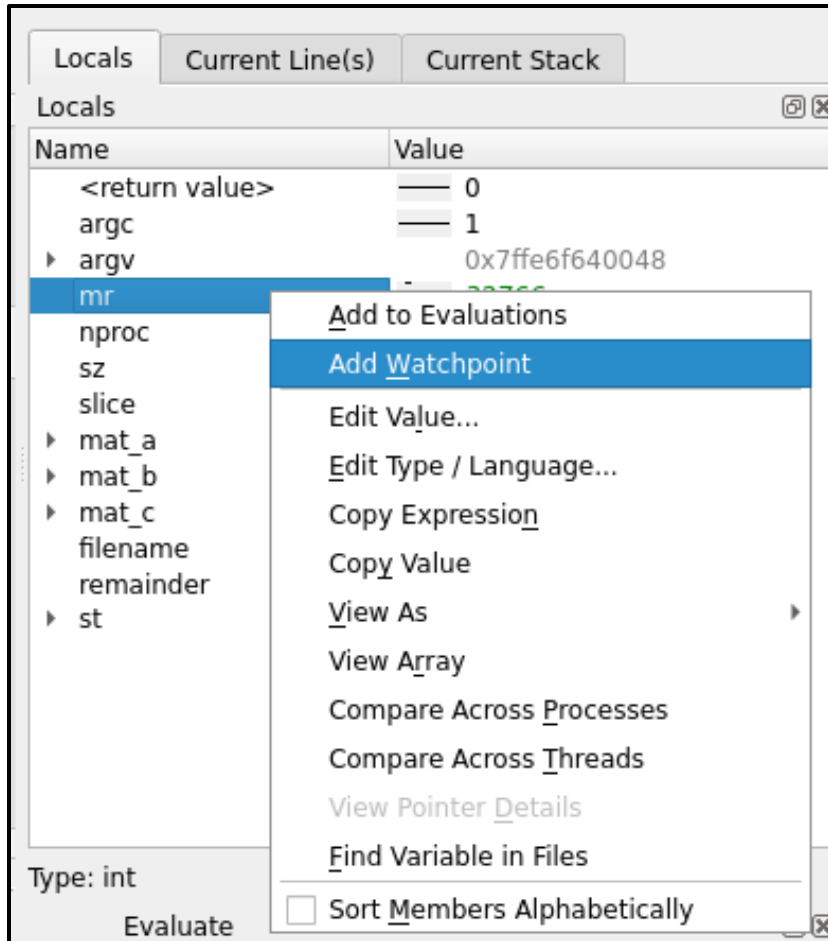
[Goto](#) [Visualize](#) [Export](#) [Full Window](#)

	j											
i	0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	4	5	6	7	8	9	10	11	12
1	2	4	6	8	10	12	14	16	18	20	22	24
2	3	6	9	12	15	18	21	24	27	30	33	36
3	4	8	12	16	20	24	28	32	36	40	44	48
4	5	10	15	20	25	30	35	40	45	50	55	60
5	6	12	18	24	30	36	42	48	54	60	66	72
6	7	14	21	28	35	42	49	56	63	70	77	84
7	8	16	24	32	40	48	56	64	72	80	88	96
8	9	18	27	36	45	54	63	72	81	90	99	108

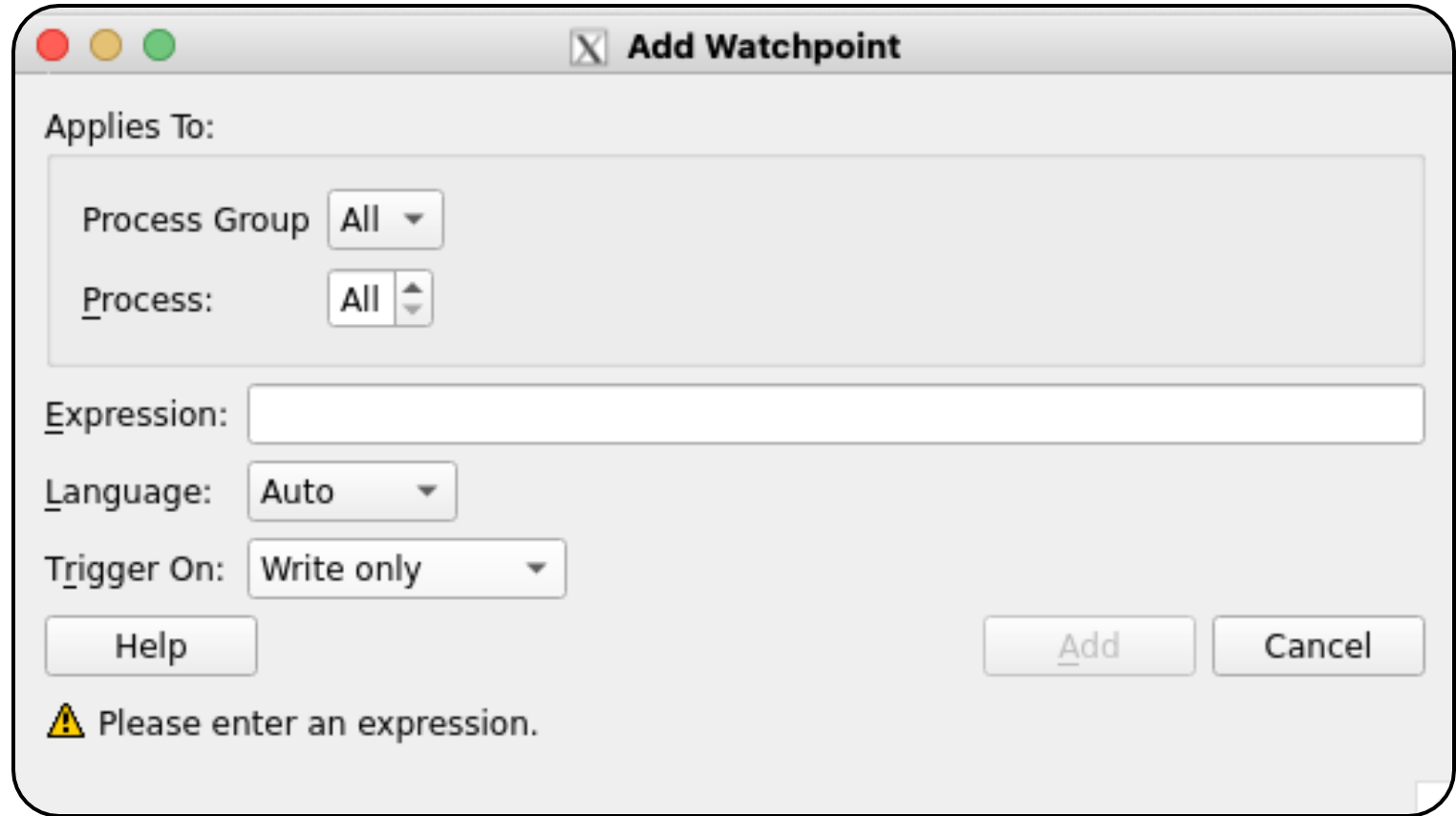
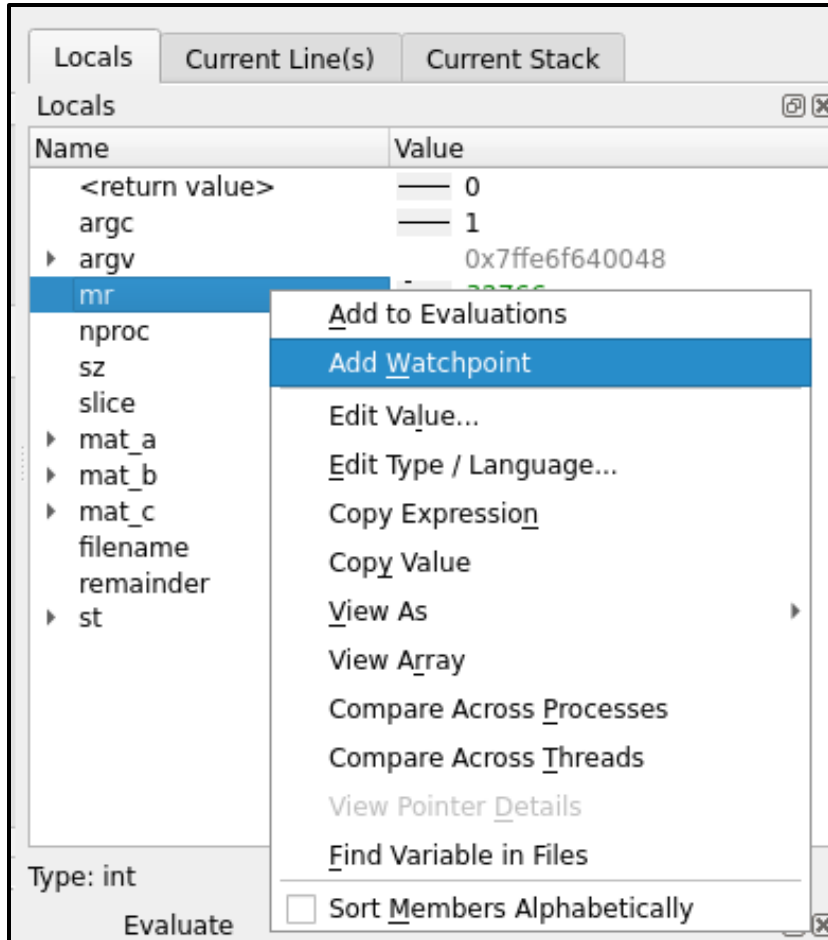
Help Close



Watchpoints



Watchpoints



Watchpoints

Expression:

Language:

Trigger On:

Add Watchpoint

Applies To:


Process Group:

Process:

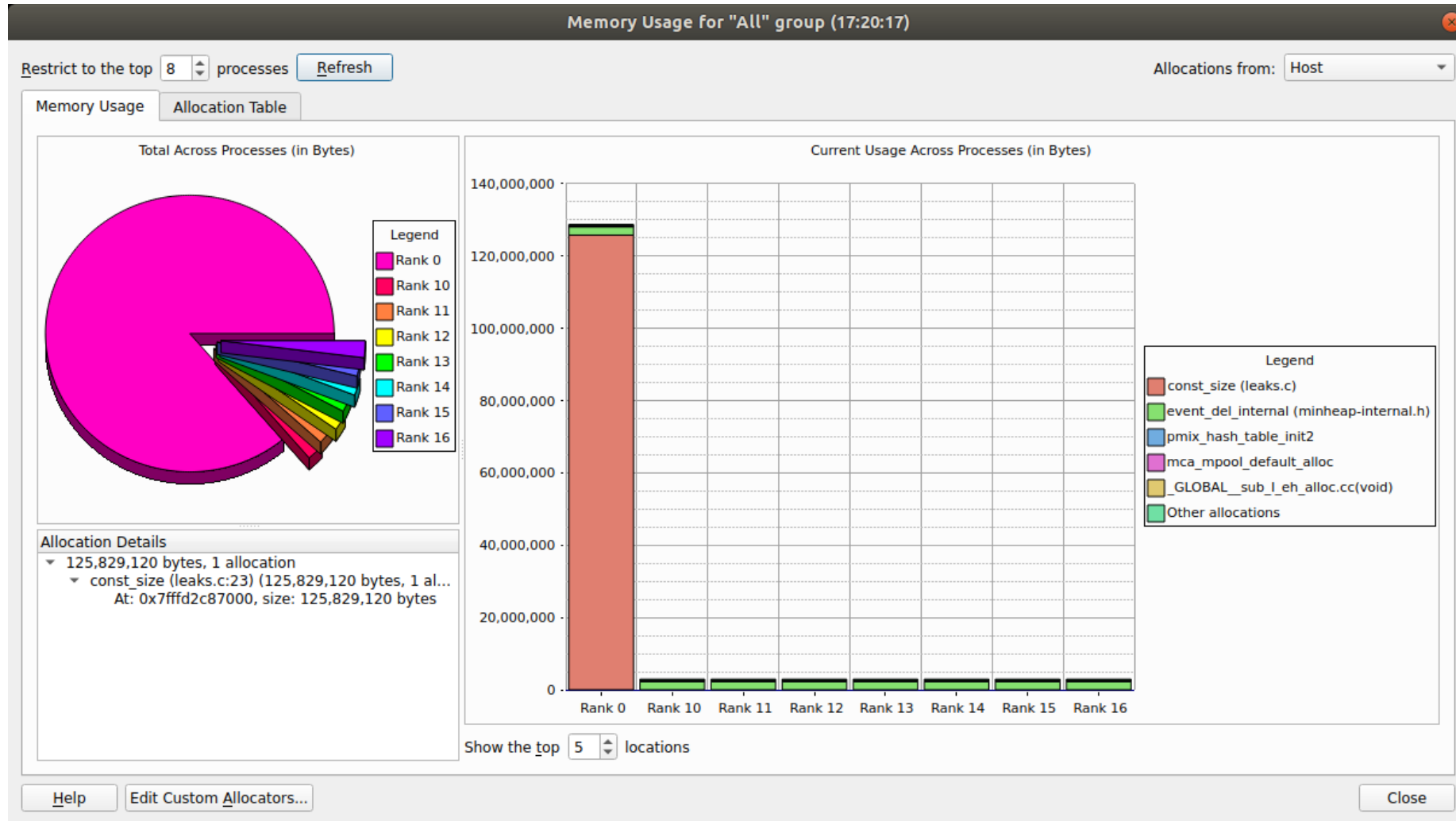
Expression:

Language:




Trigger On:

 Please enter an expression.

Memory Debugger








Tracepoints

Tracepoint	Processes	Values logged
subdomain (subdomain.f90:59)	16, ranks 0-15	ny:  16 nx:  16 nz:  64
blts (blts.f90:58)	1, rank 0	m: 1 iend: 16 ldmz: 64 k: 2 ldmx: 16 i: 2 ldz: ist: 2 j: 2 ldmy: 16
blts (blts.f90:58)	1, rank 0	m: 2 iend: 16 ldmz: 64 k: 2 ldmx: 16 i: 2 ldz: ist: 2 j: 2 ldmy: 16
blts (blts.f90:58)	1, rank 0	m: 3 iend: 16 ldmz: 64 k: 2 ldmx: 16 i: 2 ldz: ist: 2 j: 2 ldmy: 16
blts (blts.f90:58)	1, rank 0	m: 4 iend: 16 ldmz: 64 k: 2 ldmx: 16 i: 2 ldz: ist: 2 j: 2 ldmy: 16

Logbook

Input/OutputBreakpointsWatchpointsStacks (All)TracepointsTracepoint OutputLogbook




Logbook

Time	Ranks	Message
0:00	0-7	 Launching /gpfs/alpine1/scratch/misc7570/examples/mmult_c at Tue Oct 8 08:23:28 2024 Executable modified on Fri Oct 4 14:51:17 2024
0:32	0-7	 Startup complete.
▶ 0:32	n/a	Select process group All
0:32	0-7	 Add breakpoint for mmult.c:73
1:26	0-7	 Add breakpoint for mmult.c:84
28:58	0-7	Add watchpoint for mr
43:02	0-7	Remove watchpoint for mr
46:35	0-7	Add watchpoint for mr
57:39	n/a	 Comment I wrote Yay!

Save

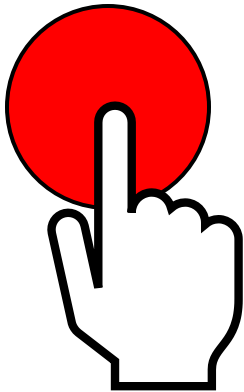
Compare

Annotate

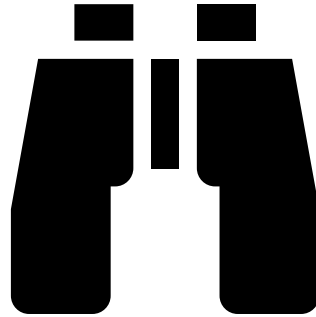


When to use each “point”

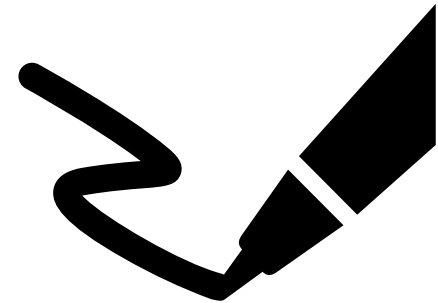
Breakpoint



Watchpoint



Tracepoint



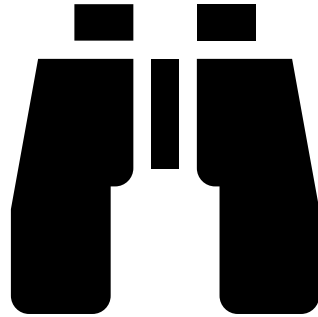
When to use each “point”

Breakpoint

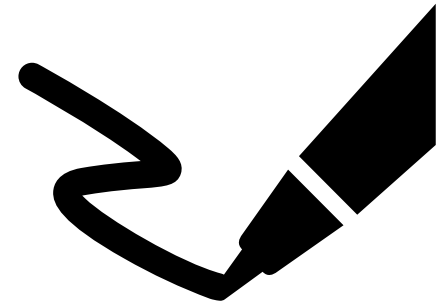
Observe
Code
Location



Watchpoint



Tracepoint



When to use each “point”

Breakpoint

Observe
Code
Location

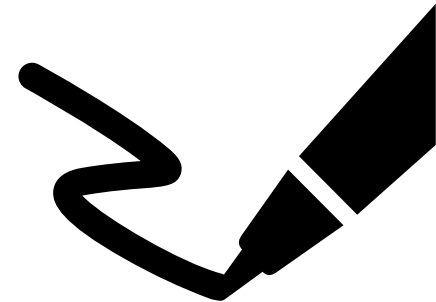


Watchpoint

Observe
Data
Location



Tracepoint



When to use each “point”

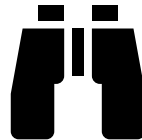
Breakpoint

Observe
Code
Location



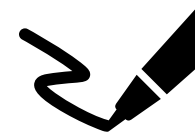
Watchpoint

Observe
Data
Location



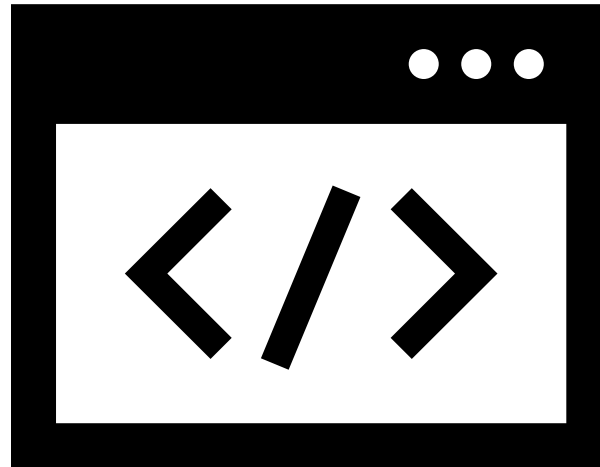
Tracepoint

Record
Code
Location



Offline Debugging

```
$ ddt --offline mpiexec -n <Num-Proc> <Program> <Arguments>
```



Offline Debugging

Debugging /gpfs/alpine1/scratch/misc7570/examples/mmult_c








Messages

Tracepoints

Output

Messages

[+] Expand All [-] Collapse All

#	Type	Time	Processes	Message
1		0:00.000	n/a	Launching mpiexec -n 8 ./mmult_c at Thu Oct 3 10:51:52 2024
2		0:04.071	n/a	No debug symbols were loaded for the glibc library. It is recommended you install the glibc debug symbols.
3		0:06.648	n/a	No debug symbols were loaded for the glibc library. It is recommended you install the glibc debug symbols.
4		0:09.313	0-7	Startup complete.
5		0:09.314	n/a	Select process group All
6		0:11.991	n/a	Debugging : mpiexec -n 8 ./mmult_c MPI implementation : Auto-Detect (Intel MPI (MPMD)) * number of processes : 8 * number of nodes : 2 Memory debugging enabled : No
7		0:11.992	0-7	Play
8		0:12.056	0-7	Process stopped in mmult (mmult.c:75) with signal SIGSEGV (Segmentation fault). Reason/Origin: address not mapped to object (attempt to access invalid address)
9				Additional Information ▶ Stacks ▶ Current Stack

Offline Debugging

```
$ ddt --offline mpiexec -n <Num-Proc> <Program> <Arguments>
```

Additional Command Line Options:

--session=SESSIONFILE

--mem-debug[=(fast/balanced/thorough/off)]

--snapshot-interval=MINUTES

--trace-at=LOCATION[,N:M:P],VAR1,VAR2,...] [if CONDITION]

Offline Debugging - Tracepoint

Messages

Tracepoints

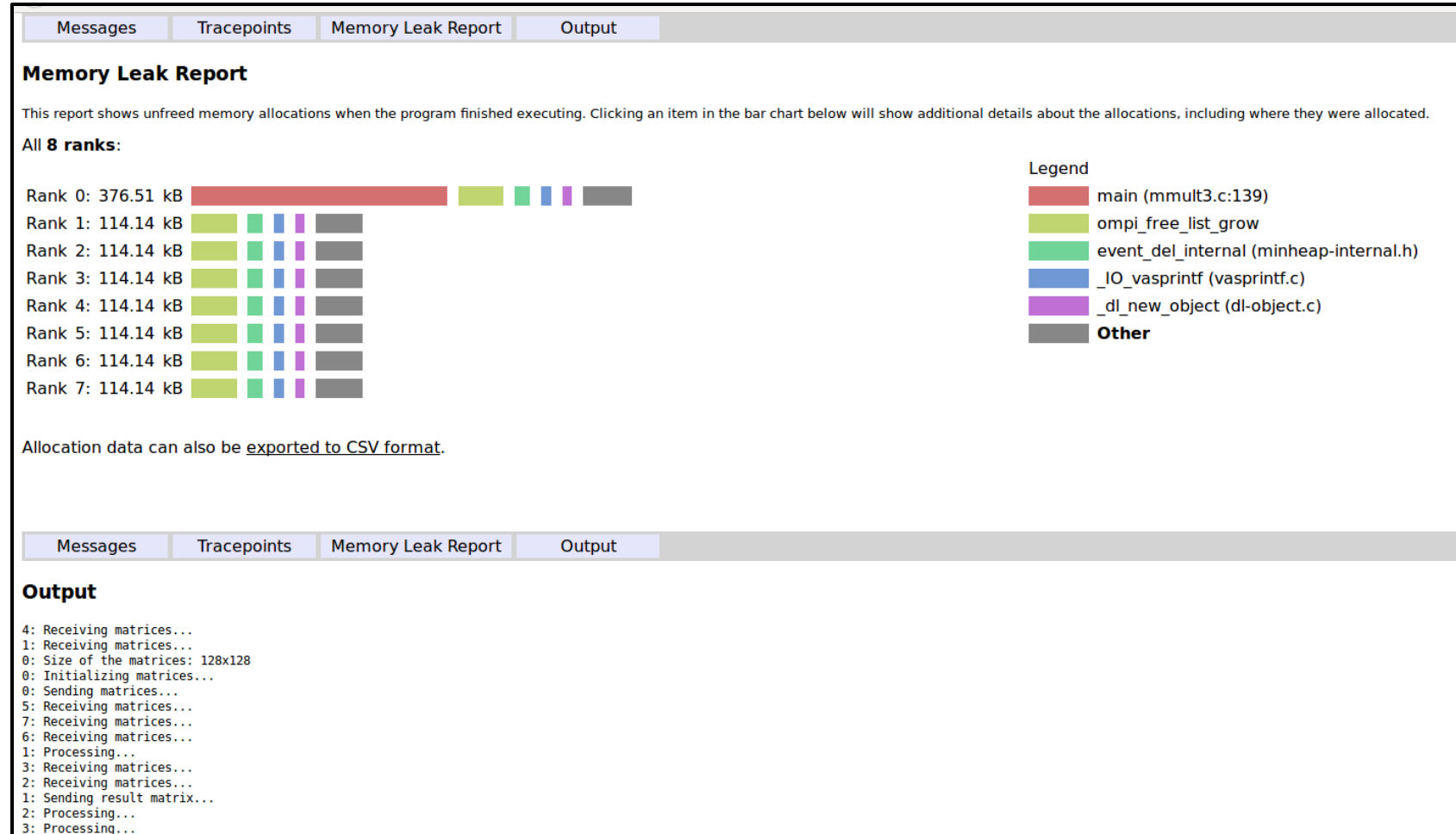
Memory Leak Report

Output

Tracepoints

#	Time	Tracepoint	Processes	Values
1	0:05.274	main (hello.c:91)	0-3	x: 0
2	0:05.274	main (hello.c:91)	0-3	x: 1000
3	0:05.575	main (hello.c:91)	0-3	x: 2000
4	0:05.766	main (hello.c:91)	0-3	x: 3000
5	0:05.766	main (hello.c:91)	0-3	x: 4000
6	0:06.066	main (hello.c:91)	0-3	x: 5000
7	0:06.276	main (hello.c:91)	0-3	x: 6000
8	0:06.276	main (hello.c:91)	0-3	x: 7000
9	0:06.546	main (hello.c:91)	0-3	x: 8000
10	0:06.738	main (hello.c:91)	0-3	x: 9000

Offline Debugging – Memory Leak



Survey and feedback



<http://tinyurl.com/curc-survey18>

56