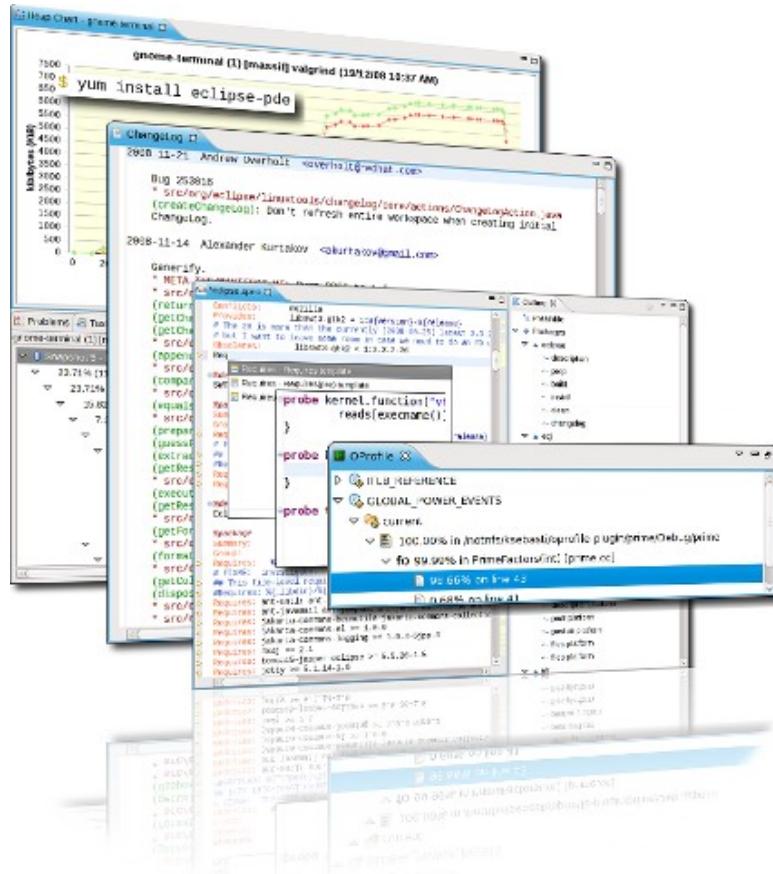


Eclipse Linux Tools Project

Andrew Overholt
Red Hat



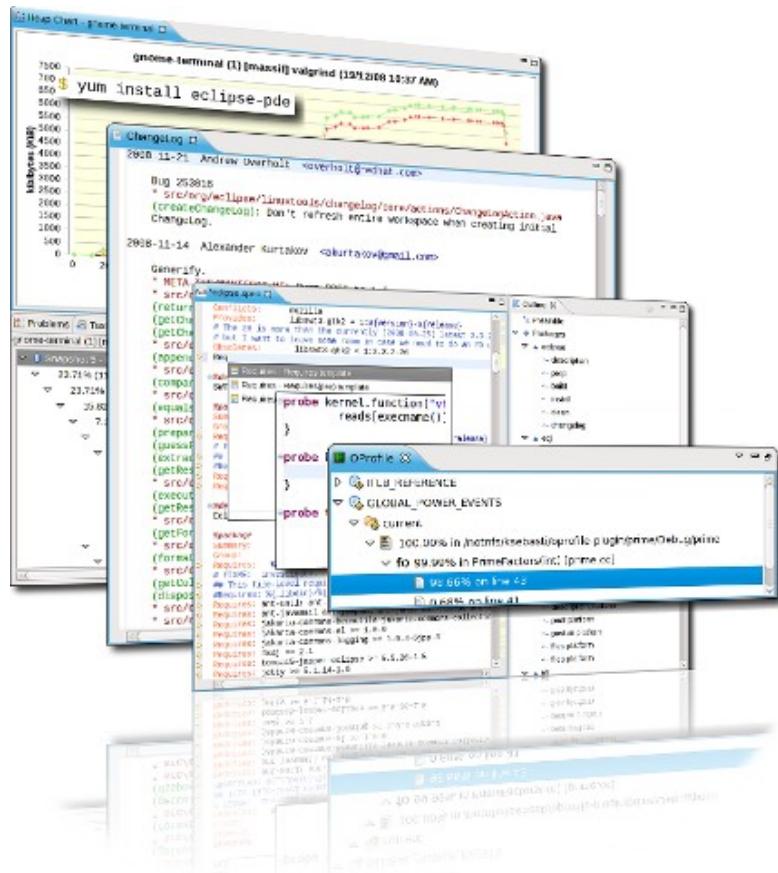
Linux Tools Project



- Project goals
- What we provide
- Project details
- How to get involved



Linux Tools Project

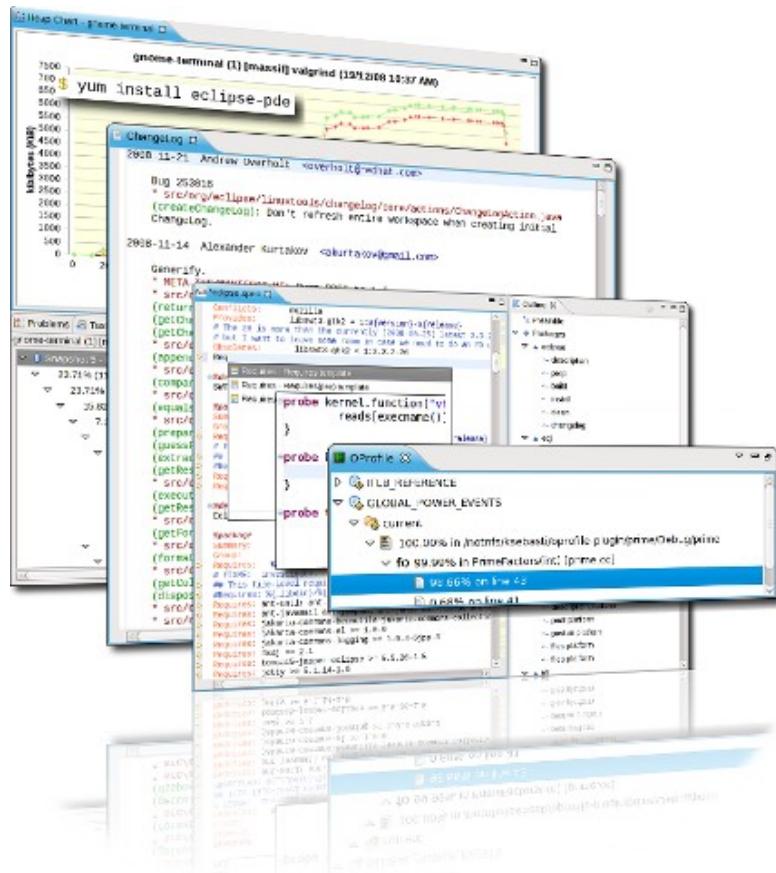


Project goals

- Increase availability and quality of Eclipse technology in Linux distributions
- Provide Eclipse tools for Linux C/C++ developers



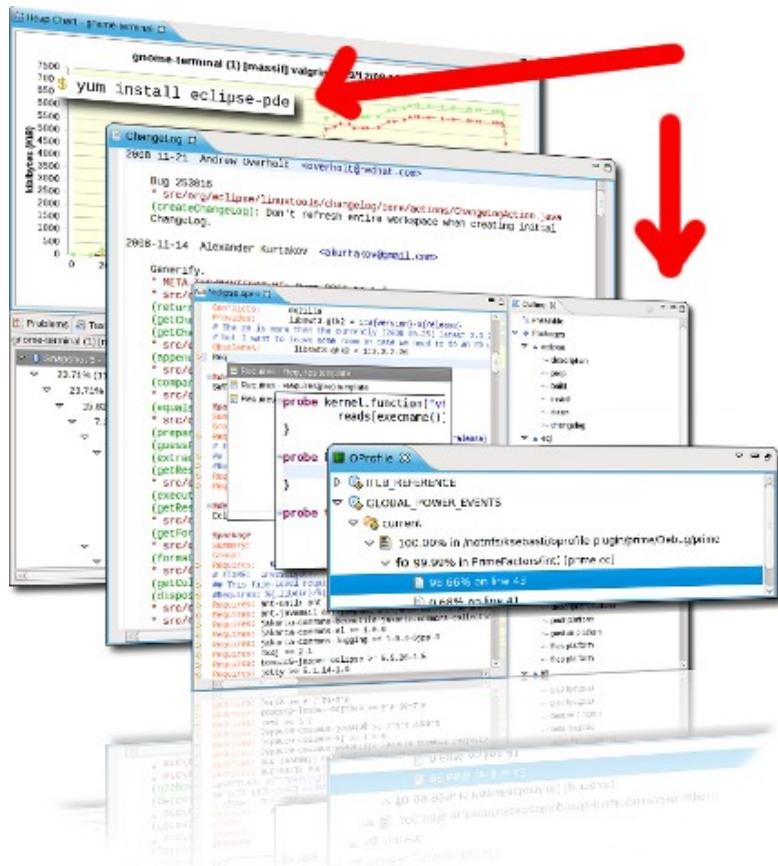
Linux Tools Project



What we provide

- Tools for Linux packagers
- Tools for C/C++ developers built on CDT

Linux Tools Project



What we provide

- Tools for Linux packagers
- Tools for C/C++ developers built on CDT



RPM .spec editor

The screenshot shows the Eclipse RPM Spec Editor interface. On the left is the main editor window titled "eclipse-oprofile.spec" containing the RPM spec file content. On the right is the "Outline" view showing the structure of the spec file.

eclipse-oprofile.spec

```
Name:           eclipse-oprofile
Version:        0.1.0
Release:        3%{?dist}
Summary:        Eclipse plugin for OProfile integration

Group:         Development/Tools
License:        EPL
URL:          http://www.eclipse.org/linuxtools/projectPages/oprofile/
## sh %{name}-fetch-src.sh
Source0:       %{name}-fetched-src-%{src_repo_tag}.tar.bz2
Source1:       %{name}-fetch-src.sh
Patch0:        %{name}-includefixes.patch
BuildRoot:     %{_tmppath}/%{name}-%{version}-%{release}-root-%(%{_id_u} -n

ExcludeArch:   ppc ppc64

BuildRequires: eclipse-pde >= 1:3.4.0
BuildRequires: eclipse-cdt >= 5.0.1
BuildRequires: eclipse-linuxprofilingframework >= 0.1.0
BuildRequires: oprofile >= 0.9.3
BuildRequires: oprofile-devel >= 0.9.3
BuildRequires: binutils-devel >= 2.18.50.0.6
Requires:      eclipse-platform >= 3.4.0
Requires:      eclipse-cdt >= 5.0.1
Requires:      eclipse-linuxprofilingframework >= 0.1.0
Requires:      oprofile >= 0.9.3
Requires:      usermode >= 1.98

%description
Eclipse plugins to integrate OProfile's profiling capabilities with the CDT.

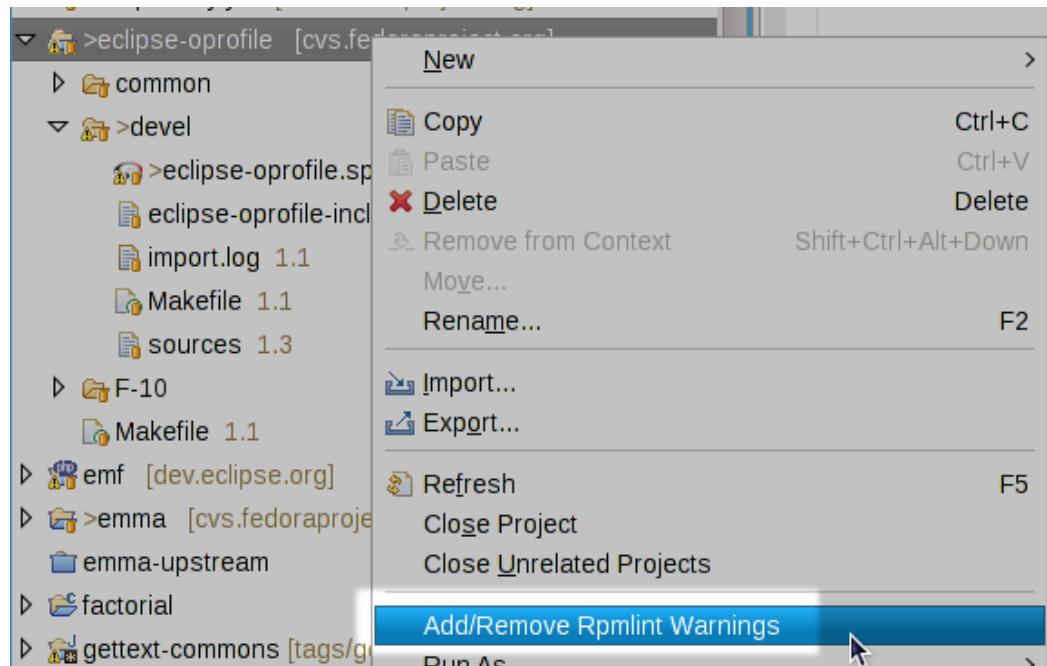
%prep
%setup -q -c
#remove binaries
rm -f org.eclipse.linuxtools.oprofile.core.linux.*/*os/linux/*/*opxml
```

Outline

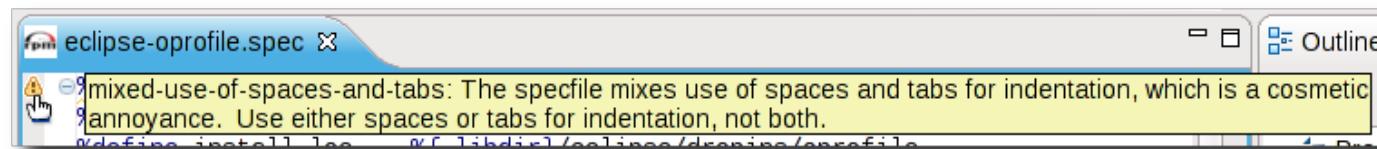
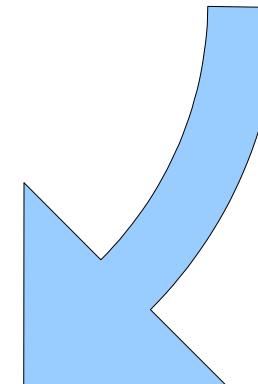
- ↳ Preamble
- ↳ Packages
- ↳ eclipse-oprofile
 - ↳ description
 - ↳ files
- ↳ prep
- ↳ build
- ↳ install
- ↳ clean
- ↳ changelog



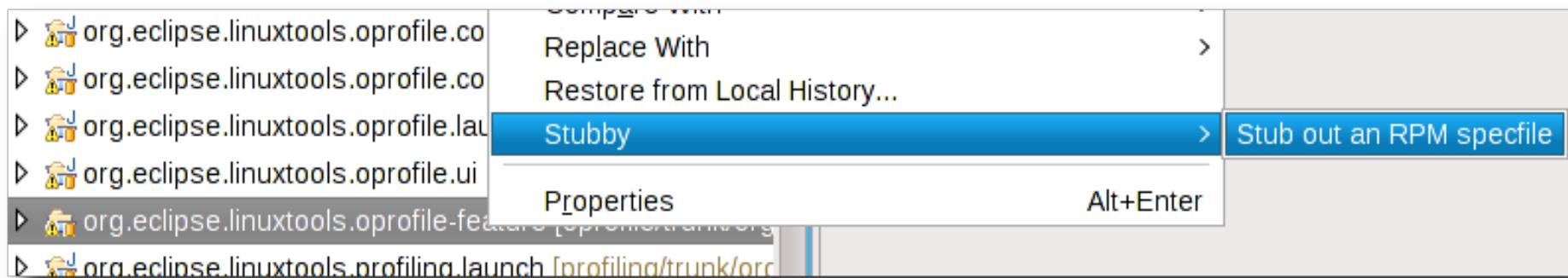
rpmlint warnings



Add/Remove rpmlint
Warnings



“RPM Stubby”



“Stub” out an RPM specfile

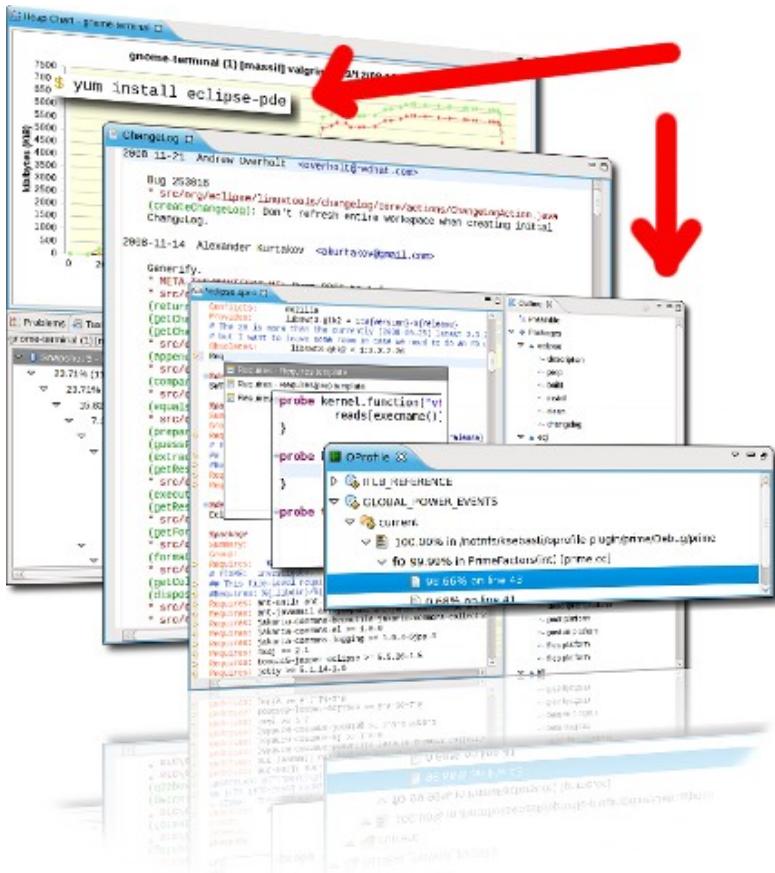


eclipse-build

Build harness for Eclipse SDK
used by a variety of Linux distributions



Linux Tools Project

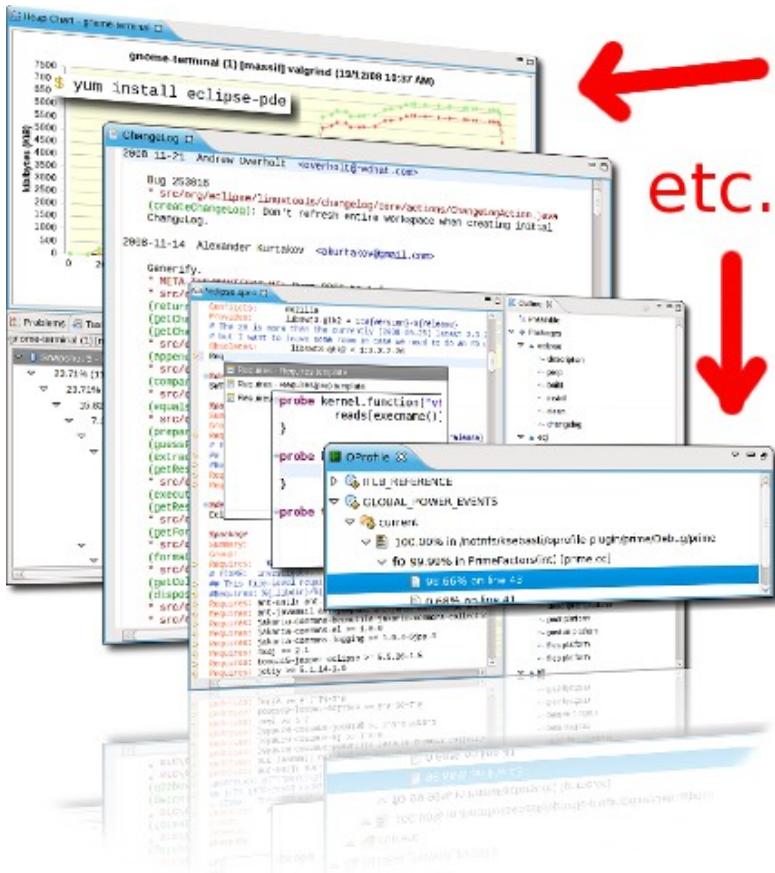


What we provide

- Tools for Linux packagers
- Tools for C/C++ developers built on CDT



Linux Tools Project



What we provide

- Tools for Linux packagers
- Tools for C/C++ developers built on CDT

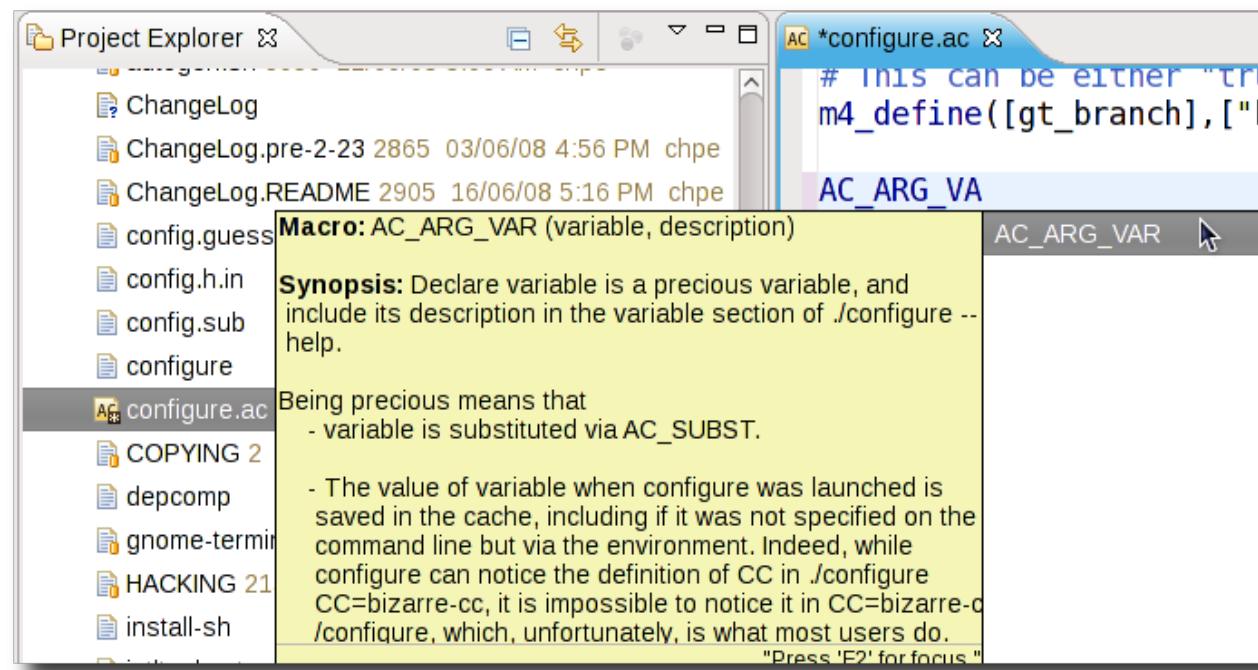


C/C++ Tools

- Customizable but good defaults
- Mainly developer-focused
- Integrate with CDT functionality
- Leverage existing Eclipse buy-in



GNU Autotools





Libhover

A screenshot of a code editor showing a tooltip for the `malloc` function. The code editor has a light blue background with syntax highlighting for C-like code. A tooltip window is overlaid on the editor, containing the following information:

glib.h
malloc (size_t size) void *

This function returns a pointer to a newly allocated block size bytes long, or a null pointer if the block could not be allocated.

The tooltip also lists other functions from the glib.h header:

- mallinfo (void) struct mallinfo
- mallinfo (void) struct mallinfo
- malloc (size_t nbytes) void *
- malloc (size_t size) void *** (highlighted)
- malloc_stats (void) void
- malloc_usable_size (void *aptr) size_t
- mallopt (int param, int value) int
- mallopt (int parameter, value) int

At the bottom of the tooltip, there is a message: "Press 'Ctrl+Space' to show Template Proposals".

```
void *blah = mall;

if (display_name
    display = gdk_d
else
{
    GSList *displ
    const char *p
    period = strr
    if (period)
    {
        gulong n;
```



OProfile

The screenshot shows the Eclipse IDE interface with the OProfile plugin. The top left window displays the C code for factorial.cpp, containing two implementations of factorial functions: a recursive factorial1 and an iterative factorial2. The bottom window shows the OProfile analysis results for the CPU_CLK_UNHALTED event, indicating that 100.00% of the time was spent in the factorial1 function. The analysis details further breakdowns of the execution time across specific lines of code.

factorial.cpp

```
if (n <= 1)
    return 1;
else
    return n * factorial1(n-1);

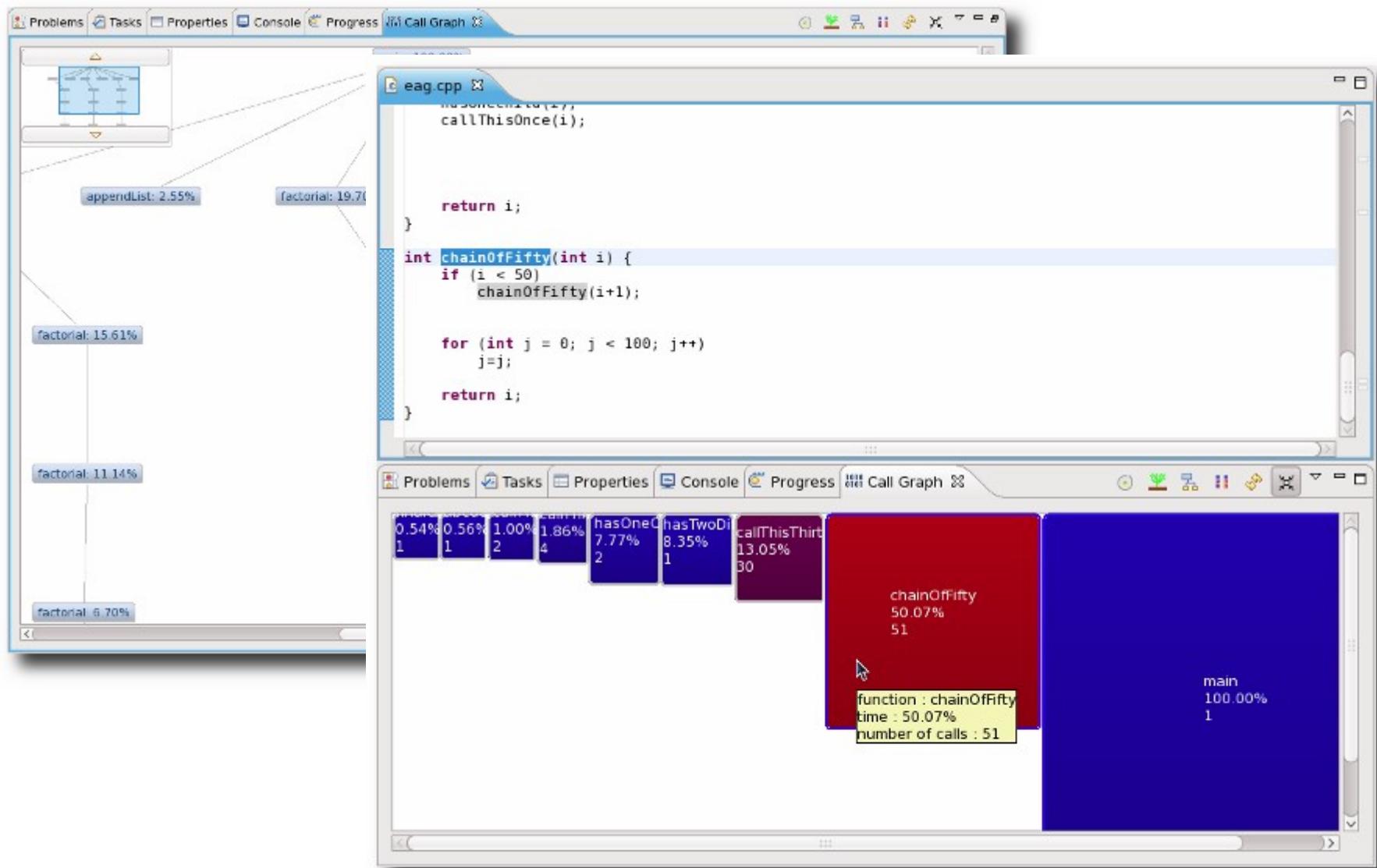
unsigned long long factorial2(unsigned
    unsigned long long ret = 1;
    for (unsigned int i = 1; i <= n; i++)
        ret *= i;
    return ret;
```

OProfile Analysis Results

- CPU_CLK_UNHALTED
 - current
 - 100.00% in /home/overholt/worksheets/runtime-EclipseApplication/factorial/D
 - fo 49.46% in factorial1(unsigned long long) [factorial.cpp]
 - 27.30% on line 15
 - 12.20% on line 16
 - 9.22% on line 11
 - 0.59% on line 12
 - 0.14% on line 13



C/C++ call graph





Valgrind

The screenshot shows a developer environment window titled "valgrindtest.c". Inside, there's a code editor with the following C code:

```
int *foo() {
    return (int *)malloc(SIZE);
}

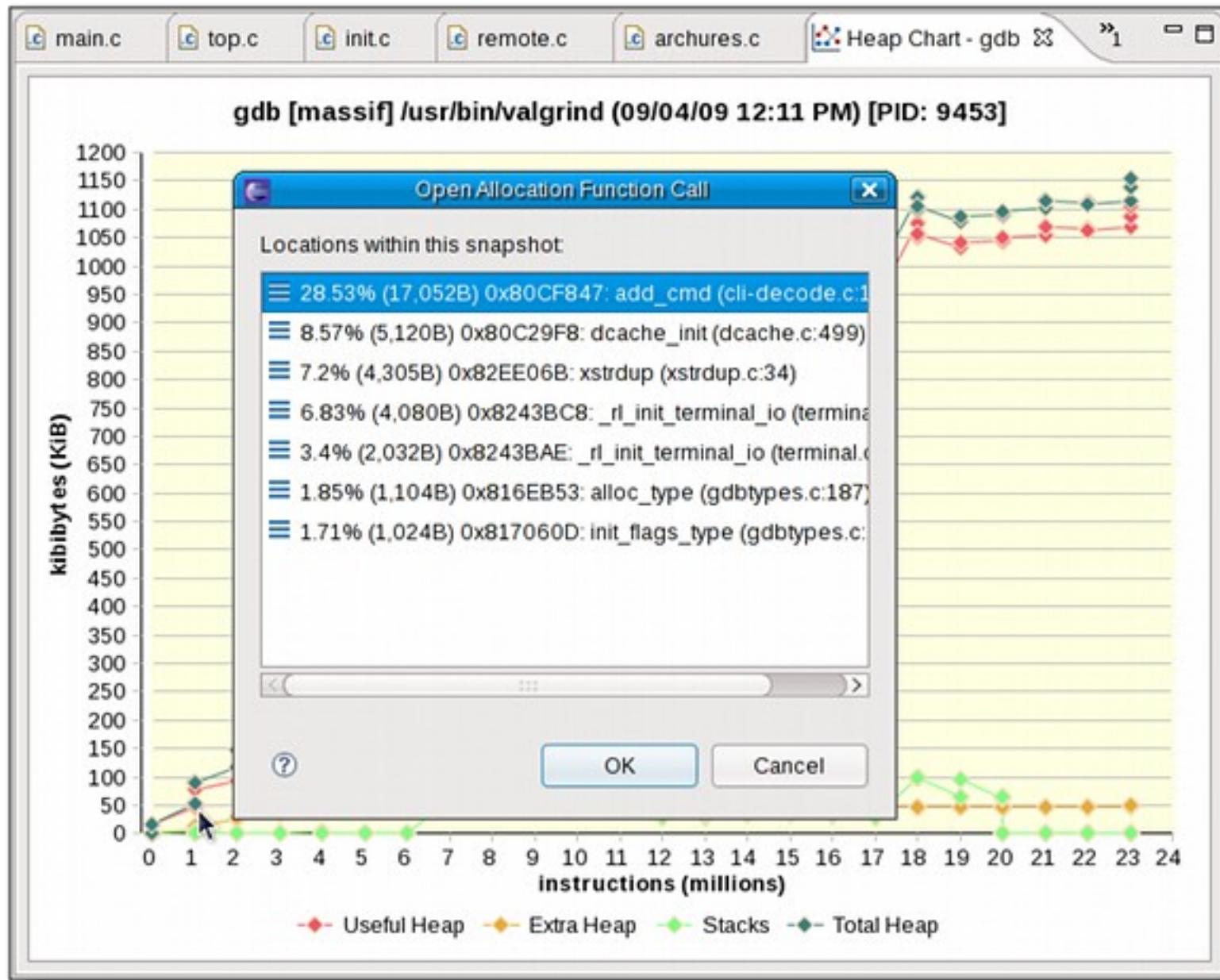
void bar(int *ptr) {
    free(ptr);
}
```

Below the code editor is a toolbar with tabs: Valgrind, Problems, Tasks, Console, Properties, Debug, and Progress. The "Console" tab is selected. The console output shows a memory leak detected by Valgrind:

```
valgrindtest (memcheck) [memcheck] /usr/bin/valgrind (17/03/09 3:07 PM)
  ▽ ✘ 4,000 bytes in 100 blocks are definitely lost in loss record 1 of 1 [pid: 28810 / tid: 1]
      └─ at 0x4A0739E: malloc (vg_replace_malloc.c:207)
          └─ by 0x400589: foo (valgrindtest.c:21)
              └─ by 0x400541: main (valgrindtest.c:12)
```



Valgrind





SystemTap

A screenshot of the SystemTap Studio interface. On the left, there is a tree view of probes categorized under 'Probe', 'Funct', and 'Kerne'. The 'Probe' tab is selected, showing nodes like '_signal', '_sunrpc' (with sub-nodes 'clnt' and 'create_client'), 'generic', 'ioblock', and 'ioscheduler'. The 'Funct' tab shows function signatures. The 'Kerne' tab shows kernel symbols. The main right pane displays a SystemTap script named 'counter2.stp'. The script contains the following code:

```
global read, write, start

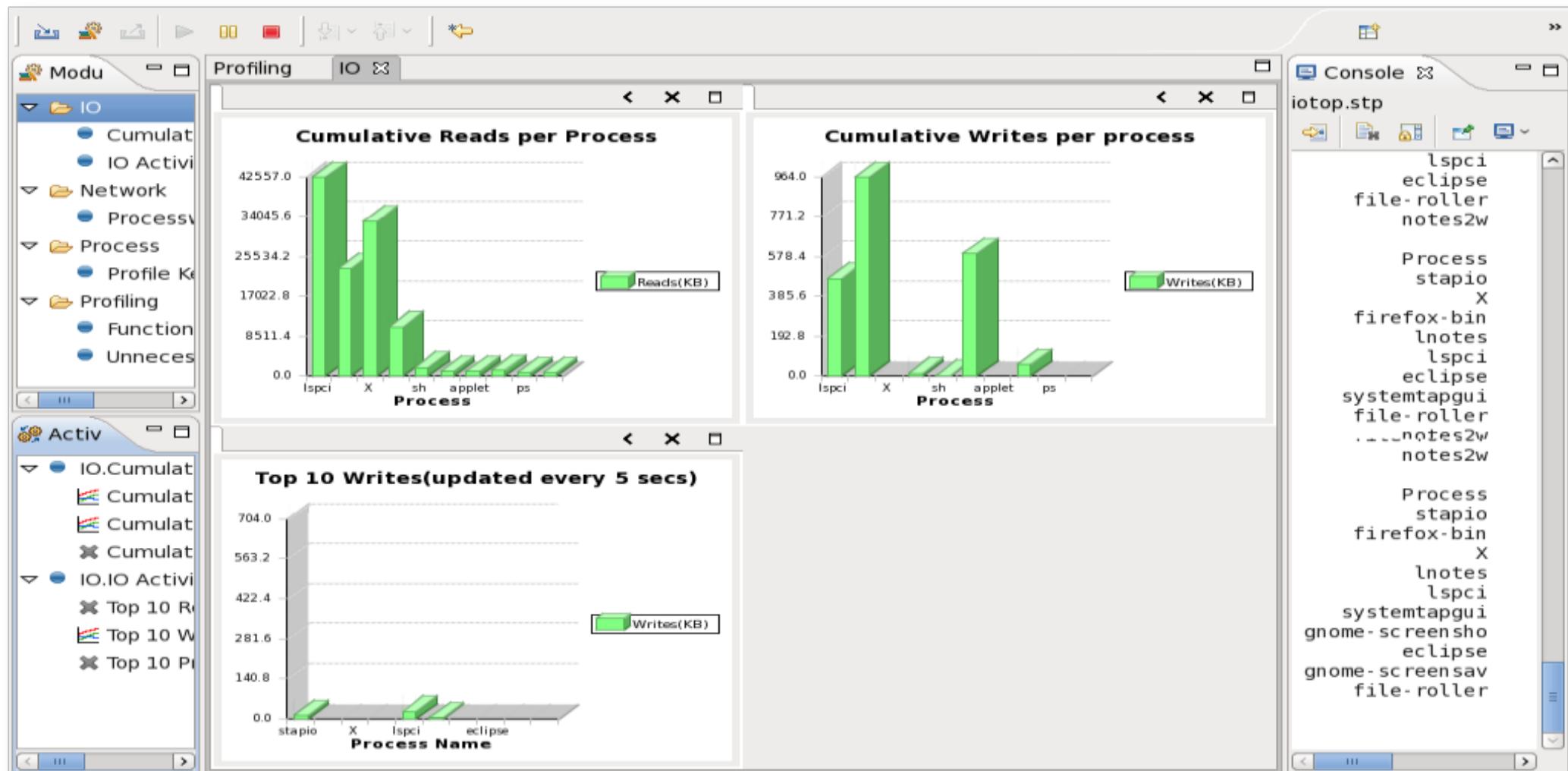
probe begin {
    start = gettimeofday_s()
}
probe syscall.write {
    write += count
}

probe timer.ms(1000) {
    printf("%d\t%d\t%d\n", (gettimeofday_s() - start),
        read=0
        write=0
}

probe syscall.read {
    read += count
}
```



SystemTap



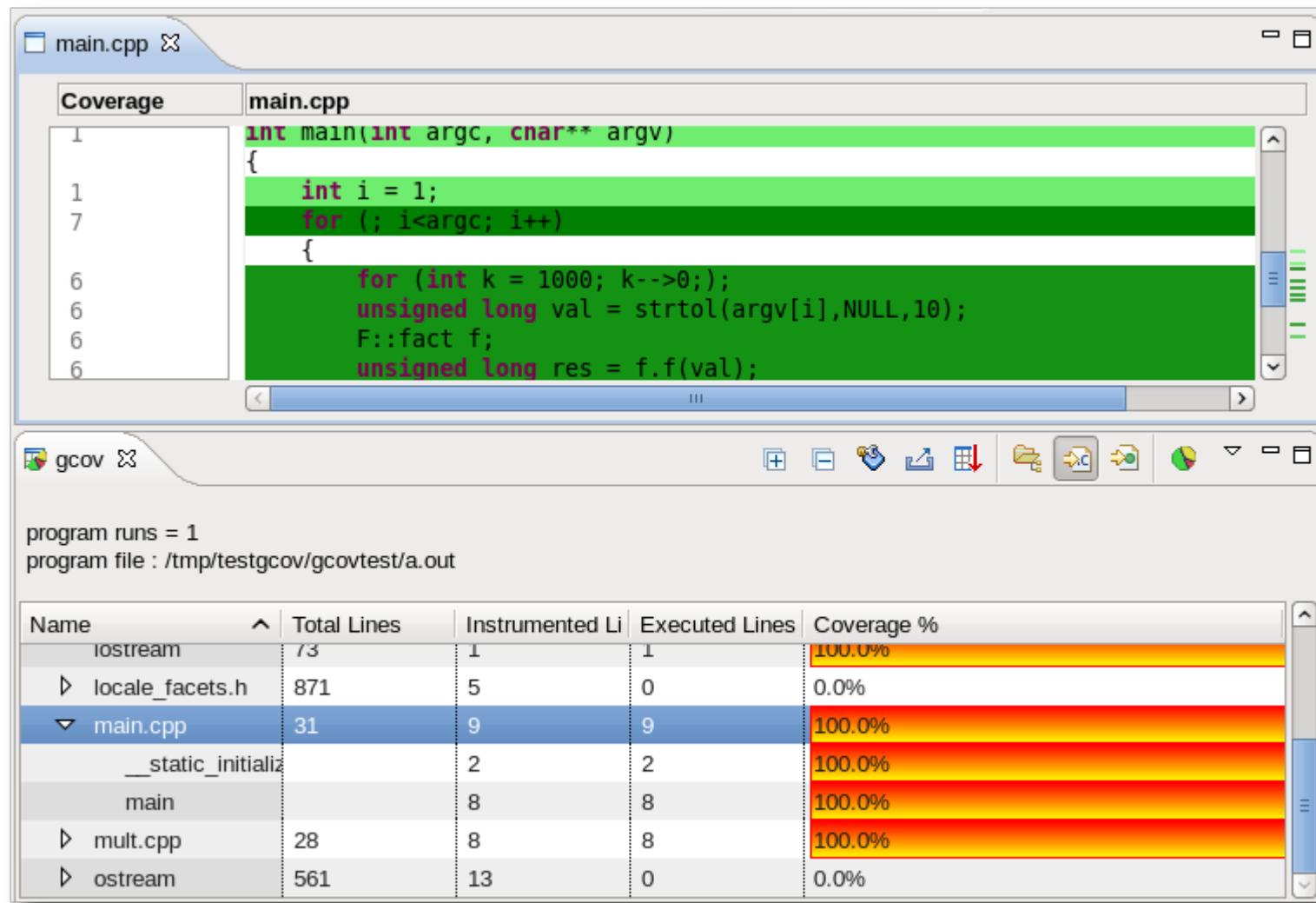


GProf

Name (location)	Samples	Calls	Time/Call	% Time
▼ Summary	216			100.0%
▼ foox.c	216			100.0%
▼ func_a	106	2	530.0ms	49.07%
▷ func_a (foox.c:38)	45			20.83%
▷ func_a (foox.c:42)	25			11.57%
▷ func_a (foox.c:45)	36			16.67%
▼ func_b	34	1	340.0ms	15.74%
▷ func_b (foox.c:52)	20			9.26%
▼ func_b (foox.c:54)	14			6.48%
0x804888c	1			0.46%
0x8048890	2			0.93%
0x8048898	2			0.93%
0x804889c	4			1.85%
0x80488a0	5			2.31%
▼ func_f	43	2	215.0ms	19.91%
▷ func_f (foox.c:31)	43			19.91%
▼ main	33	0		15.28%
▷ main (foox.c:61)	12			5.56%
▷ main (foox.c:63)	13			6.02%
▼ main (foox.c:65)	8			3.7%
0x8048a8c	1			0.46%
0x8048a90	1			0.46%
0x8048a9c	3			1.39%
0x8048aa0	1			0.46%
0x8048aac	1			0.46%
0x8048ab4	1			0.46%

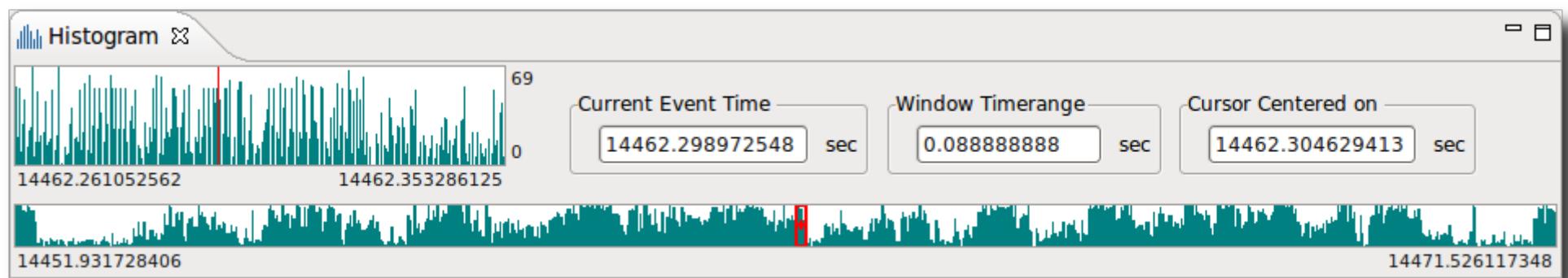


GCov

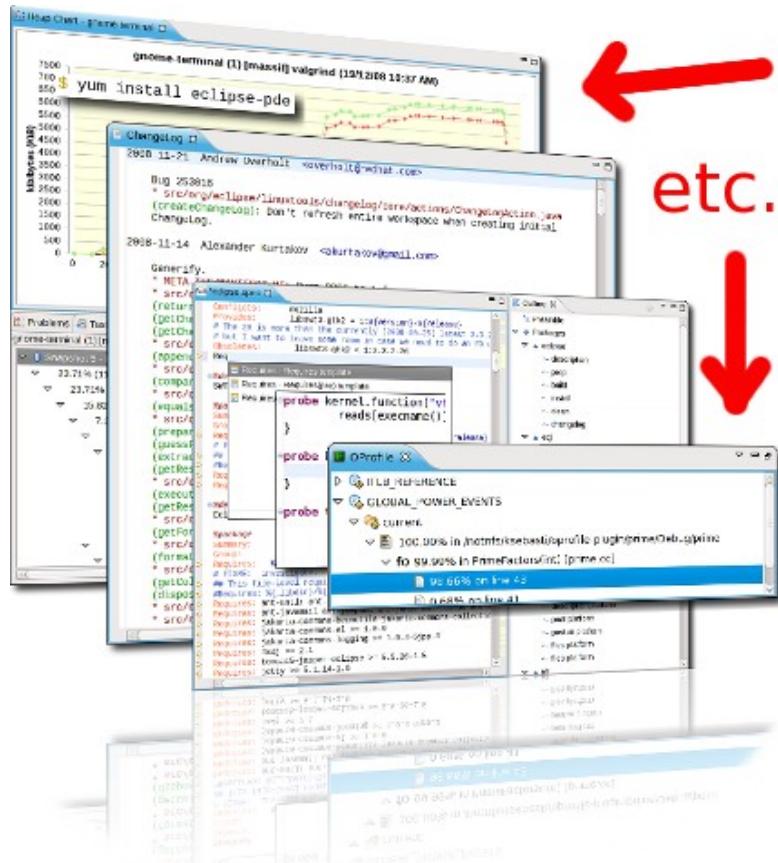




LTTng



Linux Tools Project

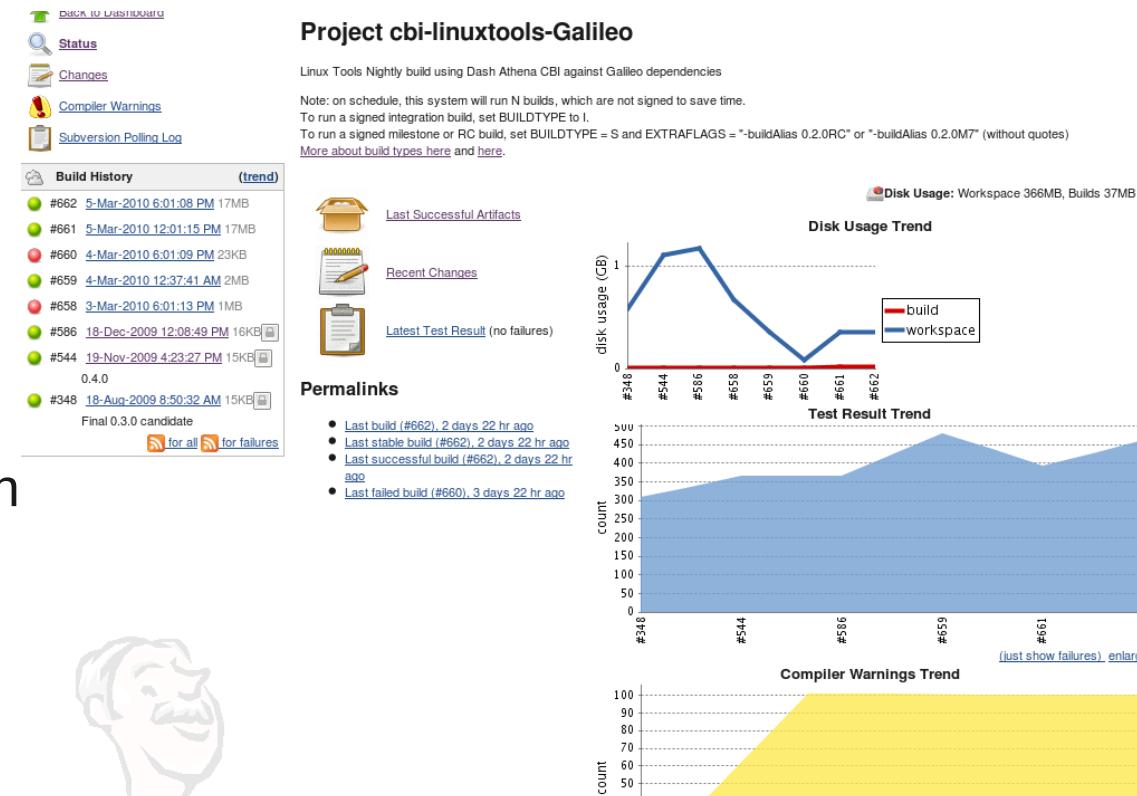


What we provide

- Tools for Linux packagers
- Tools for C/C++ developers built on CDT

Project Details

- **Web:** <http://eclipse.org/linuxtools>
- **Wiki:** http://wiki.eclipse.org/index.php/Linux_Tools_Project
- **SVN:** <svn://dev.eclipse.org/svnroot/technology/org.eclipse.linuxtools>
- **Hudson:**
- <https://build.eclipse.org/hudson>
soon to be:
<https://hudson.eclipse.org/hudson>





Current Committers

- Ericsson
- IBM
- Individuals
- Red Hat
- ST Microelectronics



Ways to contribute

- Unit testing and test coverage
- Functional and integration testing
- User guides
- Automation (release engineering, docs, etc.)
- Bug fixing
- Feature authoring



How to get involved

`linuxtools-dev@eclipse.org`

`http://eclipse.org/linuxtools`

`#eclipse-linux` on Freenode

`https://bugs.eclipse.org/bugs/enter_bug.cgi?product=Linux%20Tools`