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
Name: eclipse-oprofile
Version: 0.1.0
Release: %{?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: %{_topdir}/%{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

%build
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
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

etc.
C/C++ Tools

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

**Synopsis:** Declare variable as a precious variable, and include its description in the variable section of ./configure --help.

Being precious means that:
- variable is substituted via AC_SUBST.
- The value of variable when configure was launched is saved in the cache, including if it was not specified on the command line but via the environment. Indeed, while configure can notice the definition of CC in ./configure CC=bizarre-cc, it is impossible to notice it in CC=bizarre-cc ./configure, which, unfortunately, is what most users do.

```
m4_define([gt_branch],["b"
```

**Macro:** AC_ARG_VAR (variable, description)
OProfile
C/C++ call graph
Valgrind

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

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

Error message:

- 4,000 bytes in 100 blocks are definitely lost in loss record 1 of 1 [pid: 28810 / tid: 1]
  - at 0x4A073E: malloc (vg_replace_malloc.c:207)
  - by 0x400589: foo (valgrindtest.c:21)
  - by 0x400541: main (valgrindtest.c:12)
Valgrind

![Valgrind heap chart](image)
SystemTap

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
<table>
<thead>
<tr>
<th>Name (location)</th>
<th>Samples</th>
<th>Calls</th>
<th>Time/Call</th>
<th>% Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>216</td>
<td></td>
<td></td>
<td>100.0%</td>
</tr>
<tr>
<td>func_a</td>
<td>216</td>
<td>106</td>
<td>530.0ms</td>
<td>49.07%</td>
</tr>
<tr>
<td>func_a (foox.c:38)</td>
<td>45</td>
<td>2</td>
<td></td>
<td>20.83%</td>
</tr>
<tr>
<td>func_a (foox.c:42)</td>
<td>25</td>
<td>1</td>
<td></td>
<td>11.57%</td>
</tr>
<tr>
<td>func_a (foox.c:45)</td>
<td>36</td>
<td>1</td>
<td></td>
<td>16.67%</td>
</tr>
<tr>
<td>func_b</td>
<td>34</td>
<td>34</td>
<td>340.0ms</td>
<td>15.74%</td>
</tr>
<tr>
<td>func_b (foox.c:52)</td>
<td>20</td>
<td>1</td>
<td></td>
<td>9.26%</td>
</tr>
<tr>
<td>func_b (foox.c:54)</td>
<td>14</td>
<td>1</td>
<td></td>
<td>6.48%</td>
</tr>
<tr>
<td>0x804888c</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0.46%</td>
</tr>
<tr>
<td>0x8048890</td>
<td>2</td>
<td>2</td>
<td></td>
<td>0.93%</td>
</tr>
<tr>
<td>0x8048898</td>
<td>2</td>
<td>2</td>
<td></td>
<td>0.93%</td>
</tr>
<tr>
<td>0x804889c</td>
<td>4</td>
<td>4</td>
<td></td>
<td>1.85%</td>
</tr>
<tr>
<td>0x80488a0</td>
<td>5</td>
<td>5</td>
<td></td>
<td>3.31%</td>
</tr>
<tr>
<td>func_f</td>
<td>43</td>
<td>43</td>
<td>215.0ms</td>
<td>19.91%</td>
</tr>
<tr>
<td>func_f (foox.c:31)</td>
<td>43</td>
<td>2</td>
<td></td>
<td>19.91%</td>
</tr>
<tr>
<td>main</td>
<td>33</td>
<td>33</td>
<td></td>
<td>15.28%</td>
</tr>
<tr>
<td>main (foox.c:61)</td>
<td>12</td>
<td>1</td>
<td></td>
<td>5.56%</td>
</tr>
<tr>
<td>main (foox.c:63)</td>
<td>13</td>
<td>1</td>
<td></td>
<td>6.02%</td>
</tr>
<tr>
<td>main (foox.c:65)</td>
<td>8</td>
<td>8</td>
<td></td>
<td>3.7%</td>
</tr>
<tr>
<td>0x8048a8c</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0.46%</td>
</tr>
<tr>
<td>0x8048a90</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0.46%</td>
</tr>
<tr>
<td>0x8048a9c</td>
<td>3</td>
<td>3</td>
<td></td>
<td>1.39%</td>
</tr>
<tr>
<td>0x8048aa0</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0.46%</td>
</tr>
<tr>
<td>0x8048aac</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0.46%</td>
</tr>
<tr>
<td>0x8048ab4</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0.46%</td>
</tr>
</tbody>
</table>
```
int main(int argc, char** argv)
{
    int i = 1;
    for (; i<argc; i++)
    {
        for (int k = 1000; k-->0;)
        {
            unsigned long val = strtol(argv[i], NULL, 10);
            F::fact f;
            unsigned long res = f.f(val);
        }
    }
}
```

program runs = 1
program file: /tmp/testgcov/gcovtest/a.out

<table>
<thead>
<tr>
<th>Name</th>
<th>Total Lines</th>
<th>Instrumented L</th>
<th>Executed L</th>
<th>Coverage %</th>
</tr>
</thead>
<tbody>
<tr>
<td>iostream</td>
<td>73</td>
<td>1</td>
<td>1</td>
<td>100.0%</td>
</tr>
<tr>
<td>locale_facets.h</td>
<td>871</td>
<td>5</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td>main.cpp</td>
<td>31</td>
<td>9</td>
<td>9</td>
<td>100.0%</td>
</tr>
<tr>
<td>__static_initial</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>main</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>100.0%</td>
</tr>
<tr>
<td>mult.cpp</td>
<td>28</td>
<td>8</td>
<td>8</td>
<td>100.0%</td>
</tr>
<tr>
<td>ostream</td>
<td>561</td>
<td>13</td>
<td>0</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
LTTng
Linux Tools Project

What we provide

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

etc.
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
  - 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 Tools