uftrace: A function graph tracer for C/C++ userspace programs

Lightning Talk at CppCon 2016

September 23, 2016

Namhyung Kim, **Honggyu Kim**

LG Electronics

{namhyung.kim, hong.gyu.kim}@lge.com

uftrace

https://github.com/namhyung/uftrace

```
int main() {

}
```

```
void foo() {

}
int main() {
   foo();
}
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
$ gcc test.c
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
$ gcc test.c
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
<bar>:

  ret



<foo>:

  call <bar>
  ret



<main>:

  call <foo>
  ret
```

```
$ gcc -pg test.c
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
<bar>:
  call <mcount@plt>
  ret


<foo>:
  call <mcount@plt>
  call <bar>
  ret


<main>:
  call <mcount@plt>
  call <foo>
  ret
```

```
$ gcc -pg test.c
$ gcc -pg test.c
$ uftrace record a.out
$ gcc -pg test.c
$ uftrace record a.out
$ uftrace replay
$ gcc -pg test.c
$ uftrace record a.out
$ uftrace replay
# DURATION    TID     FUNCTION
   0.531 us [21315] | __monstartup();
   0.435 us [21315] | __cxa_atexit();
            [21315] | main() {
            [21315] |   foo() {
   0.134 us [21315] |     bar();
   0.564 us [21315] |   } /* foo */
   0.890 us [21315] | } /* main */
```

```
$ gcc -pg test.c

$ uftrace live a.out
# DURATION    TID     FUNCTION
   0.531 us [21315] | __monstartup();
   0.435 us [21315] | __cxa_atexit();
            [21315] | main() {
            [21315] |   foo() {
   0.134 us [21315] |     bar();
   0.564 us [21315] |   } /* foo */
   0.890 us [21315] | } /* main */
```

```
$ gcc -pg test.c

$ uftrace a.out
# DURATION    TID     FUNCTION
   0.531 us [21315] | __monstartup();
   0.435 us [21315] | __cxa_atexit();
            [21315] | main() {
            [21315] |   foo() {
   0.134 us [21315] |     bar();
   0.564 us [21315] |   } /* foo */
   0.890 us [21315] | } /* main */
```

```
$ gcc -pg test.c

$ uftrace a.out
# DURATION    TID     FUNCTION
   0.531 us [21315] | __monstartup();
   0.435 us [21315] | __cxa_atexit();
            [21315] | main() {
            [21315] |   foo() {
   0.134 us [21315] |     bar();
   0.564 us [21315] |   } /* foo */
   0.890 us [21315] | } /* main */
```

```
$ gcc -pg test.c

$ uftrace a.out
# DURATION     TID       FUNCTION
   0.531 us [21315] | __monstartup();
   0.435 us [21315] | __cxa_atexit();
            [21315] | main() {
            [21315] |   foo() {
   0.134 us [21315] |     bar();
   0.564 us [21315] |   } /* foo */
   0.890 us [21315] | } /* main */
```

```
$ gcc -pg test.c

$ uftrace a.out
# DURATION     TID      FUNCTION
   0.531 us [21315] | __monstartup();
   0.435 us [21315] | __cxa_atexit();
            [21315] | main() {
            [21315] |   foo() {
   0.134 us [21315] |     bar();
   0.564 us [21315] |   } /* foo */
   0.890 us [21315] | } /* main */
```

```
$ gcc -pg test.c

$ uftrace a.out
# DURATION    TID      FUNCTION
   0.531 us [21315] | __monstartup();
   0.435 us [21315] | __cxa_atexit();
            [21315] | main() {
            [21315] |   foo() {
   0.134 us [21315] |     bar();
   0.564 us [21315] |   } /* foo */
   0.890 us [21315] | } /* main */
```

```
$ gcc -pg test.c

$ uftrace -t 200ns a.out
# DURATION    TID     FUNCTION
   0.531 us [21315] | __monstartup();
   0.435 us [21315] | __cxa_atexit();
            [21315] | main() {
            [21315] |   foo() {
   0.134 us [21315] |     bar();
   0.564 us [21315] |   } /* foo */
   0.890 us [21315] | } /* main */
```

```
-t TIME, --time-filter=TIME
    Do not show small functions under the
    time threshold.
```

```
$ gcc -pg test.c

$ uftrace -t 200ns a.out
# DURATION    TID     FUNCTION
   0.616 us [32476] | __monstartup();
   0.411 us [32476] | __cxa_atexit();
            [32476] | main() {
   0.427 us [32476] |   foo();
   0.825 us [32476] | } /* main */
```

```
-t TIME, --time-filter=TIME
    Do not show small functions under the
    time threshold.
```

```
$ gcc -pg test.c
$ uftrace record a.out
$ gcc -pg test.c
$ uftrace record a.out
$ uftrace report
$ gcc -pg test.c
$ uftrace record a.out
$ uftrace report

Total time   Self time       Calls  Function
==========  ==========  ==========  ====================
  0.890 us    0.326 us           1  main
  0.564 us    0.430 us           1  foo
  0.531 us    0.531 us           1  __monstartup
  0.435 us    0.435 us           1  __cxa_atexit
  0.134 us    0.134 us           1  bar
```

```
$ gcc -pg fibonacci.c
$ gcc -pg fibonacci.c
$ uftrace fibonacci 5
fib(5) = 5
```

```
$ gcc -pg fibonacci.c
$ uftrace fibonacci 5
fib(5) = 5
# DURATION    TID     FUNCTION
   0.620 us [31321] | __monstartup();
   0.456 us [31321] | __cxa_atexit();
            [31321] | main() {
   1.478 us [31321] |   atoi();
            [31321] |   fib() {
            [31321] |     fib() {
            [31321] |       fib() {
   0.155 us [31321] |         fib();
   0.123 us [31321] |         fib();
   0.883 us [31321] |       } /* fib */
   0.125 us [31321] |       fib();
   1.483 us [31321] |     } /* fib */
            [31321] |     fib() {
   0.125 us [31321] |       fib();
   0.125 us [31321] |       fib();
   0.774 us [31321] |     } /* fib */
   2.716 us [31321] |   } /* fib */
   4.382 us [31321] |   printf();
   9.456 us [31321] | } /* main */
```

```
$ gcc -pg fibonacci.c
$ uftrace -A fib@arg1 fibonacci 5
fib(5) = 5
# DURATION    TID     FUNCTION
   0.770 us [31365] | __monstartup();
   0.492 us [31365] | __cxa_atexit();
            [31365] | main() {
   1.507 us [31365] |   atoi();
            [31365] |   fib(5) {
            [31365] |     fib(4) {
            [31365] |       fib(3) {
   1.293 us [31365] |         fib(2);
   0.172 us [31365] |         fib(1);
   2.295 us [31365] |       } /* fib */
   0.157 us [31365] |       fib(2);
   3.025 us [31365] |     } /* fib */
            [31365] |     fib(3) {
   0.150 us [31365] |       fib(2);
   0.155 us [31365] |       fib(1);
   0.917 us [31365] |     } /* fib */
   5.232 us [31365] |   } /* fib */
   4.856 us [31365] |   printf();
  12.697 us [31365] | } /* main */
```

```
$ gcc -pg fibonacci.c
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
fib(5) = 5
# DURATION    TID     FUNCTION
   0.718 us [31379] | __monstartup();
   0.464 us [31379] | __cxa_atexit();
            [31379] | main() {
   1.442 us [31379] |   atoi();
            [31379] |   fib(5) {
            [31379] |     fib(4) {
            [31379] |       fib(3) {
   1.395 us [31379] |         fib(2) = 1;
   0.174 us [31379] |         fib(1) = 1;
   2.562 us [31379] |       } = 2; /* fib */
   0.157 us [31379] |       fib(2) = 1;
   3.330 us [31379] |     } = 3; /* fib */
            [31379] |     fib(3) {
   0.152 us [31379] |       fib(2) = 1;
   0.154 us [31379] |       fib(1) = 1;
   0.959 us [31379] |     } = 2; /* fib */
   5.351 us [31379] |   } = 5; /* fib */
   5.729 us [31379] |   printf();
  13.627 us [31379] | } /* main */
```

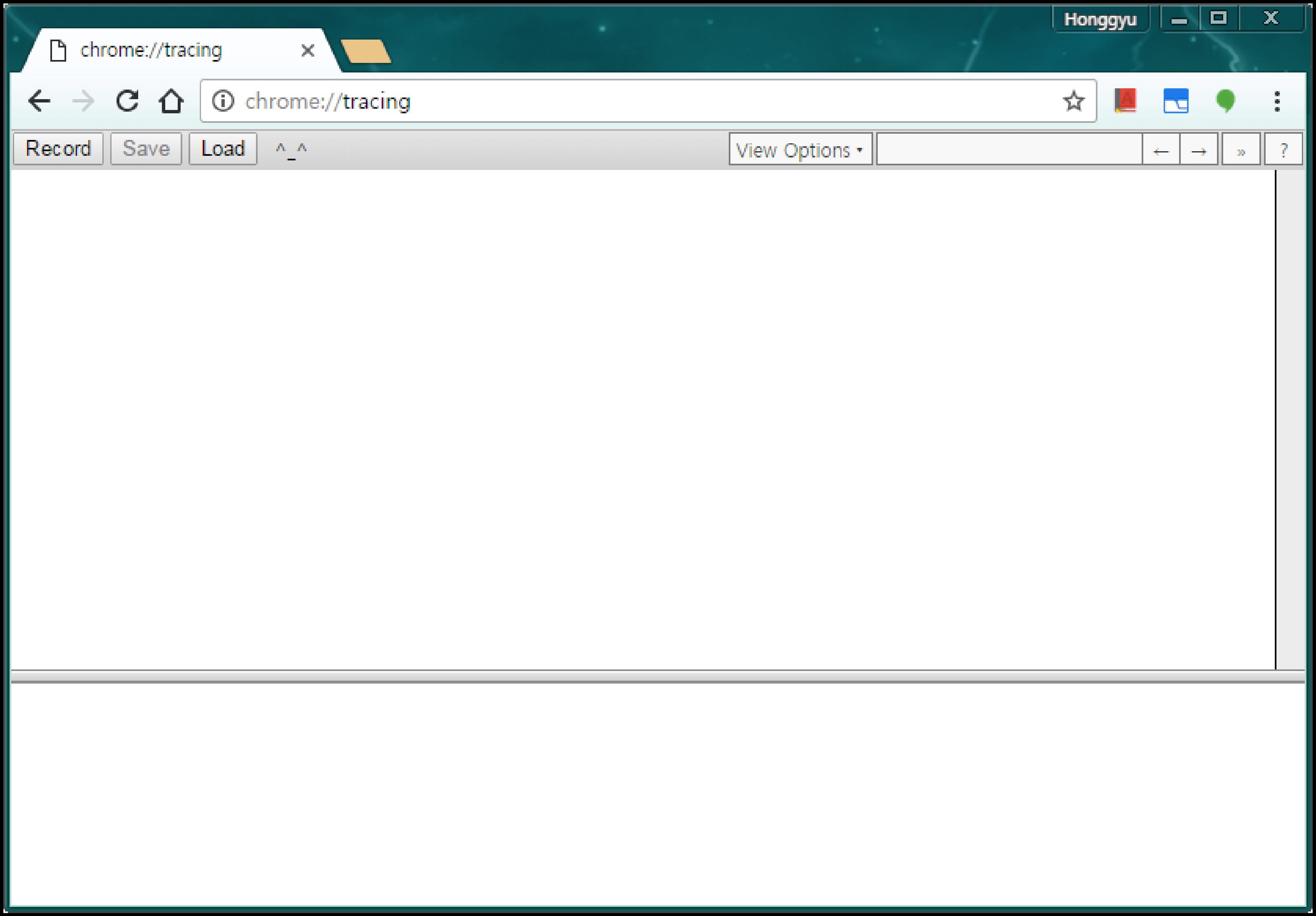
```
$ gcc -pg fibonacci.c
$ gcc -pg fibonacci.c
$ uftrace record fibonacci 5
fib(5) = 5
```

```
$ gcc -pg fibonacci.c
$ uftrace record fibonacci 5
fib(5) = 5
$ uftrace dump
$ gcc -pg fibonacci.c
$ uftrace record fibonacci 5
fib(5) = 5
$ uftrace dump --chrome
```

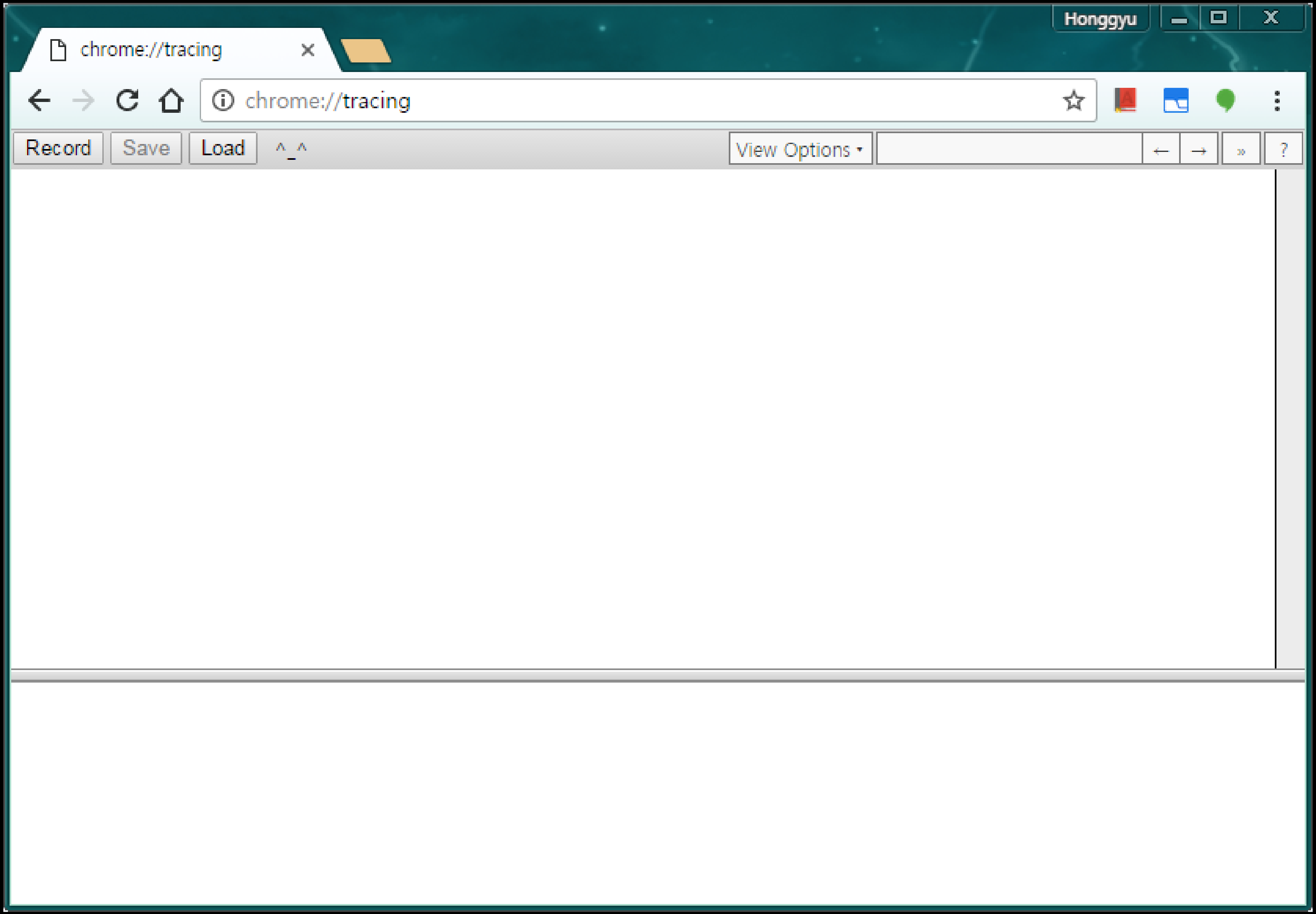
```
  $ gcc -pg fibonacci.c
  $ uftrace record fibonacci 5
  fib(5) = 5
  $ uftrace dump --chrome

{"traceEvents":[
{"ts":5913706403443,"ph":"B","pid":32256,"name":"__monstartup"},
{"ts":5913706403444,"ph":"E","pid":32256,"name":"__monstartup"},
{"ts":5913706403447,"ph":"B","pid":32256,"name":"__cxa_atexit"},
{"ts":5913706403447,"ph":"E","pid":32256,"name":"__cxa_atexit"},
{"ts":5913706403448,"ph":"B","pid":32256,"name":"main"},
{"ts":5913706403448,"ph":"B","pid":32256,"name":"atoi"},
{"ts":5913706403450,"ph":"E","pid":32256,"name":"atoi"},
{"ts":5913706403450,"ph":"B","pid":32256,"name":"fib"},
{"ts":5913706403450,"ph":"B","pid":32256,"name":"fib"},
             ...
{"ts":5913706403452,"ph":"E","pid":32256,"name":"fib"},
{"ts":5913706403453,"ph":"E","pid":32256,"name":"fib"},
{"ts":5913706403453,"ph":"E","pid":32256,"name":"fib"},
{"ts":5913706403453,"ph":"B","pid":32256,"name":"printf"},
{"ts":5913706403457,"ph":"E","pid":32256,"name":"printf"},
{"ts":5913706403458,"ph":"E","pid":32256,"name":"main"}
], "metadata": {
"command_line":"uftrace record fibonacci 5 ",
"recorded_time":"Thu Sep 22 22:31:17 2016"
} }
```

```
$ gcc -pg fibonacci.c
$ uftrace record fibonacci 5
fib(5) = 5
$ uftrace dump --chrome > fib.json
```

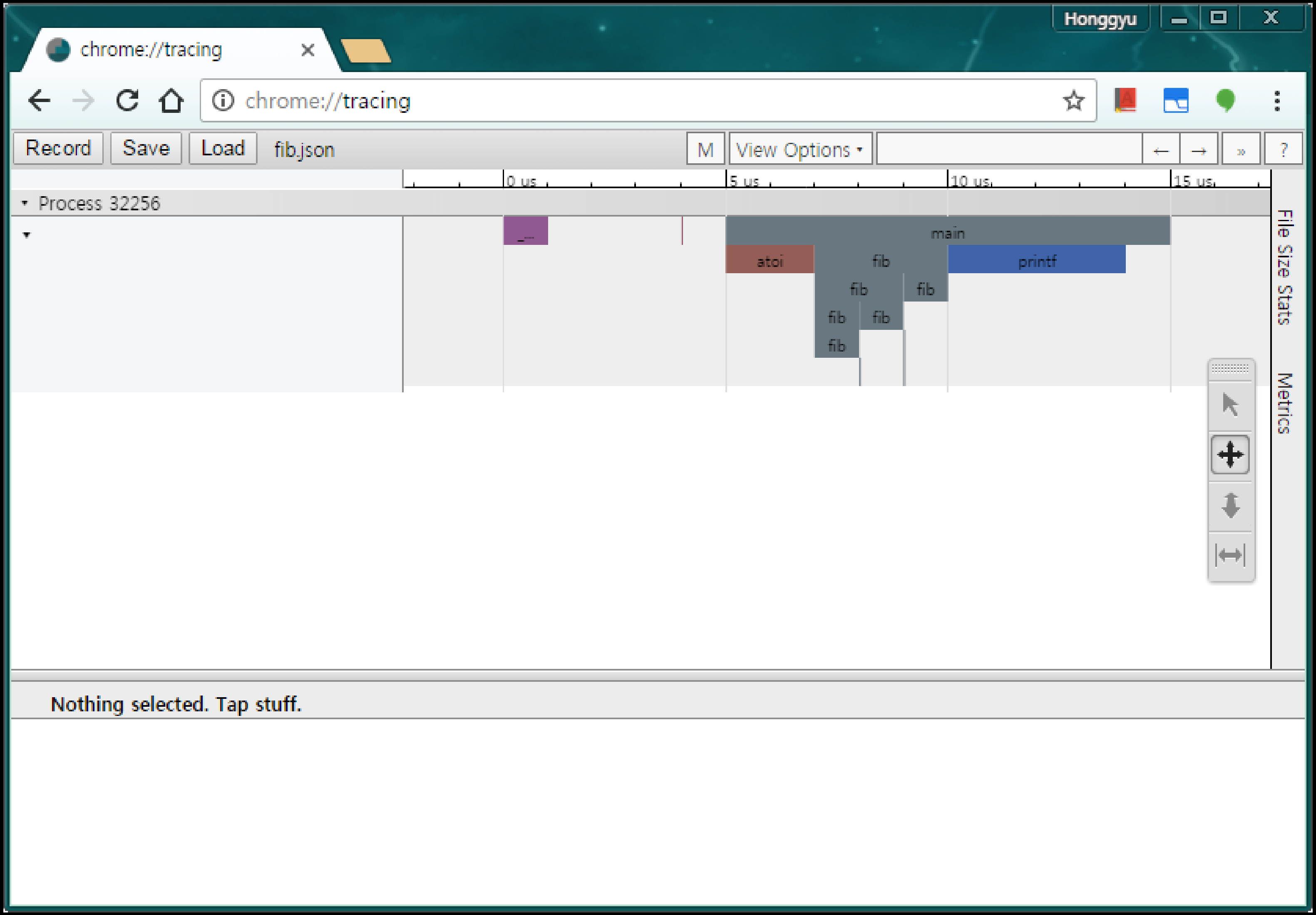
1. Open Chrome Browser



1. Open Chrome Browser
2. Load JSON file in chrome://tracing

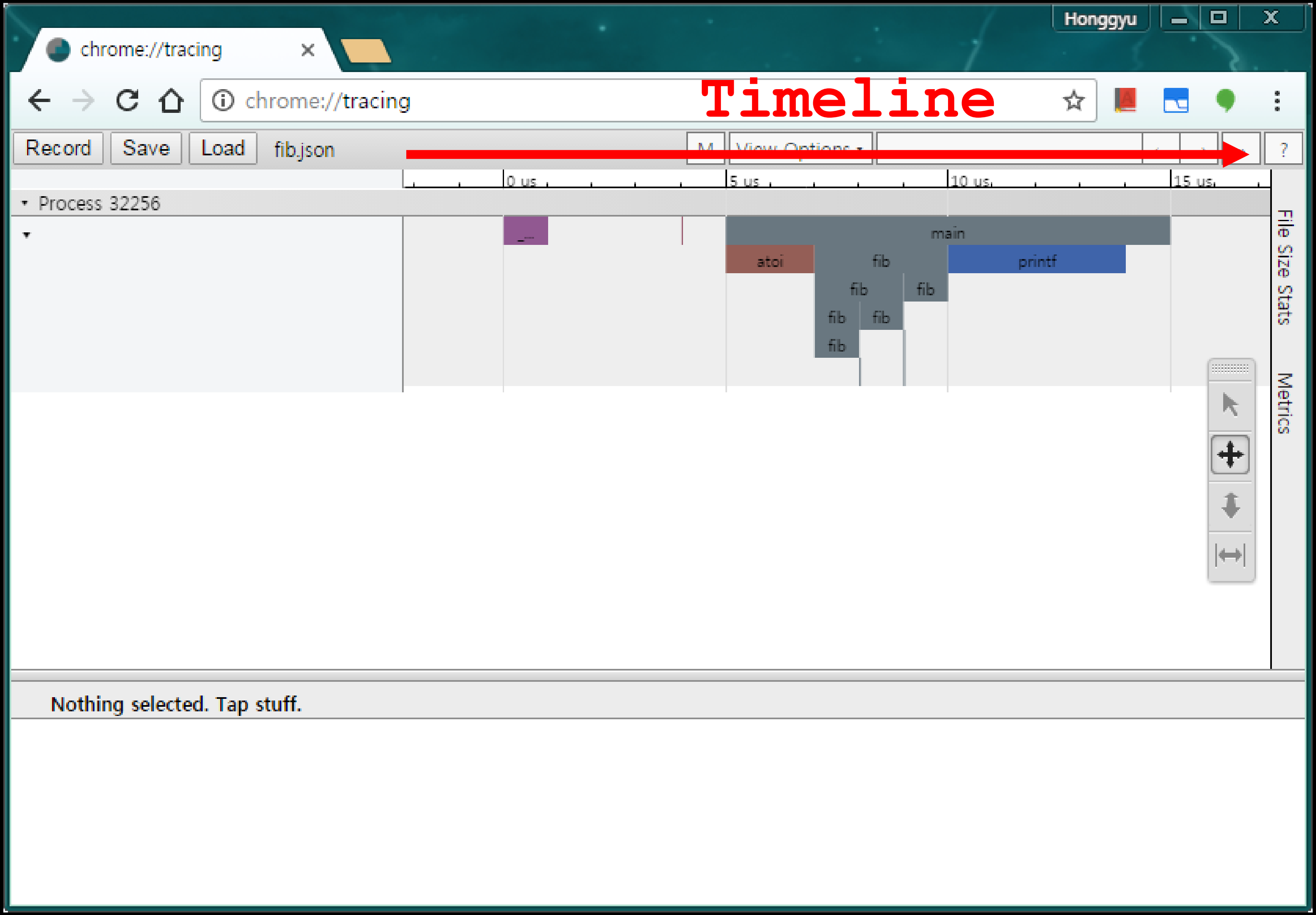


1. Open Chrome Browser
2. Load JSON file in chrome://tracing



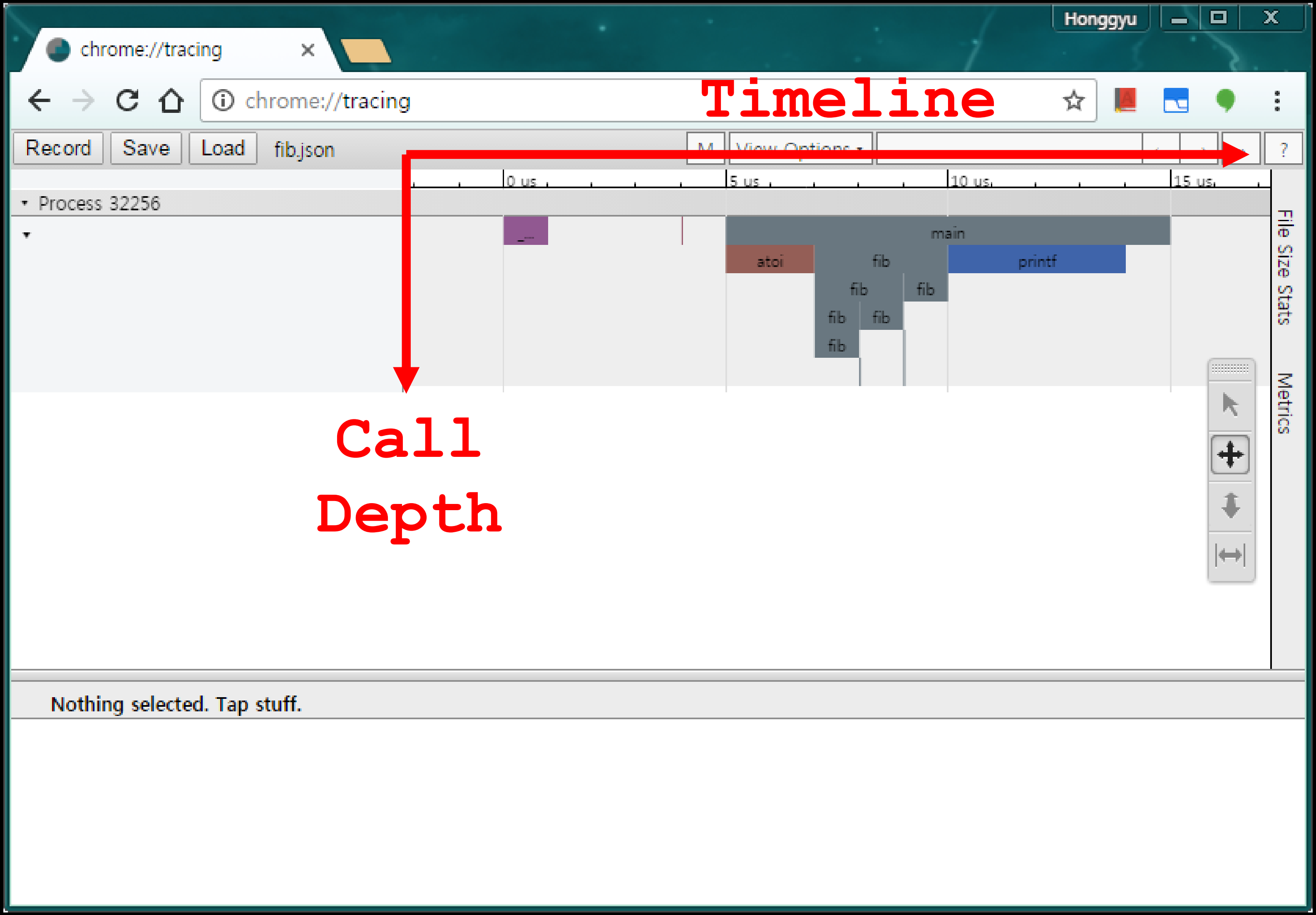
Nothing selected. Tap stuff.

1. Open Chrome Browser
2. Load JSON file in chrome://tracing



Nothing selected. Tap stuff.

1. Open Chrome Browser
2. Load JSON file in chrome://tracing



Nothing selected. Tap stuff.

Analyzing STL (shared_ptr)

Analyzing STL (shared_ptr)

```
int main()
{
  shared_ptr<int> s1(new int);



}
```

Analyzing STL (shared_ptr)

```
int main()
{
  shared_ptr<int> s1(new int);
  {
    shared_ptr<int> s2 = s1;
  }
}
```

Analyzing STL (shared_ptr)

```
$ g++ -pg shared_ptr.cc
```

```
int main()
{
  shared_ptr<int> s1(new int);
  {
    shared_ptr<int> s2 = s1;
  }
}
```

Analyzing STL (shared_ptr)

```
$ g++ -pg shared_ptr.cc
$ uftrace a.out
```

```
int main()
{
  shared_ptr<int> s1(new int);
  {
    shared_ptr<int> s2 = s1;
  }
}
```

Analyzing STL (shared_ptr)

```
$ g++ -pg shared_ptr.cc
$ uftrace -D .. -F .. -A .. -R .. a.out
```

```
int main()
{
  shared_ptr<int> s1(new int);
  {
    shared_ptr<int> s2 = s1;
  }
}
```

```
$ uftrace -D 4 \
        -F main -F "operator .*" -F "std::shared_ptr::.*" \
        -A "operator new"@arg1 -R "operator new"@retval \
        -A "operator delete"@arg1 \
        shared_ptr
```

```
# DURATION     TID     FUNCTION
            [10471] | main() {
   2.335 us [10471] |   operator new(4) = 0x1209910;
            [10471] |   std::shared_ptr::shared_ptr() {
            [10471] |     std::__shared_ptr::__shared_ptr() {
            [10471] |       std::__shared_count::__shared_count() {
   2.860 us [10471] |         operator new(24) = 0x122d630;
   0.456 us [10471] |         std::_Sp_counted_ptr::_Sp_counted_ptr();
   4.907 us [10471] |       } /* std::__shared_count::__shared_count */
   0.163 us [10471] |       std::__enable_shared_from_this_helper();
   5.982 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
   6.450 us [10471] |   } /* std::shared_ptr::shared_ptr */
            [10471] |   std::shared_ptr::shared_ptr() {
            [10471] |     std::__shared_ptr::__shared_ptr() {
            [10471] |       std::__shared_count::__shared_count() {
   0.649 us [10471] |         std::_Sp_counted_base::_M_add_ref_copy();
   1.313 us [10471] |       } /* std::__shared_count::__shared_count */
   1.735 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
   2.177 us [10471] |   } /* std::shared_ptr::shared_ptr */
            [10471] |   std::shared_ptr::~shared_ptr() {
            [10471] |     std::__shared_ptr::~__shared_ptr() {
            [10471] |       std::__shared_count::~__shared_count() {
   0.518 us [10471] |         std::_Sp_counted_base::_M_release();
   1.104 us [10471] |       } /* std::__shared_count::~__shared_count */
   1.532 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
   2.029 us [10471] |   } /* std::shared_ptr::~shared_ptr */
            [10471] |   std::shared_ptr::~shared_ptr() {
            [10471] |     std::__shared_ptr::~__shared_ptr() {
            [10471] |       std::__shared_count::~__shared_count() {
            [10471] |         std::_Sp_counted_base::_M_release() {
   3.493 us [10471] |           operator delete(0x1209910);
   0.349 us [10471] |           operator delete(0x122d630);
   7.118 us [10471] |         } /* std::_Sp_counted_base::_M_release */
   7.524 us [10471] |       } /* std::__shared_count::~__shared_count */
   7.888 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
   8.250 us [10471] |   } /* std::shared_ptr::~shared_ptr */
  24.897 us [10471] | } /* main */
```

```
# DURATION     TID      FUNCTION
             [10471] | main() {
   2.335 us [10471] |   operator new(4) = 0x1209910;
             [10471] |   std::shared_ptr::shared_ptr() {
             [10471] |     std::__shared_ptr::__shared_ptr() {
             [10471] |       std::__shared_count::__shared_count() {
   2.860 us [10471] |         operator new(24) = 0x122d630;
   0.456 us [10471] |         std::_Sp_counted_ptr::_Sp_counted_ptr();
   4.907 us [10471] |       } /* std::__shared_count::__shared_count */
   0.163 us [10471] |       std::__enable_shared_from_this_helper();
   5.982 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
   6.450 us [10471] |   } /* std::shared_ptr::shared_ptr */
             [10471] |   std::shared_ptr::shared_ptr() {
             [10471] |     std::__shared_ptr::__shared_ptr() {
             [10471] |       std::__shared_count::__shared_count() {
   0.649 us [10471] |         std::_Sp_counted_base::_M_add_ref_copy();
   1.313 us [10471] |       } /* std::__shared_count::__shared_count */
   1.735 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
   2.177 us [10471] |   } /* std::shared_ptr::shared_ptr */
             [10471] |   std::shared_ptr::~shared_ptr() {
             [10471] |     std::__shared_ptr::~__shared_ptr() {
             [10471] |       std::__shared_count::~__shared_count() {
   0.518 us [10471] |         std::_Sp_counted_base::_M_release();
   1.104 us [10471] |       } /* std::__shared_count::~__shared_count */
   1.532 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
   2.029 us [10471] |   } /* std::shared_ptr::~shared_ptr */
             [10471] |   std::shared_ptr::~shared_ptr() {
             [10471] |     std::__shared_ptr::~__shared_ptr() {
             [10471] |       std::__shared_count::~__shared_count() {
             [10471] |         std::_Sp_counted_base::_M_release() {
   3.493 us [10471] |           operator delete(0x1209910);
   0.349 us [10471] |           operator delete(0x122d630);
   7.118 us [10471] |         } /* std::_Sp_counted_base::_M_release */
   7.524 us [10471] |       } /* std::__shared_count::~__shared_count */
   7.888 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
   8.250 us [10471] |   } /* std::shared_ptr::~shared_ptr */
  24.897 us [10471] | } /* main */
```

alloc!

```
# DURATION     TID     FUNCTION
            [10471] | main() {
   2.335 us [10471] |   operator new(4) = 0x1209910;
            [10471] |   std::shared_ptr::shared_ptr() {
            [10471] |     std::__shared_ptr::__shared_ptr() {
            [10471] |       std::__shared_count::__shared_count() {
   2.860 us [10471] |         operator new(24) = 0x122d630;
   0.456 us [10471] |         std::_Sp_counted_ptr::_Sp_counted_ptr();
   4.907 us [10471] |       } /* std::__shared_count::__shared_count */
   0.163 us [10471] |       std::__enable_shared_from_this_helper();
   5.982 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
   6.450 us [10471] |   } /* std::shared_ptr::shared_ptr */
            [10471] |   std::shared_ptr::shared_ptr() {
            [10471] |     std::__shared_ptr::__shared_ptr() {
            [10471] |       std::__shared_count::__shared_count() {
   0.649 us [10471] |         std::_Sp_counted_base::_M_add_ref_copy();
   1.313 us [10471] |       } /* std::__shared_count::__shared_count */
   1.735 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
   2.177 us [10471] |   } /* std::shared_ptr::shared_ptr */
            [10471] |   std::shared_ptr::~shared_ptr() {
            [10471] |     std::__shared_ptr::~__shared_ptr() {
            [10471] |       std::__shared_count::~__shared_count() {
   0.518 us [10471] |         std::_Sp_counted_base::_M_release();
   1.104 us [10471] |       } /* std::__shared_count::~__shared_count */
   1.532 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
   2.029 us [10471] |   } /* std::shared_ptr::~shared_ptr */
            [10471] |   std::shared_ptr::~shared_ptr() {
            [10471] |     std::__shared_ptr::~__shared_ptr() {
            [10471] |       std::__shared_count::~__shared_count() {
            [10471] |         std::_Sp_counted_base::_M_release() {
   3.493 us [10471] |           operator delete(0x1209910);
   0.349 us [10471] |           operator delete(0x122d630);
   7.118 us [10471] |         } /* std::_Sp_counted_base::_M_release */
   7.524 us [10471] |       } /* std::__shared_count::~__shared_count */
   7.888 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
   8.250 us [10471] |   } /* std::shared_ptr::~shared_ptr */
  24.897 us [10471] | } /* main */
```

++refcnt

```
# DURATION    TID     FUNCTION
            [10471] | main() {
   2.335 us [10471] |   operator new(4) = 0x1209910;
            [10471] |   std::shared_ptr::shared_ptr() {
            [10471] |     std::__shared_ptr::__shared_ptr() {
            [10471] |       std::__shared_count::__shared_count() {
   2.860 us [10471] |         operator new(24) = 0x122d630;
   0.456 us [10471] |         std::_Sp_counted_ptr::_Sp_counted_ptr();
   4.907 us [10471] |       } /* std::__shared_count::__shared_count */
   0.163 us [10471] |       std::__enable_shared_from_this_helper();
   5.982 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
   6.450 us [10471] |   } /* std::shared_ptr::shared_ptr */
            [10471] |   std::shared_ptr::shared_ptr() {
            [10471] |     std::__shared_ptr::__shared_ptr() {
            [10471] |       std::__shared_count::__shared_count() {
   0.649 us [10471] |         std::_Sp_counted_base::_M_add_ref_copy();
   1.313 us [10471] |       } /* std::__shared_count::__shared_count */
   1.735 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
   2.177 us [10471] |   } /* std::shared_ptr::shared_ptr */
            [10471] |   std::shared_ptr::~shared_ptr() {
            [10471] |     std::__shared_ptr::~__shared_ptr() {
            [10471] |       std::__shared_count::~__shared_count() {
   0.518 us [10471] |         std::_Sp_counted_base::_M_release();
   1.104 us [10471] |       } /* std::__shared_count::~__shared_count */
   1.532 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
   2.029 us [10471] |   } /* std::shared_ptr::~shared_ptr */
            [10471] |   std::shared_ptr::~shared_ptr() {
            [10471] |     std::__shared_ptr::~__shared_ptr() {
            [10471] |       std::__shared_count::~__shared_count() {
            [10471] |         std::_Sp_counted_base::_M_release() {
   3.493 us [10471] |           operator delete(0x1209910);
   0.349 us [10471] |           operator delete(0x122d630);
   7.118 us [10471] |         } /* std::_Sp_counted_base::_M_release */
   7.524 us [10471] |       } /* std::__shared_count::~__shared_count */
   7.888 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
   8.250 us [10471] |   } /* std::shared_ptr::~shared_ptr */
  24.897 us [10471] | } /* main */
```

--refcnt

```
# DURATION    TID      FUNCTION
           [10471] | main() {
  2.335 us [10471] |   operator new(4) = 0x1209910;
           [10471] |   std::shared_ptr::shared_ptr() {
           [10471] |     std::__shared_ptr::__shared_ptr() {
           [10471] |       std::__shared_count::__shared_count() {
  2.860 us [10471] |         operator new(24) = 0x122d630;
  0.456 us [10471] |         std::_Sp_counted_ptr::_Sp_counted_ptr();
  4.907 us [10471] |       } /* std::__shared_count::__shared_count */
  0.163 us [10471] |       std::__enable_shared_from_this_helper();
  5.982 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
  6.450 us [10471] |   } /* std::shared_ptr::shared_ptr */
           [10471] |   std::shared_ptr::shared_ptr() {
           [10471] |     std::__shared_ptr::__shared_ptr() {
           [10471] |       std::__shared_count::__shared_count() {
  0.649 us [10471] |         std::_Sp_counted_base::_M_add_ref_copy();
  1.313 us [10471] |       } /* std::__shared_count::__shared_count */
  1.735 us [10471] |     } /* std::__shared_ptr::__shared_ptr */
  2.177 us [10471] |   } /* std::shared_ptr::shared_ptr */
           [10471] |   std::shared_ptr::~shared_ptr() {
           [10471] |     std::__shared_ptr::~__shared_ptr() {
           [10471] |       std::__shared_count::~__shared_count() {
  0.518 us [10471] |         std::_Sp_counted_base::_M_release();
  1.104 us [10471] |       } /* std::__shared_count::~__shared_count */
  1.532 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
  2.029 us [10471] |   } /* std::shared_ptr::~shared_ptr */
           [10471] |   std::shared_ptr::~shared_ptr() {
           [10471] |     std::__shared_ptr::~__shared_ptr() {
           [10471] |       std::__shared_count::~__shared_count() {
           [10471] |         std::_Sp_counted_base::_M_release() {
  3.493 us [10471] |           operator delete(0x1209910);
  0.349 us [10471] |           operator delete(0x122d630);
  7.118 us [10471] |         } /* std::_Sp_counted_base::_M_release */
  7.524 us [10471] |       } /* std::__shared_count::~__shared_count */
  7.888 us [10471] |     } /* std::__shared_ptr::~__shared_ptr */
  8.250 us [10471] |   } /* std::shared_ptr::~shared_ptr */
 24.897 us [10471] | } /* main */
```

dealloc!

Analyzing Clang

```
$ uftrace -t 2ms -F cc1_main ./clang fibonacci.c
# DURATION    TID     FUNCTION
            [ 9045] | cc1_main() {
            [ 9045] |   clang::CompilerInvocation::CreateFromArgs() {
   2.270 ms [ 9045] |     ParseCodeGenArgs();
   8.653 ms [ 9045] |   } /* clang::CompilerInvocation::CreateFromArgs */
            [ 9045] |   clang::ExecuteCompilerInvocation() {
            [ 9045] |     clang::CompilerInstance::ExecuteAction() {
   2.185 ms [ 9045] |       clang::FrontendAction::BeginSourceFile();
            [ 9045] |       clang::FrontendAction::Execute() {
            [ 9045] |         clang::CodeGenAction::ExecuteAction() {
            [ 9045] |           clang::ASTFrontendAction::ExecuteAction() {
            [ 9045] |             clang::ParseAST() {
            [ 9045] |               clang::Parser::Initialize() {
   3.841 ms [ 9045] |                 clang::Preprocessor::Lex();
   3.887 ms [ 9045] |               } /* clang::Parser::Initialize */
            [ 9045] |               clang::BackendConsumer::HandleTranslationUnit() {
            [ 9045] |                 clang::EmitBackendOutput() {
            [ 9045] |                   llvm::LLVMTargetMachine::addPassesToEmitFile() {
   2.044 ms [ 9045] |                     addPassesToGenerateCode();
   2.068 ms [ 9045] |                   } /* llvm::LLVMTargetMachine::addPassesToEmitFile */
            [ 9045] |                   llvm::legacy::PassManager::run() {
   2.196 ms [ 9045] |                     llvm::legacy::PassManagerImpl::run();
   2.196 ms [ 9045] |                   } /* llvm::legacy::PassManager::run */
   5.053 ms [ 9045] |                 } /* clang::EmitBackendOutput */
   5.076 ms [ 9045] |               } /* clang::BackendConsumer::HandleTranslationUnit */
  23.361 ms [ 9045] |             } /* clang::ParseAST */
  23.385 ms [ 9045] |           } /* clang::ASTFrontendAction::ExecuteAction */
  23.385 ms [ 9045] |         } /* clang::CodeGenAction::ExecuteAction */
  23.386 ms [ 9045] |       } /* clang::FrontendAction::Execute */
  25.651 ms [ 9045] |     } /* clang::CompilerInstance::ExecuteAction */
  25.667 ms [ 9045] |   } /* clang::ExecuteCompilerInvocation */
  34.368 ms [ 9045] | } /* cc1_main */
```

Analyzing Clang

```
$ uftrace -t 2ms -F cc1_main ./clang fibonacci.c
# DURATION     TID     FUNCTION
            [ 9045] | cc1_main() {
            [ 9045] |   clang::CompilerInvocation::CreateFromArgs() {
  2.270 ms [ 9045] |     ParseCodeGenArgs();
  8.653 ms [ 9045] |   } /* clang::CompilerInvocation::CreateFromArgs */
            [ 9045] |   clang::ExecuteCompilerInvocation() {
            [ 9045] |     clang::CompilerInstance::ExecuteAction() {
  2.185 ms [ 9045] |       clang::FrontendAction::BeginSourceFile();
            [ 9045] |       clang::FrontendAction::Execute() {
            [ 9045] |         clang::CodeGenAction::ExecuteAction() {
            [ 9045] |           clang::ASTFrontendAction::ExecuteAction() {
            [ 9045] |             clang::ParseAST() {
            [ 9045] |               clang::Parser::Initialize() {
  3.841 ms [ 9045] |                 clang::Preprocessor::Lex();
  3.887 ms [ 9045] |               } /* clang::Parser::Initialize */
            [ 9045] |               clang::BackendConsumer::HandleTranslationUnit() {
            [ 9045] |                 clang::EmitBackendOutput() {
            [ 9045] |                   llvm::LLVMTargetMachine::addPassesToEmitFile() {
  2.044 ms [ 9045] |                     addPassesToGenerateCode();
  2.068 ms [ 9045] |                   } /* llvm::LLVMTargetMachine::addPassesToEmitFile */
            [ 9045] |                   llvm::legacy::PassManager::run() {
  2.196 ms [ 9045] |                     llvm::legacy::PassManagerImpl::run();
  2.196 ms [ 9045] |                   } /* llvm::legacy::PassManager::run */
  5.053 ms [ 9045] |                 } /* clang::EmitBackendOutput */
  5.076 ms [ 9045] |               } /* clang::BackendConsumer::HandleTranslationUnit */
 23.361 ms [ 9045] |             } /* clang::ParseAST */
 23.385 ms [ 9045] |           } /* clang::ASTFrontendAction::ExecuteAction */
 23.385 ms [ 9045] |         } /* clang::CodeGenAction::ExecuteAction */
 23.386 ms [ 9045] |       } /* clang::FrontendAction::Execute */
 25.651 ms [ 9045] |     } /* clang::CompilerInstance::ExecuteAction */
 25.667 ms [ 9045] |   } /* clang::ExecuteCompilerInvocation */
 34.368 ms [ 9045] | } /* cc1_main */
```

ParseAST

Analyzing Clang

```
$ uftrace -t 2ms -F cc1_main ./clang fibonacci.c
# DURATION    TID     FUNCTION
            [ 9045] | cc1_main() {
            [ 9045] |   clang::CompilerInvocation::CreateFromArgs() {
   2.270 ms [ 9045] |     ParseCodeGenArgs();
   8.653 ms [ 9045] |   } /* clang::CompilerInvocation::CreateFromArgs */
            [ 9045] |   clang::ExecuteCompilerInvocation() {
            [ 9045] |     clang::CompilerInstance::ExecuteAction() {
   2.185 ms [ 9045] |       clang::FrontendAction::BeginSourceFile();
            [ 9045] |       clang::FrontendAction::Execute() {
            [ 9045] |         clang::CodeGenAction::ExecuteAction() {
            [ 9045] |           clang::ASTFrontendAction::ExecuteAction() {
            [ 9045] |             clang::ParseAST() {
            [ 9045] |               clang::Parser::Initialize() {
   3.841 ms [ 9045] |                 clang::Preprocessor::Lex();
   3.887 ms [ 9045] |               } /* clang::Parser::Initialize */
            [ 9045] |               clang::BackendConsumer::HandleTranslationUnit() {
            [ 9045] |                 clang::EmitBackendOutput() {
            [ 9045] |                   llvm::LLVMTargetMachine::addPassesToEmitFile() {
   2.044 ms [ 9045] |                     addPassesToGenerateCode();
   2.068 ms [ 9045] |                   } /* llvm::LLVMTargetMachine::addPassesToEmitFile */
            [ 9045] |                   llvm::legacy::PassManager::run() {
   2.196 ms [ 9045] |                     llvm::legacy::PassManagerImpl::run();
   2.196 ms [ 9045] |                   } /* llvm::legacy::PassManager::run */
   5.053 ms [ 9045] |                 } /* clang::EmitBackendOutput */
   5.076 ms [ 9045] |               } /* clang::BackendConsumer::HandleTranslationUnit */
  23.361 ms [ 9045] |             } /* clang::ParseAST */
  23.385 ms [ 9045] |           } /* clang::ASTFrontendAction::ExecuteAction */
  23.385 ms [ 9045] |         } /* clang::CodeGenAction::ExecuteAction */
  23.386 ms [ 9045] |       } /* clang::FrontendAction::Execute */
  25.651 ms [ 9045] |     } /* clang::CompilerInstance::ExecuteAction */
  25.667 ms [ 9045] |   } /* clang::ExecuteCompilerInvocation */
  34.368 ms [ 9045] | } /* cc1_main */
```

Backend Code Gen

Analyzing Clang TMP expansion

```
$ clang++ tmpfib.cc
```

```
#include <iostream>

#define fibnum 8
template <unsigned N> struct Fibonacci {
   enum { value = Fibonacci<N-1>::value + Fibonacci<N-2>::value };
};
template <> struct Fibonacci<1> { enum { value = 1 }; };
template <> struct Fibonacci<0> { enum { value = 0 }; };

int main(void)
{
   std::cout << "Fibonacci(" << fibnum << ") = ";
   std::cout << Fibonacci<fibnum>::value;
   std::cout << std::endl;
}
```

Analyzing Clang TMP expansion

```
$ clang++ tmpfib.cc
```

```
#include <iostream>

#define fibnum 8
template <unsigned N> struct Fibonacci {
   enum { value = Fibonacci<N-1>::value + Fibonacci<N-2>::value };
};
template <> struct Fibonacci<1> { enum { value = 1 }; };
template <> struct Fibonacci<0> { enum { value = 0 }; };

int main(void)
{
   std::cout << "Fibonacci(" << fibnum << ") = ";
   std::cout << Fibonacci<fibnum>::value;
   std::cout << std::endl;
}
```

Analyzing Clang TMP expansion

```
$ clang++ tmpfib.cc
```

Recursive Expansion

```
#include <iostream>

#define fibnum 8
template <unsigned N> struct Fibonacci {
   enum { value = Fibonacci<N-1>::value + Fibonacci<N-2>::value };
};
template <> struct Fibonacci<1> { enum { value = 1 }; };
template <> struct Fibonacci<0> { enum { value = 0 }; };

int main(void)
{
   std::cout << "Fibonacci(" << fibnum << ") = ";
   std::cout << Fibonacci<fibnum>::value;
   std::cout << std::endl;
}
```

Analyzing Clang TMP expansion

```
$ clang++ tmpfib.cc
```

Recursive Expansion

```
#include <iostream>

#define fibnum 8
template <unsigned N> struct Fibonacci {
   enum { value = Fibonacci<N-1>::value + Fibonacci<N-2>::value };
};
template <> struct Fibonacci<1> { enum { value = 1 }; };
template <> struct Fibonacci<0> { enum { value = 0 }; };

int main(void)
{
   std::cout << "Fibonacci(" << fibnum << ") = ";
   std::cout << Fibonacci<fibnum>::value;
   std::cout << std::endl;
}
```

Analyzing Clang TMP expansion

```
$ uftrace record -t 1ms clang++ tmpfib.cc
```

Recursive Expansion

```
#include <iostream>

#define fibnum 8
template <unsigned N> struct Fibonacci {
   enum { value = Fibonacci<N-1>::value + Fibonacci<N-2>::value };
};
template <> struct Fibonacci<1> { enum { value = 1 }; };
template <> struct Fibonacci<0> { enum { value = 0 }; };

int main(void)
{
   std::cout << "Fibonacci(" << fibnum << ") = ";
   std::cout << Fibonacci<fibnum>::value;
   std::cout << std::endl;
}
```

DEMO

Clang / LLVM

(can only be opened in chrome browsers)

DEMO

V8 JavaScript Engine

`(can only be opened in chrome browsers)`

Thanks!

https://github.com/namhyung/uftrace

```
$ gcc test.c
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
<bar>:

  ret



<foo>:

  call <bar>
  ret



<main>:

  call <foo>
  ret
```

```
$ gcc -pg test.c
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
<bar>:
  call <mcount@plt>
  ret


<foo>:
  call <mcount@plt>
  call <bar>
  ret


<main>:
  call <mcount@plt>
  call <foo>
  ret
```

```
$ gcc -pg -fno-omit-frame-pointer test.c
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
<bar>:
  push %rbp
  mov  %rsp,%rbp
  call <mcount@plt>
  ret
<foo>:
  push %rbp
  mov  %rsp,%rbp
  call <mcount@plt>
  call <bar>
  ret
<main>:
  push %rbp
  mov  %rsp,%rbp
  call <mcount@plt>
  call <foo>
  ret
```

```
$ gcc test.c
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
<bar>:

  ret



<foo>:

  call <bar>
  ret



<main>:

  call <foo>
  ret
```

```
$ gcc -finstrument-functions test.c
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
<bar>:
  call <__cyg_profile_func_enter@plt>
  ret


<foo>:
  call <__cyg_profile_func_enter@plt>
  call <bar>
  ret


<main>:
  call <__cyg_profile_func_enter@plt>
  call <foo>
  ret
```

```
$ gcc -finstrument-functions test.c
```

```
void bar() {
}
void foo() {
  bar();
}
int main() {
  foo();
}
```

```
<bar>:
  call <__cyg_profile_func_enter@plt>
  call <__cyg_profile_func_exit@plt>
  ret

<foo>:
  call <__cyg_profile_func_enter@plt>
  call <bar>
  call <__cyg_profile_func_exit@plt>
  ret

<main>:
  call <__cyg_profile_func_enter@plt>
  call <foo>
  call <__cyg_profile_func_exit@plt>
  ret
```

```
$ gcc -pg test.c

$ uftrace -D 2 a.out
# DURATION    TID     FUNCTION
   0.648 us [32431] | __monstartup();
   0.480 us [32431] | __cxa_atexit();
            [32431] | main() {
   0.215 us [32431] |   foo();
   0.717 us [32431] | } /* main */
```

```
-D DEPTH, --depth=DEPTH
      Set global trace limit in nesting level.
```

```
$ gcc -pg test.c

$ uftrace -F foo a.out
# DURATION    TID     FUNCTION
            [32432] | foo() {
  0.175 us [32432] |   bar();
  1.137 us [32432] | } /* foo */
```

```
-F FUNC, --filter=FUNC
    Set filter to trace selected functions only.
```

```
$ gcc -pg test.c

$ uftrace -N foo a.out
# DURATION    TID     FUNCTION
   0.728 us [32436] | __monstartup();
   0.505 us [32436] | __cxa_atexit();
   0.741 us [32436] | main();
```

```
-N FUNC, --notrace=FUNC
    Set filter not to trace selected functions
    (and children)
```

```
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
```

ARGUMENTS

```
<argument>    :=  <symbol> "@" <specs>
<specs>       :=  <spec>  | <spec> "," <spec>
<spec>        :=  ( <int_spec> | <float_spec> | <ret_spec> )
<int_spec>    :=  "arg" N [ "/" <format> [ <size> ] ] [ "%" ( <reg> | <stack> ) ]
<float_spec>  :=  "fparg" N [ "/" ( <size> | "80" ) ] [ "%" ( <reg> | <stack> ) ]
<ret_spec>    :=  "retval" [ "/" <format> [ <size> ] ]
<format>      :=  "i" | "u" | "x" | "s" | "c" | "f"
<size>        :=  "8" | "16" | "32" | "64"
<reg>         :=  <arch-specific register name>  # "rdi", "xmm0", "r0", ...
<stack>       :=  "stack" [ "+" ] <offset>
```

```
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
```

ARGUMENTS

```
<argument>     :=  <symbol> "@" <specs>
<specs>        :=  <spec>  | <spec> "," <spec>
<spec>         :=  ( <int_spec> | <float_spec> | <ret_spec> )
<int_spec>     :=  "arg" N [ "/" <format> [ <size> ] ] [ "%" ( <reg> | <stack> ) ]
<float_spec>   :=  "fparg" N [ "/" ( <size> | "80" ) ] [ "%" ( <reg> | <stack> ) ]
<ret_spec>     :=  "retval" [ "/" <format> [ <size> ] ]
<format>       :=  "i" | "u" | "x" | "s" | "c" | "f"
<size>         :=  "8" | "16" | "32" | "64"
<reg>          :=  <arch-specific register name>  # "rdi", "xmm0", "r0", ...
<stack>        :=  "stack" [ "+" ] <offset>
```

```
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
```

ARGUMENTS

```
<argument>     :=  <symbol> "@" <specs>
<specs>        :=  <spec>  | <spec> "," <spec>
<spec>         :=  ( <int_spec> | <float_spec> | <ret_spec> )
<int_spec>     :=  "arg" N [ "/" <format> [ <size> ] ] [ "%" ( <reg> | <stack> ) ]
<float_spec>   :=  "fparg" N [ "/" ( <size> | "80" ) ] [ "%" ( <reg> | <stack> ) ]
<ret_spec>     :=  "retval" [ "/" <format> [ <size> ] ]
<format>       :=  "i" | "u" | "x" | "s" | "c" | "f"
<size>         :=  "8" | "16" | "32" | "64"
<reg>          :=  <arch-specific register name>  # "rdi", "xmm0", "r0", ...
<stack>        :=  "stack" [ "+" ] <offset>
```

```
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
```

ARGUMENTS

```
<argument>     :=  <symbol> "@" <specs>
<specs>        :=  <spec>  | <spec> "," <spec>
<spec>         :=  ( <int_spec> | <float_spec> | <ret_spec> )
<int_spec>     :=  "arg" N [ "/" <format> [ <size> ] ] [ "%" ( <reg> | <stack> ) ]
<float_spec>   :=  "fparg" N [ "/" ( <size> | "80" ) ] [ "%" ( <reg> | <stack> ) ]
<ret_spec>     :=  "retval" [ "/" <format> [ <size> ] ]
<format>       :=  "i" | "u" | "x" | "s" | "c" | "f"
<size>         :=  "8" | "16" | "32" | "64"
<reg>          :=  <arch-specific register name>  # "rdi", "xmm0", "r0", ...
<stack>        :=  "stack" [ "+" ] <offset>
```

Analyzing Kernel Functions

```
$ gcc -pg hello-cppcon.c
```

Analyzing Kernel Functions

```
$ gcc -pg hello-cppcon.c
$ sudo uftrace -k a.out
Hello CppCon!
```

Analyzing Kernel Functions

```
$ gcc -pg hello-cppcon.c
$ sudo uftrace -k a.out
Hello CppCon!
# DURATION    TID     FUNCTION
   0.395 us [ 8926] | __monstartup();
   0.354 us [ 8926] | __cxa_atexit();
            [ 8926] | main() {
            [ 8926] |   puts() {
   0.572 us [ 8926] |     sys_newfstat();
   1.316 us [ 8926] |     __do_page_fault();
   4.123 us [ 8926] |   } /* puts */
            [ 8926] |   fflush() {
   5.229 us [ 8926] |     sys_write();
   6.454 us [ 8926] |   } /* fflush */
  11.171 us [ 8926] | } /* main */
```