19

```
void run_rt_thread()
{
    std::mutex m;
    set_realtime_max_priority();

    get_file_size();
    do_malloc_free();
    do_vector_reserve();
    do_mutex_lock_unlock (m);

    set_realtime_min_priority();
}

int main()
{
    std::thread t1 ([&] { run_rt_thread();});
    t1.join();
```

```
std::uintmax_t get_file_size()
{
    std::filesystem::path awk_path ("/usr/bin/awk");
    std::uintmax_t file_size
        = std::filesystem::file_size (awk_path);

    return file_size;
}

void do_malloc_free()
{
    auto m = malloc (1024);
    free (m);
}

void do_vector_reserve()
{
    std::vector<int> v;
    v.reserve (42);
}

void do_mutex_lock_unlock (std::mutex& m)
{
    std::unique_lock l (m);
}
```


What


to


catch?

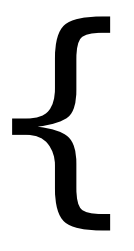

want


do


we

- System calls
- malloc/free
- lock/unlock

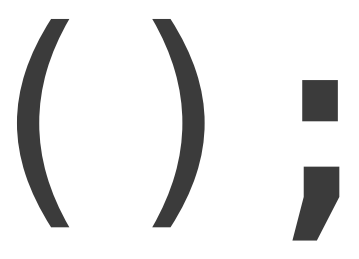
get_file_size



std::mutex

```
set_realtime_max_priority();
```


void



do_mutex_lock_unlock

```
do_vector_reserve();
```

```
set_realtime_min_priority();
```


int

run_rt_thread()


(m);

main()

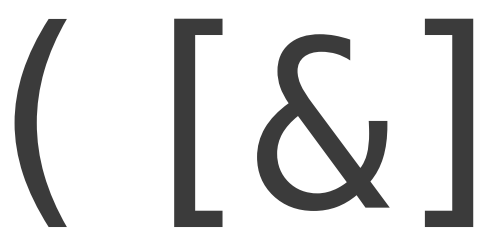
```
do_malloc_free();
```

```
run_rt_thread();});
```

std::thread

t1

```
t1.join();
```



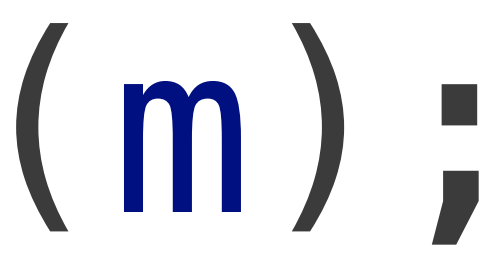
do_malloc_free

do_vector_reserve

do_mutex_lock_unlock

free

```
file_size;
```


(m) ;

std::filesystem::file_size

("/usr/bin/awk");

std::filesystem::path

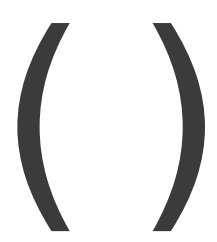
malloc

get_file_size()

return

file_size

(std::mutex&



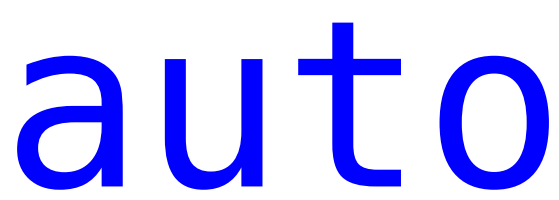
(awk_path);


void

```
(42);
```

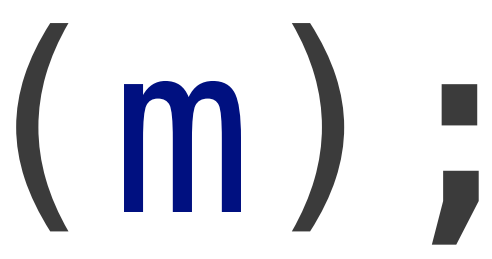
(1024);

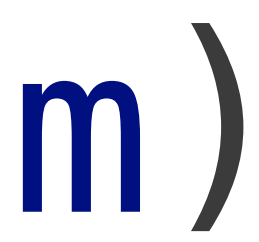
std::uintmax_t


auto

std::uintmax_t

awk_path


(m) ;



v.reserve

```
std::vector<int>
```

std::unique_lock

20

get_file_size

do_malloc_free

do_vector_reserve

do_mutex_lock_unlock