

4

3

```
const auto policy = SCHED_FIFO;  
pthread_t thread = pthread_self();
```

```
// Set the real-time scheduling policy and priority  
struct sched_param param;  
param.sched_priority = 47;
```

```
pthread_setschedparam(thread, policy, &param);
```

```
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();  
}
```



```
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);  
  
}
```

```
const auto policy = SCHED_FIFO;  
pthread_t thread = pthread_self();  
  
// Set the real-time scheduling policy and priority  
struct sched_param param;  
param.sched_priority = 47;  
  
pthread_setschedparam(thread, policy, &param);
```

```
sudo dtrace -c $BUILD_DIR/example -s rt_check.d
```

rt_check.d

```
pid$target::syscall:entry,  
  
pid$target::malloc:entry,  
pid$target::free:entry,  
  
pid$target::pthread_*:entry,  
  
pid$target::stat*:entry,  
pid$target::stat64*:entry,  
pid$target::fstat*:entry,  
pid$target::fstat64*:entry,  
pid$target::*write*:entry,  
pid$target::read*:entry  
  
/curthread->sched_pri>=47/  
{  
}
```