# IO profile和trace的插桩库


## 邹永浩

## 2019211168


### `LD_PRELOAD` 测试

劫持 `read` 和 `write` 函数

```
typedef ssize_t (*WRITE)(int fd, const void *buf, size_t count);

int write(int fd, const void *buf, size_t count)
{
    static void *handle = NULL;
    static WRITE old_write = NULL;

    if (!handle)
    {
        handle = dlopen("libc.so.6", RTLD_LAZY);
        old_write = (WRITE)dlsym(handle, "write");
    }
    printf("write func called. count = %ld\n", count);
    return old_write(fd, buf, count);
}

typedef ssize_t (*READ)(int fd, void *buf, size_t count);

int read(int fd, void *buf, size_t count)
{
    static void *handle = NULL;
    static WRITE old_write = NULL;

    if (!handle)
    {
        handle = dlopen("libc.so.6", RTLD_LAZY);
        old_write = (WRITE)dlsym(handle, "read");
    }
    printf("read func called. count = %ld\n", count);
    return old_write(fd, buf, count);
}
```

编译指令为：

```
gcc inject.c -shared -fPIC -DPIC -o inject.so -ldl
```

运行 `cat` 可以看到成功捕获函数:

```
zyh@WIN-HOME:~/MeasuringComputerPerformance/hw9$ LD_PRELOAD=$PWD/inject.so cat test
read func called. count = 131072
write func called. count = 5
test
read func called. count = 131072
```

## 添加 Profile 和 Trace相关代码

```
int write(int fd, const void *buf, size_t count)
{
    total_write_count++;
    total_write_size += count;

    // ...
    printf("write func called. count = %ld\n", count);
    struct timeval start;
    struct timeval end;
    unsigned long diff;
    gettimeofday(&start, NULL);
    int result = old_write(fd, buf, count);

    gettimeofday(&end, NULL);
    diff = 1000000 * (end.tv_sec - start.tv_sec) + end.tv_usec - start.tv_usec;
    printf("write time is %ld\n", diff);
    total_write_time += diff;
    return result;
}

void print_statistics()
{
    printf("total read is %ld\n", total_read_count);
    printf("total write is %ld\n", total_write_count);
    printf("average read size is %f\n", (double)total_read_size /
total_read_count);
    printf("average write size is %f\n", (double)total_write_size /
total_write_count);
    printf("average read time is %f\n", total_read_time / total_read_count);
    printf("average write time is %f\n", total_write_time / total_write_count);
}

typedef void (*EXIT)(int status) __attribute__((noreturn));

// 如果调用exit则输出统计结果
void exit(int status)
{
    printf("exit func called.\n");
    print_statistics();
    static void *handle = NULL;
    static EXIT old_exit = NULL;

    if (!handle)
    {
        handle = dlopen("libc.so.6", RTLD_LAZY);
        old_exit = (EXIT)dlsym(handle, "exit");
    }

    old_exit(status);
```

```
}

// 如果main函数结束则输出统计结果
__attribute__((destructor)) void main() {
    print_statistics();
}
```

## 测试`redis`

```
LD_PRELOAD=$PWD/inject.so ~/redis-5.0.8/src/redis-server
```

可以看到程序会一直输出`Trace`信息

```
zyh@WIN-HOME:~/MeasuringComputerPerformance/hw9$ LD_PRELOAD=$PWD/inject.so ~/redis-5.0.8/src/redis-server
12546:C 16 Apr 2020 21:55:05.852 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
12546:C 16 Apr 2020 21:55:05.852 # Redis version=5.0.8, bits=64, commit=00000000, modified=0, pid=12546, just started
12546:C 16 Apr 2020 21:55:05.852 # Warning: no config file specified, using the default config. In order to specify a config file use /home/zyh/redis-5.0.8/src/redis-server /path/to/redis.conf
12546:M 16 Apr 2020 21:55:05.852 # You requested maxclients of 10000 requiring at least 10032 max file descriptors.
12546:M 16 Apr 2020 21:55:05.852 # Server can't set maximum open files to 10032 because of OS error: Operation not permitted.
12546:M 16 Apr 2020 21:55:05.852 # Current maximum open files is 4096. maxclients has been reduced to 4064 to compensate for low ulimit. If you need higher maxclients increase 'ulimit -n'.
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 5.0.8 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 12546
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

12546:M 16 Apr 2020 21:55:05.853 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
12546:M 16 Apr 2020 21:55:05.853 # Server initialized
12546:M 16 Apr 2020 21:55:05.853 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then
reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
12546:M 16 Apr 2020 21:55:05.853 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the comma
nd 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled.
12546:M 16 Apr 2020 21:55:05.853 * DB loaded from disk: 0.000 seconds
12546:M 16 Apr 2020 21:55:05.853 * Ready to accept connections
read func called. count = 1
write time is 1
read func called. count = 4096
write time is 14
read func called. count = 1
write time is 1
read func called. count = 4096
write time is 8
read func called. count = 1
write time is 27
read func called. count = 4096
write time is 19
read func called. count = 1
write time is 2
read func called. count = 4096
```

程序结束时输出统计信息如下：

```
write time is 1
read func called. count = 1
write time is 2
read func called. count = 4096
write time is 8
12546:M 16 Apr 2020 21:55:08.060 # User requested shutdown...
12546:M 16 Apr 2020 21:55:08.060 * Saving the final RDB snapshot before exiting.
12546:M 16 Apr 2020 21:55:08.066 * DB saved on disk
12546:M 16 Apr 2020 21:55:08.066 # Redis is now ready to exit, bye bye...
exit func called.
total read is 47
total write is 6
average read size is 2004.936170
average write size is 12.166667
total read time is 6.127660
total write time is 1.500000
```

## 参考文献


https://www.jianshu.com/p/f78b16bd8905

https://www.cnblogs.com/LittleHann/p/3854977.html#_lab2_2_0

https://www.tutorialspoint.com/unix_system_calls/read.htm