



阅码场™

www.yomocode.com

多线程编程(3)

阅码场™

www.yomocode.com

麦当劳喜欢您来，喜欢您再来



扫描关注
Linux阅码场



多线程编程(3)

线程安全和可重入

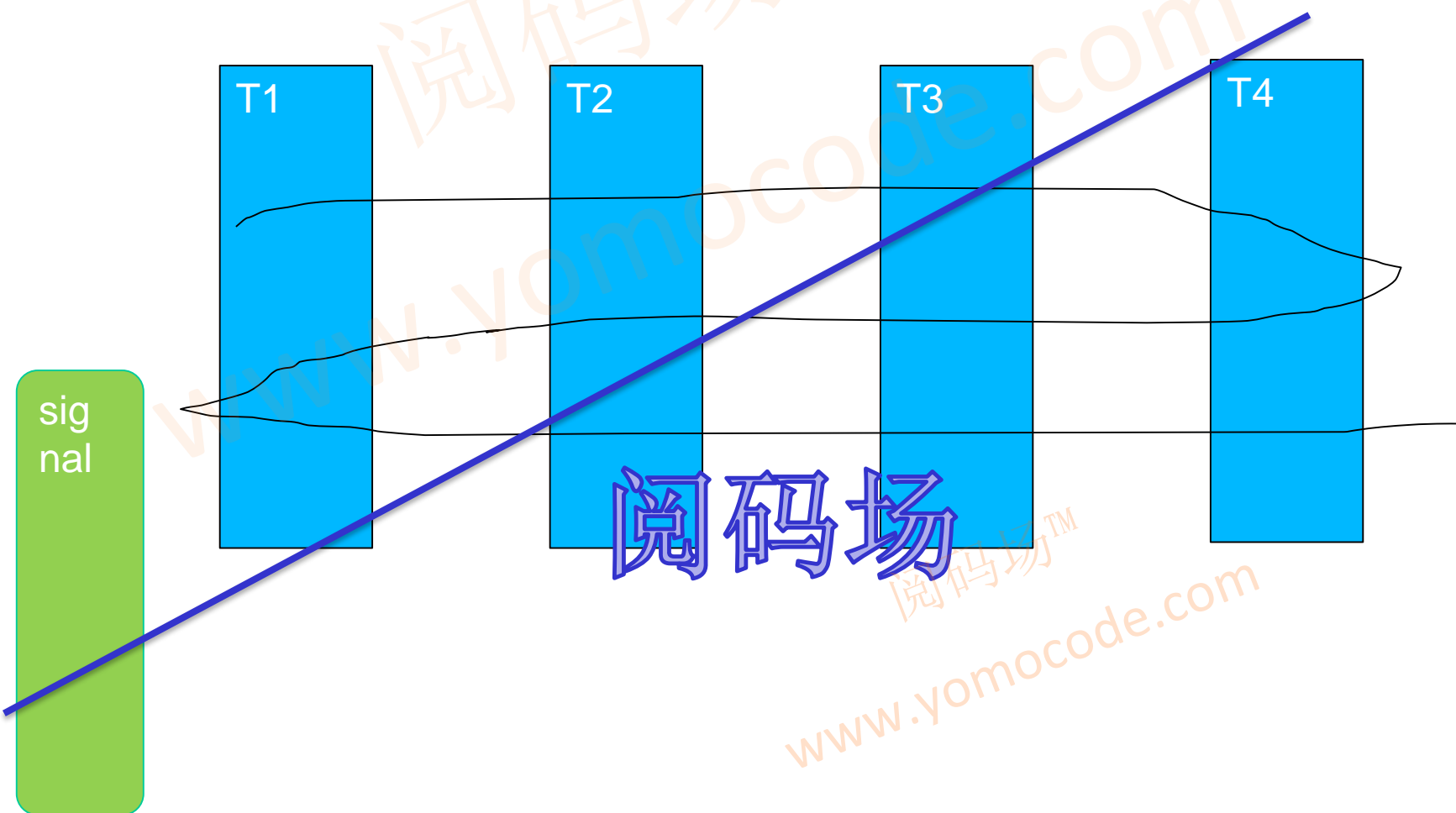
3.1 线程安全和可重入的重要性

3.2 什么时候要求线程安全和可重入?

3.3 典型的不安全的Linux库函数

3.4 多线程情况下的信号处理

典型场景



线程安全和异步(信号)安全

阅码场

“可重入”
通常强调这
一点

有的文
献说的
可重入
同时强
调2点

信号安全

不访问全局资源的
函数；访问全局资
源但是保存和恢复
的函数。

不能用mutex等。

线程安全

malloc

free

printf

不访问全局资源的
函数；
访问全局资源但是
加mutex的函数

线程不安全的通常不可重入；
可重入的通常线程安全
(极少量例外)

参考：

<https://deadbeef.me/2017/09/reentrant-threadsafe>

```
int t;

void swap(int *x, int *y) {
    int s;
    // save global variable
    s = t;
    t = *x;
    *x = *y;

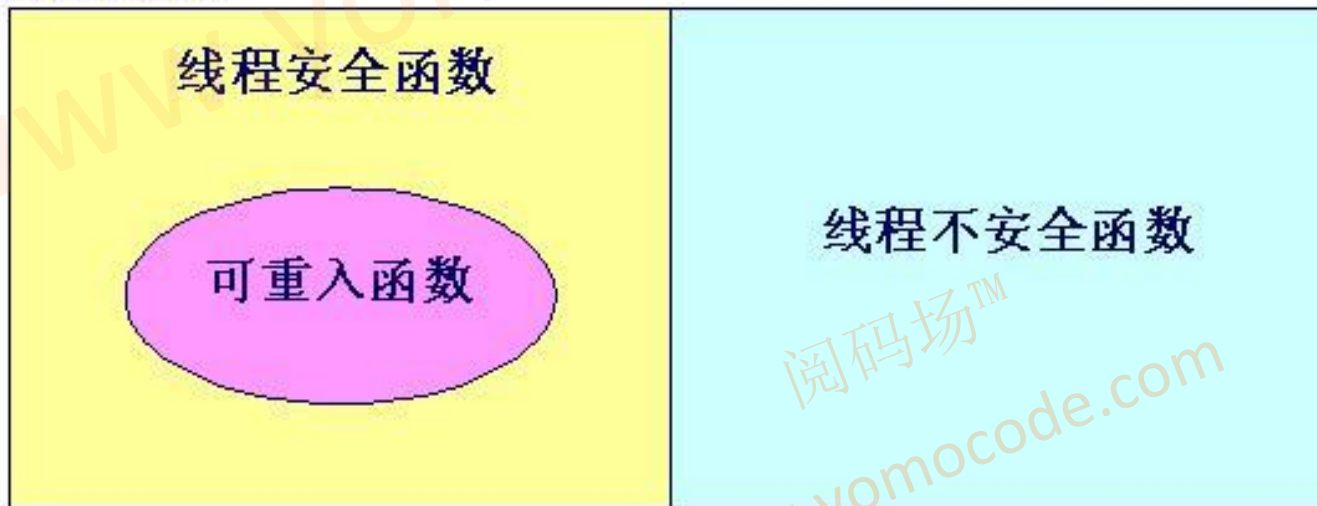
    // `my_func()` could be called here
    *y = t;
    // restore global variable
    t = s;
}

void my_func(){
    int x = 1, y = 2;
    swap(&x, &y);
}
```

不学究地理解多线程状态下的“可重入”

- 可重入函数满足两条件：一，函数是线程安全的；二，函数是可中断（对于Linux而言，异步的信号）的，执行了中断处理例程后，再回过头来继续执行函数，结果仍然正确

所有的函数



“非学究可重入”与线程安全注意事项

- ❖ 如果一个函数中用到了全局或静态变量(const除外), 那么它不是线程安全的, 也不是可重入的;
- ❖ 如果我们对它加以改进, 在访问全局或静态变量时使用互斥量或信号量等方式加锁, 则可以使它变成线程安全的, 但此时它仍然是不可重入的, 因为通常加锁方式是针对不同线程的访问, 而对同一线程可能出现问题;
- ❖ 如果将函数中的全局或静态变量去掉, 改成函数参数等其他形式, 则有可能使函数变成既线程安全, 又可重入。

异步信号同步化

阅码场



```
signal_handler()
```

```
{
```

```
    printf ->
```

```
    write
```

```
}
```

担心这一组函数

<u>asctime()</u>	<u>ecvt()</u>	<u>gethostent()</u>	<u>getutxline()</u>	<u>putc_unlocked()</u>
<u>basename()</u>	<u>encrypt()</u>	<u>getlogin()</u>	<u>gmtime()</u>	<u>putchar_unlocked()</u>
<u>catgets()</u>	<u>endgrent()</u>	<u>getnetbyaddr()</u>	<u>hcreate()</u>	<u>putenv()</u>
<u>crypt()</u>	<u>endpwent()</u>	<u>getnetbyname()</u>	<u>hdestroy()</u>	<u>pututxline()</u>
<u>ctime()</u>	<u>endutxent()</u>	<u>getnetent()</u>	<u>hsearch()</u>	<u>rand()</u>
<u>dbm_clearerr()</u>	<u>fcvt()</u>	<u>getopt()</u>	<u>inet_ntoa()</u>	<u>readdir()</u>
<u>dbm_close()</u>	<u>ftw()</u>	<u>getprotobyname()</u>	<u>l64a()</u>	<u>setenv()</u>
<u>dbm_delete()</u>	<u>gcvrt()</u>	<u>getprotobyname()</u>	<u>lgamma()</u>	<u>setgrent()</u>
<u>dbm_error()</u>	<u>getc_unlocked()</u>	<u>getprotoent()</u>	<u>lgammaf()</u>	<u>setkey()</u>
<u>dbm_fetch()</u>	<u>getchar_unlocked()</u>	<u>getpwent()</u>	<u>lgammal()</u>	<u>setpwent()</u>
<u>dbm_firstkey()</u>	<u>getdate()</u>	<u>getpwnam()</u>	<u>localeconv()</u>	<u>setutxent()</u>
<u>dbm_nextkey()</u>	<u>getenv()</u>	<u>getpwuid()</u>	<u>localtime()</u>	<u>strerror()</u>
<u>dbm_open()</u>	<u>getgrent()</u>	<u>getservbyname()</u>	<u>lrand48()</u>	<u>strtok()</u>
<u>dbm_store()</u>	<u>getgrgid()</u>	<u>getservbyport()</u>	<u>mrnd48()</u>	<u>ttyname()</u>
<u>dimame()</u>	<u>getgrnam()</u>	<u>getservent()</u>	<u>nftw()</u>	<u>unsetenv()</u>
<u>dllerror()</u>	<u>gethostbyaddr()</u>	<u>getutxent()</u>	<u>nl_langinfo()</u>	<u>wcstombs()</u>
<u>drand48()</u>	<u>gethostbyname()</u>	<u>getutxid()</u>	<u>ptsname()</u>	<u>wctomb()</u>

谢谢!

阅码场™

www.yomocode.com

阅码场™

www.yomocode.com



阅码场出品