
阅码场
YOMOCODE

多线程编程(4)

麦当劳喜欢您来，喜欢您再来

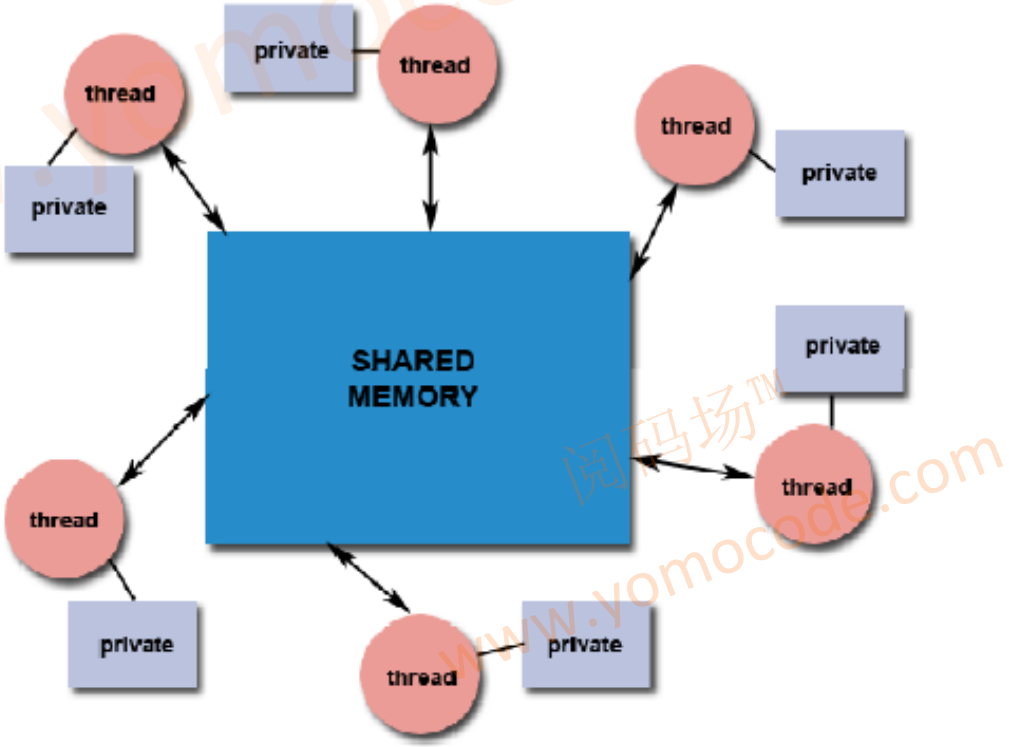
扫描关注
Linux阅码场

多线程编程(4)

多线程与栈
4.1 线程的栈
4.2 guard page

线程栈的双重特性

栈：线程独有，保存其运行状态和局部自动变量的。栈在线程开始的时候初始化，每个线程的栈互相独立（但内存属于整个进程），因此，栈是thread safe的。



栈的guard page

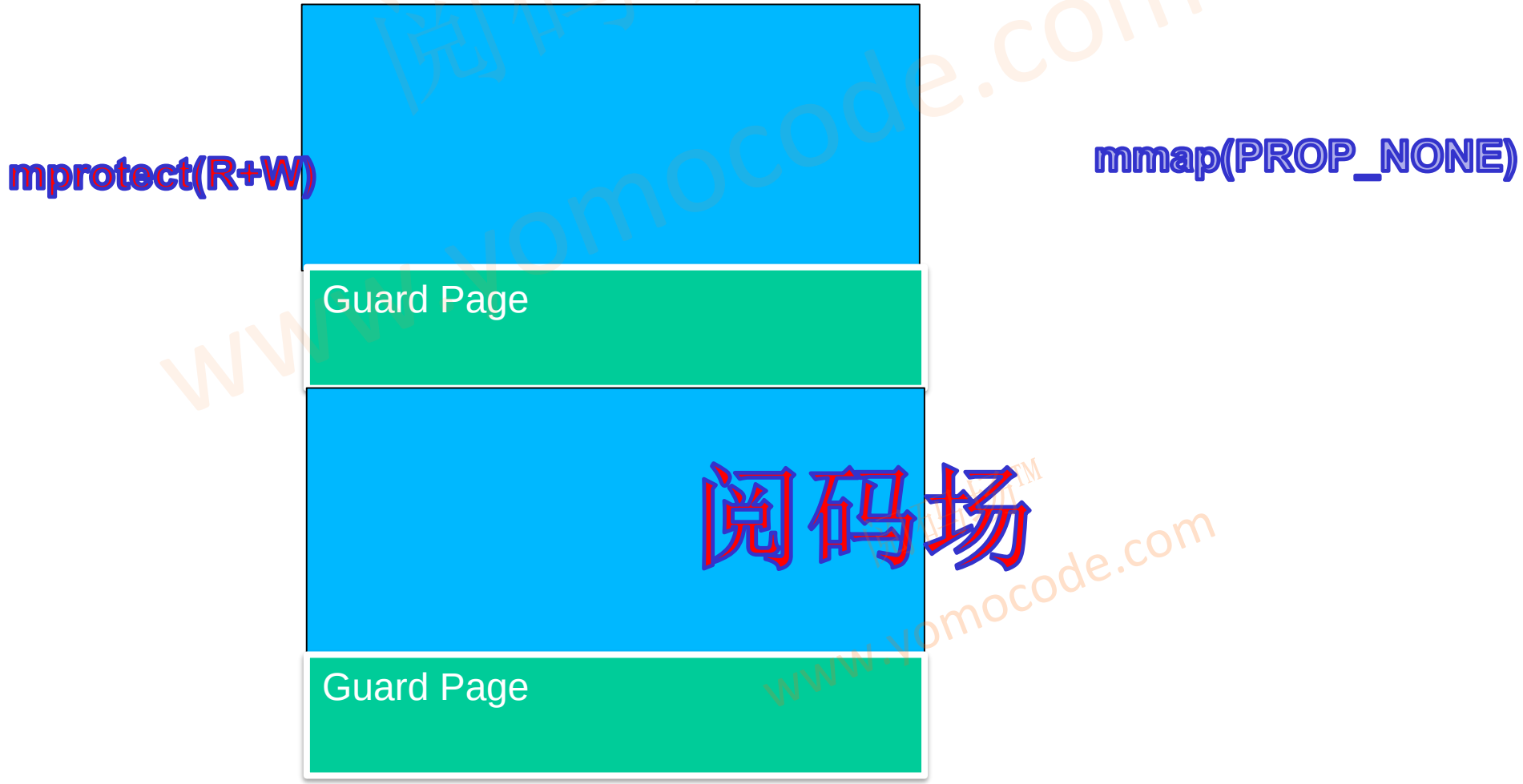
POSIX库API

```
#include <pthread.h>

    int pthread_attr_setguardsize(pthread_attr_t *attr, size_t guardsize);
    int pthread_attr_getguardsize(const pthread_attr_t *attr, size_t
*guardsize);
```

有意思的strace 的结果

```
mmap(NULL, 20480, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fbfcd14c000
mprotect(0x7fbfcd14d000, 16384, PROT_READ|PROT_WRITE) = 0


mmap(NULL, 20480, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fbfcd147000
mprotect(0x7fbfcd148000, 16384, PROT_READ|PROT_WRITE) = 0
```

pthread_create创建的线程内存情况



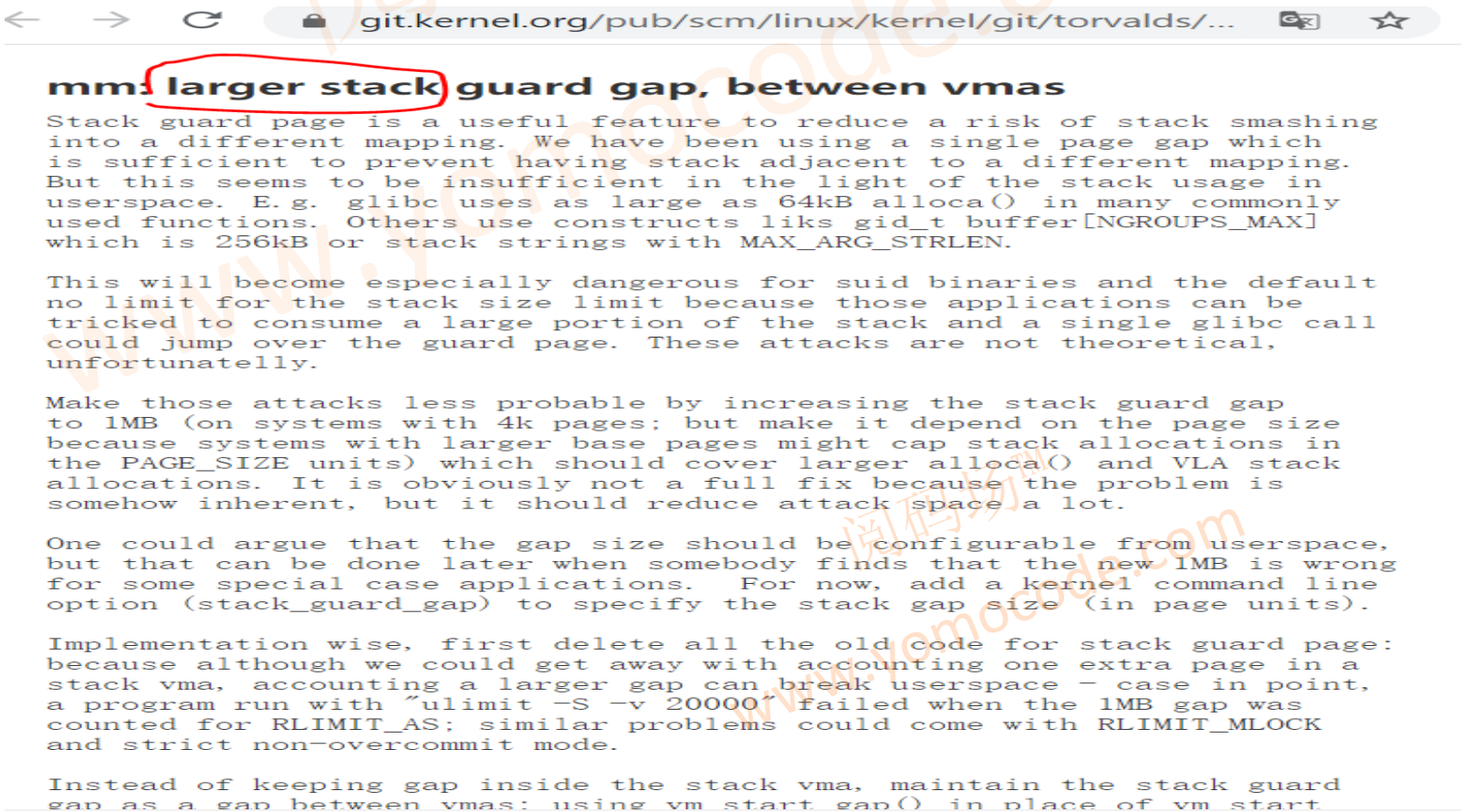
内核提供的针对GROWSDOWN栈的guard page

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=320b2b8de12698082609ebbc1a17165727f4c893

```
 /*
+ * This is like a special single-page "expand_downwards()",
+ * except we must first make sure that 'address-PAGE_SIZE'
+ * doesn't hit another vma.
+ *
+ * The "find_vma()" will do the right thing even if we wrap
+ */
+static inline int check_stack_guard_page(struct vm_area_struct *vma, unsigned long address)
+{
+       address &= PAGE_MASK;
+       if ((vma->vm_flags & VM_GROWSDOWN) && address == vma->vm_start) {
+               address -= PAGE_SIZE;
+               if (find_vma(vma->vm_mm, address) != vma)
+                       return -ENOMEM;
+
+               expand_stack(vma, address);
+       }
+       return 0;
+}
+
+/*
  * We enter with non-exclusive mmap_sem (to exclude vma changes,
  * but allow concurrent faults), and pte mapped but not yet locked.
  * We return with mmap_sem still held, but pte unmapped and unlocked.
@@ -2772,6 +2792,9 @@ static int do_anonymous_page(struct mm_struct *mm, struct vm_area_struct *vma,
        spinlock_t *ptl;
        pte_t entry;

+       if (check_stack_guard_page(vma, address) < 0)
+               return VM_FAULT_SIGBUS;
+
        if (!(flags & FAULT_FLAG_WRITE)) {
                entry = pte_mkspecial(pfn_pte(my_zero_pfn(address),
                                              vma->vm_page_prot));
```

内核的一些改进

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=1be7107fbe18eed3e319a6c3e83c78254b693acb



谢谢!

阅码场出品