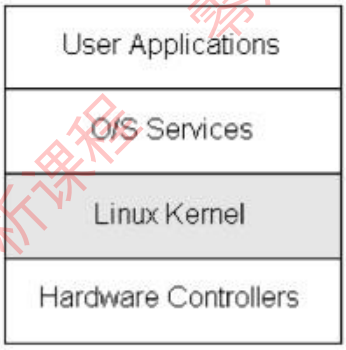
第 0008 讲 内存映射原理机制



一、Linux 内存映射原理机制

创建内存映射时，在进程的用户虚拟地址空间中分配一个虚拟内存区域。内核采用延迟分配物理内存的策略，在进程第一次访问虚拟页的时候，产生缺页异常。如果是文件映射，那么分配物理页，把文件指定区间的数据读到物理页中，然后在页表中把虚拟页映射到物理页。如果是匿名映射，就分配物理页，然后在页表中把虚拟页映射到物理页。

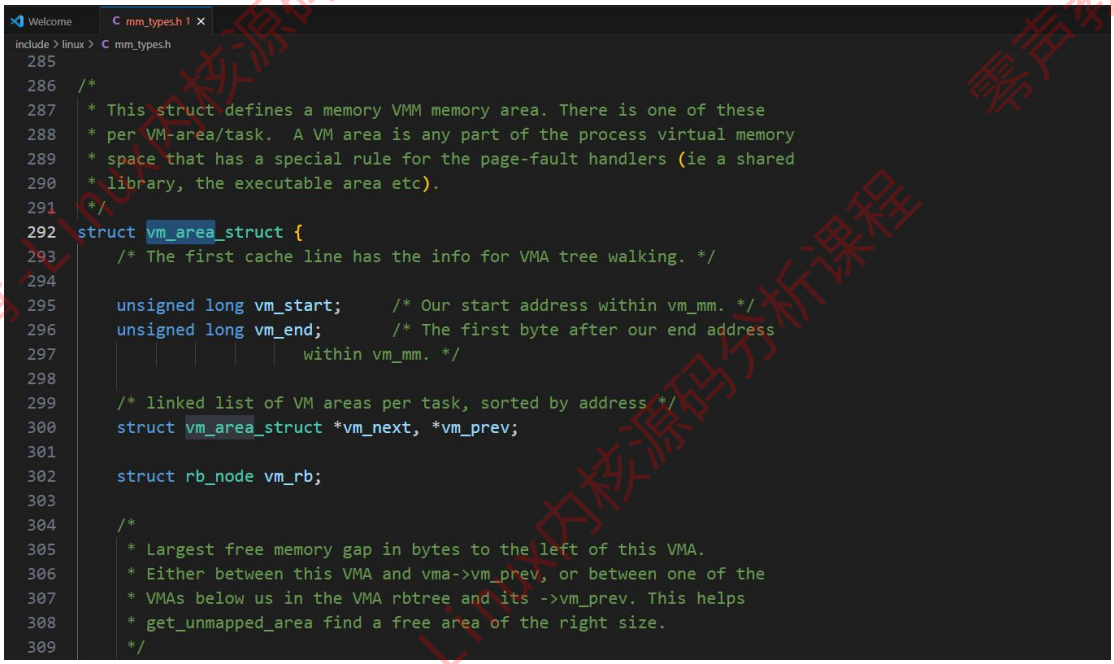
Linux 内核中的内存映射是指将一个文件或设备的内容映射到进程的虚拟地址空间，使得进程可以像访问内存一样访问这些内容。通过内存映射，进程可以避免频繁的磁盘读写操作，提高文件和设备的访问效率。分为两种：

（1）文件映射：文件支持的内存映射，把文件的一个区间映射到进程的虚拟地址空间，数据源是存储设备上的文件。

（2）匿名映射：没有文件支持的内存映射，把物理内存映射到进程的虚拟地址空间，没有数据源。

二、虚拟内存区域数据结构（vm_area_struct）

vm_area_struct 是用来描述一个进程的虚拟内存区域的数据结构，在 Linux 系统中，每个进程都有一个虚拟地址空间，由多个不同的 vm_area_struct 组成。主要核心成员如下：



三、内存映射（系统调用实战操作）

1、mmap()

系统调用是 Linux 中的一种内存映射文件的方式，它可以将一个文件或者设备映射到进程的虚拟地址空间中，从而实现对该文件或设备的访问。mmap()函数的原型如下：

```
void *mmap(void *addr, size_t length, int prot, int flags, int fd, off_t offset);
```

2、munmap()

在 Linux 中，munmap()系统调用用于释放由 mmap()函数创建的内存映射区域。munmap()函数的原型如下：

```
int munmap(void *addr, size_t length);
```

3、mprotect()

在 Linux 中，mprotect()系统调用用于修改内存区域的保护属性。mprotect()函数的原型如下：

```
int mprotect(void *addr, size_t len, int prot);
```