

# 实验四

---

## 实验四

### 题目4-1：内存管理——虚拟空间分布

#### 实验目的

#### 实验内容

#### 实验设计原理

#### 实验步骤

#### 实验结果及分析

#### 程序代码

### 题目4-2：内存管理——内存监控检查回收

#### 实验目的

#### 实验内容

#### 实验设计原理

#### 实验步骤

#### 实验结果及分析

#### 程序代码

## 题目4-1：内存管理——虚拟空间分布

---

### 实验目的

---

了解Linux进程虚拟地址空间的组成。

### 实验内容

---

编写C语言程序，分配位于堆和栈上的变量，观察变量的地址分布和变化。

### 实验设计原理

---

- 进程空间主要由以下几个分布组成：
  - 程序段(Text):程序代码在内存中的映射，存放函数体的二进制代码。
  - 初始化过的数据(Data):在程序运行初已经对变量进行初始化的数据。
  - 未初始化过的数据(BSS):在程序运行初未对变量进行初始化的数据。
  - 栈 (Stack):存储局部、临时变量，函数调用时，存储函数的返回指针，用于控制函数的调用和返回。在程序块开始时自动分配内存,结束时自动释放内存，其操作方式类似于数据结构中的栈。
  - 堆 (Heap):存储动态内存分配,需要程序员手工分配,手工释放.注意它与数据结构中的堆是两回事，分配方式类似于链表。
- Linux的虚拟地址空间范围为0 ~ 4G，Linux内核将这4G字节的空间分为两部分，将最高的1G字节（从虚拟地址0xC0000000到0xFFFFFFFF）供内核使用，称为“内核空间”。而将较低的3G字节（从虚拟地址0x00000000到0xBFFFFFFF）供各个进程使用，称为“用户空间”。因为每个进程可以通过系统调用进入内核，因此，Linux内核由系统内的所有进程共享。于是，从具体进程的角度来看，每个进程可以拥有4G字节的虚拟空间。

- Linux使用两级保护机制：0级供内核使用，3级供用户程序使用，每个进程有各自的私有用户空间（0~3G），这个空间对系统中的其他进程是不可见的，最高的1GB字节虚拟内核空间则为所有进程以及内核所共享。内核空间中存放的是内核代码和数据，而进程的用户空间中存放的是用户程序的代码和数据。不管是内核空间还是用户空间，它们都处于虚拟空间中。虽然内核空间占据了每个虚拟空间中的最高1GB字节，但映射到物理内存却总是从最低地址（0x00000000）。另外，使用虚拟地址可以很好的保护内核空间被用户空间破坏，虚拟地址到物理地址转换过程由操作系统和CPU共同完成(操作系统为CPU设置好页表，CPU通过MMU单元进行地址转换)。
- Linux进程标准的内存段布局除了用户不可见的内核虚拟存储器外，用户空间同样分为栈、堆、内存映射段、BSS和数据段以及Text。

## 实验步骤

---

编码定义字符指针s1、s2、s5均指向同一字符串“abcde”，定义字符串数组s3以及指向int型的指针数组s4，定义整型a=5、b=6。输出一遍所有变量的地址后通过子函数将一串d作为参数压入栈并再次定义一遍同名变量并输出各个地址位置。最终输出主函数与子函数的地址。

## 实验结果及分析

---



```

long int *s4[100];
char *s5 = "abcde";
int a = 5;
int b =6;//a,b在栈上, &a>&b地址反向增长

printf("variables address in main function: s1=%p s2=%p s3=%p s4=%p s5=%p
a=%p b=%p \n",
    s1,s2,s3,s4,s5,&a,&b);
printf("variables address in processcall:\n");
print("dddddddd",5);//参数入栈从右至左进行,p先进栈,str后进 &p>&str
printf("main=%p print=%p \n",main,print);
//打印代码段中主函数和子函数的地址,编译时先编译的地址低,后编译的地址高main<print
}

void print(char *str,int p)
{
    char *s1 = "abcde"; //abcde在常量区,s1在栈上
    char *s2 = "abcde"; //abcde在常量区,s2在栈上 s2-s1=6可能等于0,编译器优化了相同的
    常量,只在内存保存一份
    //而&s1>&s2
    char s3[] = "abcdeeee";//abcdeeee在常量区,s3在栈上,数组保存的内容为abcdeeee的一份拷贝

    long int *s4[100];
    char *s5 = "abcde";
    int a = 5;
    int b =6;
    int c;
    int d; //a,b,c,d均在栈上, &a>&b>&c>&d地址反向增长
    char *q=str;
    int m=p;
    char *r=(char *)malloc(1);
    char *w=(char *)malloc(1) ; // r<w 堆正向增长

    printf("s1=%p s2=%p s3=%p s4=%p s5=%p a=%p b=%p c=%p d=%p str=%p q=%p p=%p
m=%p r=%p w=%p \n",
        s1,s2,s3,s4,s5,&a,&b,&c,&d,&str,q,&p,&m,r,w);
    /* 栈和堆是在程序运行时候动态分配的,局部变量均在栈上分配。
    栈是反向增长的,地址递减;malloc等分配的内存空间在堆空间。堆是正向增长的,地址递增。
    r,w变量在栈上(则&r>&w), r,w所指内容在堆中(即r<w)。*/
}

```

## 题目4-2：内存管理——内存监控检查回收

### 实验目的

掌握查看实时监控内存、内存回收的方法;  
进一步掌握虚拟存储器的实现方法并理解虚拟内存、磁盘缓存的概念;  
掌握基本的内存管理知识。

### 实验内容

1. 用free命令监控内存使用情况,用vmstat命令监视虚拟内存使用情况。
2. 用命令ps检查和回收内容,用kill命令回收泄漏的内存。

# 实验设计原理

- 在提示符后输入命令free：  
Mem行显示物理内存——  
total列显示共有的可用内存(不显示核心使用的物理内存,通常大约1MB),  
used列显示被使用的内存总额,  
free列显示全部空闲的内存,  
shared列显示多个进程共享的内存总额,  
buffers列显示磁盘缓存的当前大小;  
Swap显示交换空间的信息,与上一行类似——如果该行为全0则没有使用交换空间。  
默认状态下,free命令以千字节(即1024字节为单位)显示内存使用情况。若使用-h参数,则以字节为单位显示内存使用情况;若使用-m参数,则以兆字节为单位显示内存使用情况。若命令带-s参数,则不间断地监视内存使用情况,如#free -b -s5,则表示该命令在终端窗口中连续不断地报告内存的使用情况,每5s更新一次。
- vmstat命令是一个通用监控程序,是VirtualMemoryStatistics(虚拟内存统计)的缩写。若vmstat命令没有带任何命令行参数,将得到一次性的报告。vmstat命令报告主要的活动类型有进程(procs)、内存(以千字节为单位)、交换分区(以千字节为单位)、来自块设备(硬盘驱动器)的输入输出量、系统中断(每秒发生的次数),以及中央处理单元(CPU)分配给用户、系统和空闲时分别占用的比例。
- 可以用ps、kill两个命令检测内存使用情况和进行回收。使用超级用户权限时,用命令ps可列出所有正在运行的程序名称和对应的进程号(PID)。kill命令的工作原理是向Linux操作系统的内核送出一个系统操作信号和程序的进程号(PID)。

## 实验步骤

### 1.用free命令监控内存使用情况

```
# free
# free -b -s5
```

用vmstat命令监视虚拟内存使用情况。

```
# vmstat
```

### 2.检查和回收内容

用命令ps列出所有正在运行的程序名称、对应的进程号(PID)等信息。

```
#ps v
```

手动执行一次不回收的Ping指令,造成内存泄漏的现象。  
再次使用ps查看Ping指令对应的PID,用kill命令回收泄漏的内存。

```
# kill -9 <PID>
```

## 实验结果及分析

实验五 多核多线程编程

剩余体验时间: 02:55:02

结束体验

实验手册云产品资源实验报告

体验云账号, 创建资源后生成

收起

子用户名: u-zngpw5hh@1915471854401972

子用户密码: Th5Ry5Kj7X9fa1R

AK ID: LTAI5IQV71KBAHPLKbY9KoNy

AK Secret: MPlaruG6rDnE7IGiY2u5wMlPUhw...

注意: 若登录子账号, 需打开控制台进行登录。

一键复制子账号登录链接

ECS服务器

磁盘ID: d-uf6533bjifujak3z53r

ECS公网地址: 106.14.10.5

ECS登录名: root

登录密码: lq7Sx1Ma9C

ECS实例ID: i-uf6gwz144knq4rpuclj

IP白名单: 0.0.0.0/0

地域: 华东 2 (上海)

2. root@iZuf6gwz144knq4rpuclj: ~

Welcome to Alibaba Cloud Elastic Compute Service !

[root@iZuf6gwz144knq4rpuclj ~]# free

total used free shared buff/cache available

Mem: 8008576 204408 7485824 496 318344 7566532

Swap: 0 0 0

[root@iZuf6gwz144knq4rpuclj ~]# free -b -s5

-bash: free -b -s5: command not found

[root@iZuf6gwz144knq4rpuclj ~]# free -b -s5

total used free shared buff/cache available

Mem: 8200781824 207872000 7665061888 475136 327847936 7748722688

Swap: 0 0 0

total used free shared buff/cache available

Mem: 8200781824 207880192 7665070080 475136 327831552 7748726784

Swap: 0 0 0

total used free shared buff/cache available

Mem: 8200781824 207872000 7665070080 475136 327839744 7748734976

Swap: 0 0 0

total used free shared buff/cache available

Mem: 8200781824 207937536 7664992256 475136 327852032 7748661248

Swap: 0 0 0

total used free shared buff/cache available

Mem: 8200781824 209874944 7663054848 475136 327852032 7746732032

Swap: 0 0 0

total used free shared buff/cache available

Mem: 8200781824 209977344 7662960640 475136 327843840 7746637824

Swap: 0 0 0

total used free shared buff/cache available

Mem: 8200781824 209973248 7662960640 475136 327847936 7746641920

Swap: 0 0 0

default

命令终端 已连接 华东2(上海) i-uf6gwz144knq4rpuclj 106.14.10.522 vx1petouhq 14pt 39/1 39 Rows 132 Cols en\_US.UTF-8

实验五 多核多线程编程

剩余体验时间: 02:54:12

结束体验

实验手册云产品资源实验报告

体验云账号, 创建资源后生成

收起

子用户名: u-zngpw5hh@1915471854401972

子用户密码: Th5Ry5Kj7X9fa1R

AK ID: LTAI5IQV71KBAHPLKbY9KoNy

AK Secret: MPlaruG6rDnE7IGiY2u5wMlPUhw...

注意: 若登录子账号, 需打开控制台进行登录。

一键复制子账号登录链接

ECS服务器

磁盘ID: d-uf6533bjifujak3z53r

ECS公网地址: 106.14.10.5

ECS登录名: root

登录密码: lq7Sx1Ma9C

ECS实例ID: i-uf6gwz144knq4rpuclj

IP白名单: 0.0.0.0/0

地域: 华东 2 (上海)

2. root@iZuf6gwz144knq4rpuclj: ~

Mem: 8200781824 209973248 7662960640 475136 327847936 7746641920

Swap: 0 0 0

Mem: 8200781824 209965056 7662960640 475136 327856128 7746646016

Swap: 0 0 0

Mem: 8200781824 209965056 7662960640 475136 327856128 7746650112

Swap: 0 0 0

Mem: 8200781824 209965056 7662960640 475136 327856128 7746650112

Swap: 0 0 0

Mem: 8200781824 209956864 7662960640 475136 327864320 7746650112

Swap: 0 0 0

Mem: 8200781824 209956864 7662960640 475136 327864320 7746658304

Swap: 0 0 0

Mem: 8200781824 209952768 7662960640 475136 327868416 7746662400

Swap: 0 0 0

Mem: 8200781824 209944576 7662960640 475136 327876608 7746666496

Swap: 0 0 0

^Z

[1]+ Stopped free -b -s5

[root@iZuf6gwz144knq4rpuclj ~]# vmstat

procs-----memory-----swap-----io-----system-----cpu-----

r b swpd free buff cache si so bi bo in cs us sy id wa st

1 0 0 7482608 18968 301252 0 0 96 359 477 853 1 1 99 0 0

[root@iZuf6gwz144knq4rpuclj ~]#

default

命令终端 已连接 华东2(上海) i-uf6gwz144knq4rpuclj 106.14.10.522 vx1petouhq 14pt 39/35 39 Rows 132 Cols en\_US.UTF-8

实验五 多核多线程编程

剩余体验时间: 01:26:25

结束体验

实验手册云产品资源实验报告

体验云账号, 创建资源后生成

收起

子用户名: u-zngpw5hh@1915471854401972

子用户密码: Th5Ry5Kj7X9fa1R

AK ID: LTAI5IQV71KBAHPLKbY9KoNy

AK Secret: MPlaruG6rDnE7IGiY2u5wMlPUhw...

注意: 若登录子账号, 需打开控制台进行登录。

一键复制子账号登录链接

ECS服务器

磁盘ID: d-uf6533bjifujak3z53r

ECS公网地址: 106.14.10.5

ECS登录名: root

登录密码: lq7Sx1Ma9C

ECS实例ID: i-uf6gwz144knq4rpuclj

IP白名单: 0.0.0.0/0

地域: 华东 2 (上海)

2. root@iZuf6gwz144knq4rpuclj: ~

1138 ttyS0 Ss+ 0:00 1 41 110166 868 0.0 /sbin/agetty --keep-baud 115200,38400,9600 ttyS0 vt220

1139 tty1 Ss+ 0:00 0 41 110166 864 0.0 /sbin/agetty --noclear tty1 linux

1645 pts/0 Ss 0:00 0 885 114662 2064 0.0 -bash

1698 pts/0 T 0:00 0 11 152684 1472 0.0 free -b -s5

11974 pts/0 R+ 0:00 0 90 153237 1524 0.0 ps v

[root@iZuf6gwz144knq4rpuclj ~]# ping 192.168.1.7

PING 192.168.1.7 (192.168.1.7) 56(84) bytes of data.

^Z

[2]+ Stopped ping 192.168.1.7

[root@iZuf6gwz144knq4rpuclj ~]# ^C

[root@iZuf6gwz144knq4rpuclj ~]# ps v

PID TTY STAT TIME MAJFL TRS DRS RSS %MEM COMMAND

1138 ttyS0 Ss+ 0:00 1 41 110166 868 0.0 /sbin/agetty --keep-baud 115200,38400,9600 ttyS0 vt220

1139 tty1 Ss+ 0:00 0 41 110166 864 0.0 /sbin/agetty --noclear tty1 linux

1645 pts/0 Ss 0:00 0 885 114662 2064 0.0 -bash

1698 pts/0 T 0:00 0 11 152684 1472 0.0 free -b -s5

11992 pts/0 T 0:00 0 54 128501 1280 0.0 ping 192.168.1.7

12047 pts/0 R+ 0:00 0 90 153237 1524 0.0 ps v

[root@iZuf6gwz144knq4rpuclj ~]# free

total used free shared buff/cache available

Mem: 8008576 214612 7439024 496 354940 7553596

Swap: 0 0 0

[root@iZuf6gwz144knq4rpuclj ~]# vmstat

procs-----memory-----swap-----io-----system-----cpu-----

r b swpd free buff cache si so bi bo in cs us sy id wa st

1 0 0 7439116 24308 330640 0 0 7 25 457 857 0 0 100 0 0

[root@iZuf6gwz144knq4rpuclj ~]# kill -9 11992

-bash: kill -9: command not found

[root@iZuf6gwz144knq4rpuclj ~]# kill -9 11992

[root@iZuf6gwz144knq4rpuclj ~]# ps v

PID TTY STAT TIME MAJFL TRS DRS RSS %MEM COMMAND

1138 ttyS0 Ss+ 0:00 1 41 110166 868 0.0 /sbin/agetty --keep-baud 115200,38400,9600 ttyS0 vt220

1139 tty1 Ss+ 0:00 0 41 110166 864 0.0 /sbin/agetty --noclear tty1 linux

1645 pts/0 Ss 0:00 0 885 114662 2068 0.0 -bash

1698 pts/0 T 0:00 0 11 152684 1472 0.0 free -b -s5

21232 pts/0 R+ 0:00 0 90 153237 1524 0.0 ps v

[2]+ Killed ping 192.168.1.7

[root@iZuf6gwz144knq4rpuclj ~]#

default

命令终端 已连接 华东2(上海) i-uf6gwz144knq4rpuclj 106.14.10.522 vx1petouhq 14pt 39/35 39 Rows 132 Cols en\_US.UTF-8

内存管理程序通过映射机制把用户程序的逻辑地址映射到物理地址。当用户程序运行时，如果发现程序需要的虚拟地址没有对应的物理内存，即发出请求页要求。如果有空闲的内存可供分配，就请求分配内

存(用到内存分配和回收机制)，并把正在使用的物理页记录在缓存中（用到缓存机制）。如果没有足够的内存可供分配,则调用交换机制，腾出一部分内存。另外，在地址映射中要通过TLB( Translation Lookaside Buffer, 页表缓冲)寻找物理页；交换机制中用到交换缓存，并且把物理页内容交换到交换文件中，也要修改页表来映射文件地址。

## 程序代码

---

```
free
free -b -s5
vmstat
ps v
ping 192.168.1.7
kill -9 <PID>
```