

实 验 报 告

课程名称：操作系统试验

实 验 四：操作系统资源使用状态监控

班 级：

学生姓名：*Steven*

学 号：

专 业：

指导教师：

学 期：2022—2023 学年秋季学期

成 绩：

云南大学信息学院

一、实验目的

- 1、掌握操作系统中的文件目录结构，重点掌握 `proc` 目录下的文件结构；
- 2、掌握操作系统的内存、CPU、磁盘等状态信息的查询方式，以及状态信息中的字段含义；
- 3、掌握 `vmstat` 工具的基本使用方法，能够通过编程实现资源的动态申请及状态观察和验证。通过 `bonnie++` 基准测试程序，对操作系统进程测试，并分析结果。

二、知识要点

- 1、`proc` 文件结构和内容；
- 2、系统的主要状态信息查看方式，及其含义；
- 3、`vmstat` 系统状态监控程序的使用；
- 4、`bonnie++` 系统基准测试程序的测试和验证。

三、实验预习（要求做实验前完成）

- 1、了解 `linux` 系统中常用命令的使用方法；
- 2、掌握 `proc` 文件内容，系统状态的内容；
- 3、熟悉 `vmstat` 工具；
- 4、熟悉 `bonnie++` 基准测试工具。

四、实验内容和试验结果

结合课程所讲授内容以及课件中的试验讲解，完成以下试验。请分别描述程序的流程，附上源代码，并将试验结果截图附后。

1、编写 `shell` 脚本(命名为 `meminfo.sh`)，定时读取 `/proc/meminfo` 中的 `MemTotal`、`MemFree`、`Active` 等信息。编写一个负载程序，不断向操作系统申请内存，观察负载程序运行时 `meminfo` 中这三个数据的变化，并使用 `excel` 等工具绘制折线图。贴出负载程序的源码，并给出运行这个负载程序时的 `MemFree` 和 `Active` 的变化曲线。

(1) 源代码

`load.c`

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

```

int main()
{
    char *p;
    while(1){
        p = (char*) malloc(1024*sizeof(char));
        memset(p, 1, 1024);
        sleep(2);
    }
    return 0;
}

```

meminfo.sh

```

memfile="/proc/meminfo"

getmeminfo() {
    a=$(more $memfile)
    FREE=$(echo "$a" | awk 'NR==2{print $2}')
    PHYMEM=$(echo "$a" | awk 'NR==1{print $2}')
    ACTIVE=$(echo "$a" | awk 'NR==7{print $2}')
    Inactive=$(echo "$a" | awk 'NR==8{print $2}')
}

while true; do
    getmeminfo
    echo "Total Mem="$PHYMEM",FREE Mem="$FREE",Active
Mem="$ACTIVE", Inactive="$Inactive""
    sleep 2
done

```

(2) 运行结果:

```

Total Mem=7712804,FREE Mem=756928,Active Mem=1979848, Inactive=1595464
Total Mem=7712804,FREE Mem=746148,Active Mem=1979860, Inactive=1606452
Total Mem=7712804,FREE Mem=735376,Active Mem=1979836, Inactive=1617640
Total Mem=7712804,FREE Mem=721380,Active Mem=1979888, Inactive=1631184
Total Mem=7712804,FREE Mem=716564,Active Mem=1979928, Inactive=1635800
Total Mem=7712804,FREE Mem=704248,Active Mem=1979916, Inactive=1649400
Total Mem=7712804,FREE Mem=691076,Active Mem=1979856, Inactive=1662392
Total Mem=7712804,FREE Mem=681892,Active Mem=1979872, Inactive=1670724
Total Mem=7712804,FREE Mem=671692,Active Mem=1979884, Inactive=1681948
Total Mem=7712804,FREE Mem=647712,Active Mem=1979888, Inactive=1705152
Total Mem=7712804,FREE Mem=634188,Active Mem=1979904, Inactive=1716932
Total Mem=7712804,FREE Mem=622276,Active Mem=1979876, Inactive=1728248
Total Mem=7712804,FREE Mem=613080,Active Mem=1992048, Inactive=1732096
Total Mem=7712804,FREE Mem=602884,Active Mem=1992076, Inactive=1744976
Total Mem=7712804,FREE Mem=591296,Active Mem=1992156, Inactive=1753344
Total Mem=7712804,FREE Mem=591124,Active Mem=1992176, Inactive=1759120
Total Mem=7712804,FREE Mem=592932,Active Mem=1992484, Inactive=1764852
Total Mem=7712804,FREE Mem=591332,Active Mem=1992564, Inactive=1760692
Total Mem=7712804,FREE Mem=578944,Active Mem=1992592, Inactive=1773116
Total Mem=7712804,FREE Mem=564792,Active Mem=1992600, Inactive=1785808
Total Mem=7712804,FREE Mem=542784,Active Mem=1992628, Inactive=1807636
Total Mem=7712804,FREE Mem=538048,Active Mem=1992940, Inactive=1818632
Total Mem=7712804,FREE Mem=521180,Active Mem=1992924, Inactive=1830792
Total Mem=7712804,FREE Mem=511540,Active Mem=1992936, Inactive=1839988
Total Mem=7712804,FREE Mem=504800,Active Mem=1992936, Inactive=1845072
Total Mem=7712804,FREE Mem=499652,Active Mem=1992920, Inactive=1856268
Total Mem=7712804,FREE Mem=483044,Active Mem=1992916, Inactive=1869676
Total Mem=7712804,FREE Mem=465464,Active Mem=1992948, Inactive=1885208

```

(运行过程中的部分截图)

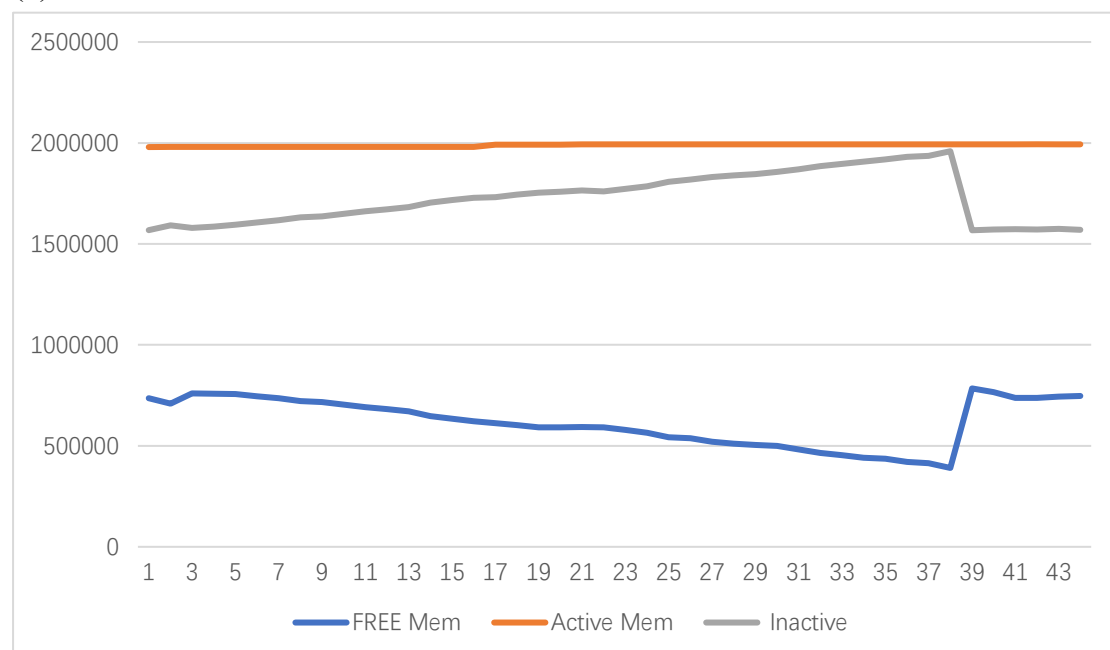
(3) 数据表

Total Mem	FREE Mem	Active Mem	Inactive
7712804	736036	1979740	1568540
7712804	709756	1979752	1591900
7712804	760120	1979768	1580056
7712804	758228	1979780	1585920
7712804	756928	1979848	1595464
7712804	746148	1979860	1606452
7712804	735376	1979836	1617640
7712804	721380	1979888	1631184
7712804	716564	1979928	1635800
7712804	704248	1979916	1649400
7712804	691076	1979856	1662392
7712804	681892	1979872	1670724
7712804	671692	1979884	1681948
7712804	647712	1979888	1705152
7712804	634188	1979904	1716932
7712804	622276	1979876	1728248
7712804	613080	1992048	1732096
7712804	602884	1992076	1744976
7712804	591296	1992156	1753344
7712804	591124	1992176	1759120
7712804	592932	1992484	1764852
7712804	591332	1992564	1760692
7712804	578944	1992592	1773116
7712804	564792	1992600	1785808
7712804	542784	1992628	1807636
7712804	538048	1992940	1818632
7712804	521180	1992924	1830792
7712804	511540	1992936	1839988
7712804	504800	1992936	1845072
7712804	499652	1992920	1856268
7712804	483044	1992916	1869676
7712804	465464	1992948	1885208
7712804	453636	1992968	1896552
7712804	441752	1992972	1907556
7712804	437108	1992988	1919420
7712804	420484	1992972	1931600
7712804	413580	1992964	1936576
7712804	390708	1992992	1959060
7712804	784568	1993016	1567688
7712804	766044	1993000	1572032
7712804	737900	1993004	1573360
7712804	737316	1993640	1571396

7712804	743628	1993020	1574436
7712804	746968	1993056	1570464

(运行过程中的全部数据)

(4) 内存变化图



当启动程序后，Free Memory 稳定降低，Inactive Memory 随之升高，说明负载程序正常执行。Active Memory 在此过程中保持稳定。图中第 38s 时结束负载程序，此时系统将占用的内存重新释放，各项数据回归最初未执行负载程序的状态。

2、使用 vmstat 工具监控系统中的资源使用状态，运行 bonnie++ 基准测试程序，观察、记录并分析 bonnie++ 运行过程中的数据变化。(vmstat 可定时读取相关的 proc 文件，按一定格式给出进程、内存、交换、磁盘 I/O、CPU 利用率等统计信息)。

```

^Croot@localhost:/C_projects/Exp4# bonnie++ -u root -r 256 -s 512
Using uid:0, gid:0.
Writing a byte at a time...done
Writing intelligently...done
Rewriting...done
Reading a byte at a time...done
Reading intelligently...done
start 'em...done...done...done...done...done...
Create files in sequential order...done.
Stat files in sequential order...done.
Delete files in sequential order...done.
Create files in random order...done.
Stat files in random order...done.
Delete files in random order...done.
Version 1.98      -----Sequential Output----- --Sequential Input- --Random-
-Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Name:Size etc    /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
localhost      512M 1816k 99 330m 40 332m 44 4113k 98 +++++ + + + + + + +
Latency         7117us 339us 1118us 15743us 56us 9531us
Version 1.98      -----Sequential Create----- -----Random Create-----
localhost        -Create-- --Read-- -Delete-- -Create-- --Read-- -Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
16 16384 46 16384 43 16384 50 16384 52 16384 41 16384 51
Latency         80933us 5342us 44322us 133ms 6586us 5303us
1.98,1.98,localhost,1.1669903090,512M,.8192,5,1816,99,337620,40,340324,44,4113,98,+++++,+++
0.2257,52.2994,41,3010,51,7117us,339us,1118us,15743us,56us,9531us,80933us,5342us,44322us,133

```

(执行 bonnie++)

1	0	2035544	570872	3980	2291232	0	0	0	0	0	0	0	0	100	0	0
1	0	2035544	570896	3980	2291232	0	0	0	0	0	0	0	0	100	0	0
1	0	2035032	568148	3980	2291256	0	0	0	0	0	0	0	0	100	0	0
1	0	2035032	575708	3980	2291088	0	0	0	0	0	0	0	0	100	0	0
1	0	2035032	587608	3980	2278180	0	0	0	0	0	0	0	0	100	0	0
1	0	2035032	604356	3980	2263356	0	0	0	0	0	0	0	0	100	0	0
1	0	2035032	625684	3980	2249668	0	0	0	0	0	0	0	0	100	0	0
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	2035032	639964	3980	2236036	0	0	0	0	0	0	0	0	100	0	0
1	0	2035032	671376	3980	2215356	0	0	0	0	0	0	0	0	100	0	0
1	0	2035032	659076	3980	2225612	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	653372	3980	2234280	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	640244	3976	2243000	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	636368	3968	2251748	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	623048	3952	2259492	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	607808	3952	2272344	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	583236	3952	2286612	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	577980	3944	2287920	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	576432	3944	2287380	0	0	0	0	0	0	0	0	100	0	0
1	0	2034520	581828	3944	2287124	0	0	0	0	0	0	0	0	100	0	0
1	0	2034520	581028	3944	2287496	0	0	0	0	0	0	0	0	100	0	0
1	0	2034520	581340	3944	2286896	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	584260	3944	2283012	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	614840	3944	2270856	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	674736	3944	2214076	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	692824	3944	2200668	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	713124	3944	2185640	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	727628	3944	2170780	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	726456	3944	2168040	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	725700	3944	2168040	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	732436	3944	2168068	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	730672	3944	2168068	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	735036	3944	2168068	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	731984	3944	2168068	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	727652	3944	2168084	0	0	0	0	0	0	0	0	100	0	0

（vmstat 监视过程）

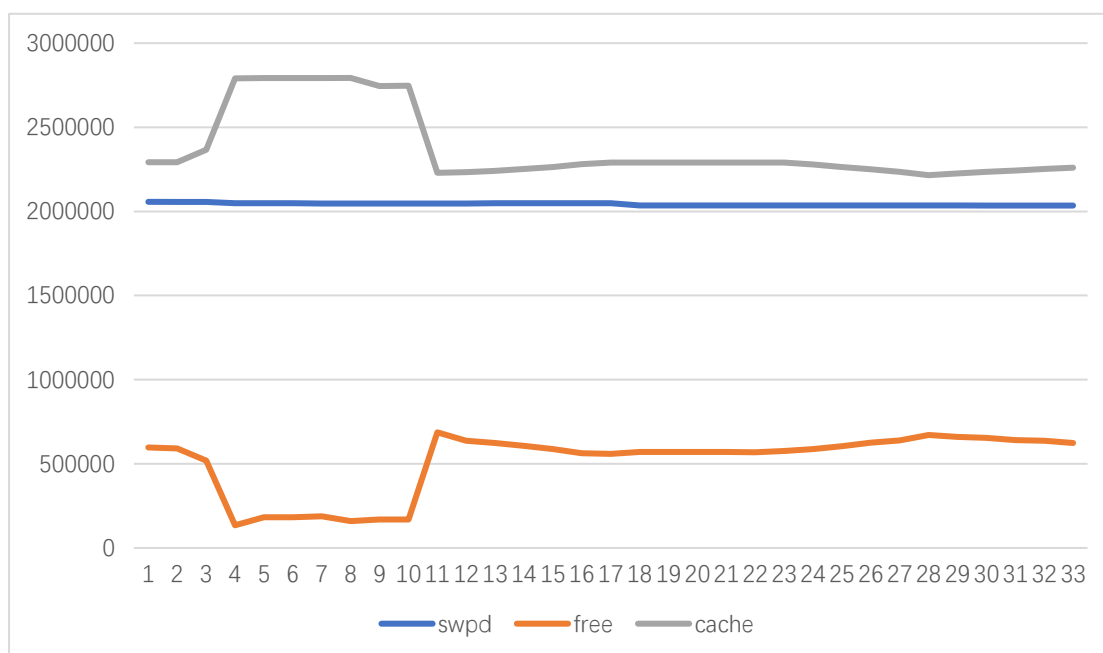
在 vmstat 监控到的数据中，选出变化明显的几列数据，绘制成以时间为横轴的折线图，每个折线图选两处典型的变化说明其意义。

在 vmstat 的数据中，以下 3 列变化相对明显

swpd	free	cache
2057048	596716	2292240
2056792	592396	2292360
2056792	519488	2365532
2048088	134716	2791200
2048088	182008	2791708
2048088	183280	2792520
2047576	188024	2792524
2047576	159344	2793288
2047576	169844	2745864
2047576	168712	2747752
2047576	687100	2229584
2047576	636756	2233908
2047832	624772	2241760
2047832	606060	2251964
2047832	587080	2263596
2047832	563420	2280124
2047832	558284	2290320
2035544	570456	2290624

2035544	570628	2291232
2035544	570872	2291232
2035544	570896	2291232
2035032	568148	2291256
2035032	575708	2291088
2035032	587608	2278180
2035032	604356	2263356
2035032	625684	2249668
2035032	639964	2236036
2035032	671376	2215356
2035032	659076	2225612
2034776	653372	2234280
2034776	640244	2243000
2034776	636368	2251748
2034776	623048	2259492

(数据汇总)



上表可以很明显看出, free 和 cache 两者之和大概为定值。在 3s 时, Bonnie++ 开始运行, Free 减少、Cache 增加。在 4s 时, Bonnie++ 运行时申请的 256MB 空间分配完成, Free 和 Cache 保持相对稳定。10s 时, Bonnie++ 运行完成, 删除了生成的临时文件, 释放内存, 同时 Cache 也对应减少。

五、问题讨论

1、proc/meminfo 中的 MemFree 表示什么? 观察它的变化, 当应用程序不断向操作系统请求内存时, MemFree 会逐渐降低, 当其降低到一定程度之后, 会发生什么变化? 修改 meminfo.sh 脚本, 监控/proc/meminfo 中更多的数据, 观察 MemFree

增大时，哪些数据减小？此时 MemFree 增加的内存来自哪里？

(1) /proc/meminfo 中的 MemFree 表示未使用的 RAM 量（单位：KB）

(2) 当应用程序不断向操作系统请求内存时，MemFree 会逐渐降低，而 MemActive 会逐渐增大，当 MemFree 降低到一定程度之后，若再申请较大的内存，即应用程序申请的内存比系统的总内存还多时，分配操作便会失败。

(3) 脚本修改如下：

meminfo_new.sh

```
memfile="/proc/meminfo"

getmeminfo() {
    a=$(more $memfile)
    FREE=$(echo "$a" | awk 'NR==2{print $2}')
    PHYMEM=$(echo "$a" | awk 'NR==1{print $2}')
    ACTIVE=$(echo "$a" | awk 'NR==7{print $2}')
    CACHE=$(echo "$a" | awk 'NR==5{print $2}')
}

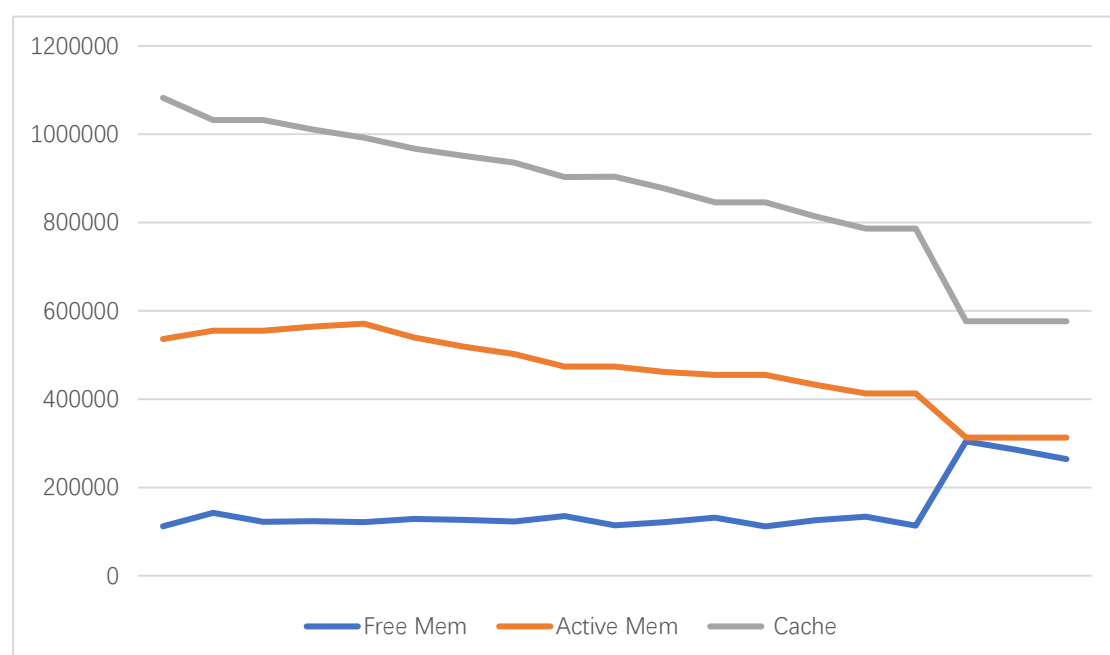
while true; do
    getmeminfo
    echo "Total Mem="$PHYMEM",FREE Mem="$FREE",Active
Mem="$ACTIVE", Cache="$CACHE""
    sleep 2
done
```

监视结果：

```
Total Mem=7712804,FREE Mem=277108,Active Mem=1950428, Cache=2512224
Total Mem=7712804,FREE Mem=274368,Active Mem=1950636, Cache=2513736
Total Mem=7712804,FREE Mem=272612,Active Mem=1950668, Cache=2513488
Total Mem=7712804,FREE Mem=272772,Active Mem=1950664, Cache=2513488
Total Mem=7712804,FREE Mem=278000,Active Mem=1949812, Cache=2513476
Total Mem=7712804,FREE Mem=276976,Active Mem=1949808, Cache=2513476
Total Mem=7712804,FREE Mem=278720,Active Mem=1949844, Cache=2513476
Total Mem=7712804,FREE Mem=280116,Active Mem=1949812, Cache=2513480
Total Mem=7712804,FREE Mem=268076,Active Mem=1949808, Cache=2513496
Total Mem=7712804,FREE Mem=236460,Active Mem=1949836, Cache=2513528
Total Mem=7712804,FREE Mem=225672,Active Mem=1949844, Cache=2513520
Total Mem=7712804,FREE Mem=203080,Active Mem=1949840, Cache=2513524
Total Mem=7712804,FREE Mem=190156,Active Mem=1949824, Cache=2513524
Total Mem=7712804,FREE Mem=177680,Active Mem=1949848, Cache=2514872
Total Mem=7712804,FREE Mem=174692,Active Mem=1949852, Cache=2514880
Total Mem=7712804,FREE Mem=164056,Active Mem=1949852, Cache=2514892
Total Mem=7712804,FREE Mem=151204,Active Mem=1949820, Cache=2514892
Total Mem=7712804,FREE Mem=144580,Active Mem=1949828, Cache=2514840
Total Mem=7712804,FREE Mem=129948,Active Mem=1949828, Cache=2514848
Total Mem=7712804,FREE Mem=117424,Active Mem=1949996, Cache=2514856
Total Mem=7712804,FREE Mem=112964,Active Mem=1959648, Cache=2514884
Total Mem=7712804,FREE Mem=99916,Active Mem=1959676, Cache=2514884
Total Mem=7712804,FREE Mem=127296,Active Mem=1942420, Cache=2483504
Total Mem=7712804,FREE Mem=121092,Active Mem=1942364, Cache=2483220
Total Mem=7712804,FREE Mem=110380,Active Mem=1942308, Cache=2483272
Total Mem=7712804,FREE Mem=94488,Active Mem=1942424, Cache=2483416
Total Mem=7712804,FREE Mem=97884,Active Mem=1956340, Cache=2468008
```

数据汇总：

FREE Mem	Active Mem	Cached
111844	536068	1082312
142252	555184	1032276
121840	555184	1032276
123356	564184	1010348
121556	570840	992156
128688	539504	967676
126600	519008	951128
122472	502240	935472
135368	473404	903212
114452	473840	903320
121136	461308	876648
131812	454612	845968
111668	454664	845968
125696	432708	813636
133512	413144	786060
113352	413160	786060
304228	312952	576304
284928	313108	576304
264516	313108	576304



2、说明程序中如何管理空闲页框。当 **bonnie++** 运行时，是否观察到内存交换？如果观察到内存交换，请截图，并说明交换发生在什么时候？（注意观察 **swpd**、**si**、**so**、**free**、**cache**、**buff** 等指标的变化。）

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	2035032	639964	3980	2236036	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2035032	671376	3980	2215356	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2035032	659076	3980	2225612	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	653372	3980	2234280	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	640244	3976	2243000	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	636368	3968	2251748	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	623048	3952	2259492	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	607808	3952	2272344	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	583236	3952	2286612	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	577980	3944	2287920	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034776	576432	3944	2287380	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034520	581828	3944	2287124	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034520	581028	3944	2287496	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034520	581340	3944	2286896	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	584260	3944	2283012	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	614840	3944	2270856	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	674736	3944	2214076	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	692824	3944	2200668	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	713124	3944	2185640	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	727628	3944	2170780	0	0	0	0	0	0	0	0	0	100	0	0
1	0	2034008	726456	3944	2168040	0	0	0	0	0	0	0	0	0	100	0	0

3、观察 CPU 利用率的变化情况（包括用户态（us）和系统态（sy）），在 bonnie++运行的过程中，当文件系统进行什么操作时，CPU 利用率最高？

在 us 和 sy 较高的时刻，bo 值较高，此时系统正在进行较频繁的写操作。