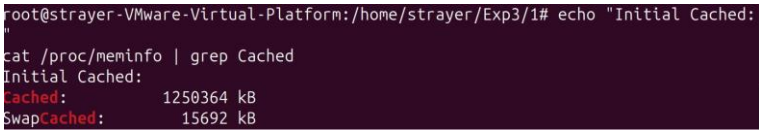


四川大学计算机学院、软件学院

实验报告

学号: 2023141460321 姓名: 孙谦昊 专业: 计算机科学与技术 班级: 行政七班 第 12 周

课程名称	操作系统实验	实验课时	8-9 节
实验项目	Linux 虚拟内存管理	实验时间	2025.05.07
实验目的	1.熟悉 Linux 内存管理的常见命令 2.理解 Linux 内存管理机制		
实验环境	VM WorkStation Pro、ubuntu-24.04.2-desktop-amd64		
实 验 内 容 (算法、程序、步骤和方法)	<p><b>1. 创建一个大文件（512M）,然后清除缓存，使用程序读取该文件并计时，使用 cat /proc/meminfo 观察运行前后 cached 使用情况；然后再次重复运行程序读取，比较两次速度差异，说明原因。</b></p> <p>为满足实验第 1 部分的要求，首先借助 dd if=/dev/zero of=/tmp/bigfile bs=1M count=512 命令创建一个大小为 512M 的文件。接下来分别借助：</p> <pre>cat /proc/meminfo   grep Cached time dd if=/tmp/bigfile of=/dev/null sudo sync &amp;&amp; echo 3   sudo tee /proc/sys/vm/drop_caches cat /proc/meminfo   grep Cached time dd if=/tmp/bigfile of=/dev/null cat /proc/meminfo   grep Cached</pre> <p>命令对此时操作系统的缓存初始状态进行查看、第一次读取文件并计时、清除缓存、查看清除缓存后的缓存情况、第二次读取文件并计时和查看最终缓存情况。</p> <p><b>2. 执行 find / -name .conf 命令，使用 cat /proc/meminfo 观察执行前后 buffers 使用情况；然后再次执行命令，比较两次速度差异，说明原因。</b></p> <p>为满足实验第 2 部分的要求，在 Linux 终端中分别输入以下命令：</p> <pre>cat /proc/meminfo   grep Buffers time find / -name .conf cat /proc/meminfo   grep Buffers time find / -name .conf cat /proc/meminfo   grep Buffers</pre>		

	<p>命令对操作系统的 buffers 初始情况进行查看、执行 find 命令在目录树中查找文件、查看 find 命令执行后操作系统的 buffers 的情况、再次执行 find 命令在目录书中查找文件和查看操作系统的 buffers 的最终情况。</p> <p>3. 运行内存消耗程序，直到耗尽系统内存 进程被杀死，使用 top 命令观测在这个过程中 free, used, buffer/cache, swap 使用情况变化，记录并说明原因。（建议虚拟机分配 4G 内存，实验前注意保存文件）。</p> <p>为满足实验第 2 部分的要求，首先创建一个简单的 Python 脚本 memory_consumer.py，并在 memory_consumer.py 中编写入以下部分代码：</p> <pre>import os try:     memory = []     while True:         memory.append([0] * 1000000) except MemoryError:     print('Memory exhausted') EOF</pre> <p>随后借助 python3 /tmp/memory_consumer.py &amp; PID=\$! 运行该程序、借助 top 命令 top -b -n 10 -p \$PID 实时观察内存的使用情况。借助：</p> <pre>while ps -p \$PID &gt; /dev/null; do     cat /proc/meminfo   grep -E 'MemFree MemUsed Buffers Cached SwapTotal SwapFree'     sleep 5 done</pre> <p>不断的查看内存的使用情况。同时借助 wait \$PID 等待操作系统自动终止（kill）该程序。</p>
数据记录和计算	<p>一、实验环节 1</p> <p>在创建大文件后，查看操作系统此时的内存为：</p>  <pre>root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/1# echo "Initial Cached: " cat /proc/meminfo   grep Cached Initial Cached: Cached:      1250364 kB SwapCached:    15692 kB</pre> <p>Cached 表示当前被操作系统缓存的文件数据的大小为 1250364KB，SwapCached 表示当前被缓存的交换空间数据的大小为 15692KB。在第一次读取文件并计时、清除缓存、查看清除缓存后的缓存情况后，可以查看到操作系统的内存变化为：</p>

```

root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/1# echo "First read:"
time dd if=/tmp/bigfile of=/dev/null
First read:
1048576+0 records in
1048576+0 records out
536870912 bytes (537 MB, 512 MiB) copied, 5.80284 s, 92.5 MB/s

real    0m5.808s
user    0m1.512s
sys     0m4.075s

root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/1# sudo sync && echo 3 |
sudo tee /proc/sys/vm/drop_caches
3
root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/1# echo "Cached after cl
earing:"
cat /proc/meminfo | grep Cached
Cached after clearing:
Cached:          460168 kB
SwapCached:      15896 kB

```

观察到第一次读取文件一共耗时 0m5.808s, 清除缓存后的 cached 和 SwapCached 分别变为 460168KB 和 15896KB。

在第二次读取文件并计时和查看最终缓存情况时, 可以观察到:

```

root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/1# echo "Second read:"
time dd if=/tmp/bigfile of=/dev/null
Second read:
1048576+0 records in
1048576+0 records out
536870912 bytes (537 MB, 512 MiB) copied, 6.18173 s, 86.8 MB/s

real    0m6.192s
user    0m0.542s
sys     0m4.962s

root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/1# echo "Final Cached:"
cat /proc/meminfo | grep Cached
Final Cached:
Cached:          985444 kB
SwapCached:      15964 kB

```

第二次读取文件一共耗时 0m6.192s, cached 和 SwapCached 分别为 985444KB 和 15964KB。

## 二、实验环节 2

在对操作系统的 buffers 初始情况进行查看时, 可以观察到:

```

root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/2# echo "Initial Buffers
:"
cat /proc/meminfo | grep Buffers
Initial Buffers:
Buffers:         11604 kB

```

操作系统初始的 buffers 大小为 11604KB。在执行 find 命令在目录树中查找文件和查看 find 命令执行后, 操作系统的 buffers 的情况变为:

```

root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/2# echo "First find comm
and:"
time find / -name .conf
First find command:
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
find: '/proc/4346/task/4346/net': Invalid argument
find: '/proc/4346/net': Invalid argument

real    0m8.630s
user    0m1.005s
sys     0m3.907s
root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/2# echo "Buffers after f
irst find:"
cat /proc/meminfo | grep Buffers
Buffers after first find:
Buffers:         99852 kB

```

操作系统总共耗时 0m8.630s 在整个目录树中查找指定的文件。在执行完系

列指令后操作系统的 buffers 使用量为 99852KB。

随后再次执行 find 命令在目录书中查找文件和查看操作系统的 buffers 的最终情况，可以观察到：

```
root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/2# echo "Second find command:"
time find / -name .conf
Second find command:
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
find: '/proc/4346/task/4346/net': Invalid argument
find: '/proc/4346/net': Invalid argument

real    0m3.476s
user    0m1.475s
sys     0m1.806s
root@strayer-VMware-Virtual-Platform:/home/strayer/Exp3/2# echo "Final Buffers:"

cat /proc/meminfo | grep Buffers
Final Buffers:
Buffers:          99868 kB
```

操作系统总共耗时 0m3.476s 来执行在整个目录树中查找指定的文件。在执行完系列操作后操作系统的 buffers 使用量为 99868KB，基本上没有变化。

### 三、实验环节 3

执行系列命令后，操作系统的终端输出如下内容：

```
Running memory consumer program...
[2] 6839
Monitoring memory usage with top...
top - 20:09:10 up 30 min, 1 user, load average: 1.50, 1.14, 0.90
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6.1 us, 3.4 sy, 0.0 ni, 90.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3866.7 total, 792.9 free, 2441.0 used, 1075.3 buff/cache
MiB Swap: 3741.0 total, 3301.0 free, 440.0 used, 1425.7 avail Mem

  PID USER      PR  NI   VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 6839 root        20   0 122776 97628 6492 R 90.9   2.5   0:00.21 python3

top - 20:09:13 up 30 min, 1 user, load average: 1.70, 1.19, 0.92
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.4 us, 8.1 sy, 0.0 ni, 86.2 id, 0.8 wa, 0.0 hi, 0.5 si, 0.0 st
MiB Mem : 3866.7 total, 331.1 free, 3437.0 used, 464.1 buff/cache
MiB Swap: 3741.0 total, 3106.9 free, 634.1 used, 429.7 avail Mem

  PID USER      PR  NI   VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 6839 root        20   0 1553104 1.5g 6364 R 98.0  38.4   0:03.16 python3

top - 20:09:16 up 30 min, 1 user, load average: 1.70, 1.19, 0.92
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie

%Cpu(s): 5.5 us, 13.9 sy, 0.0 ni, 77.7 id, 2.0 wa, 0.0 hi, 1.0 si, 0.0 st
MiB Mem : 3866.7 total, 105.5 free, 3739.9 used, 351.6 buff/cache
MiB Swap: 3741.0 total, 2574.1 free, 1166.9 used, 126.8 avail Mem

  PID USER      PR  NI   VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 6839 root        20   0 2451944 2.1g 3036 R 97.7  55.7   0:06.09 python3

top - 20:09:19 up 30 min, 1 user, load average: 1.80, 1.22, 0.93
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.5 us, 14.5 sy, 0.0 ni, 79.7 id, 1.6 wa, 0.0 hi, 0.8 si, 0.0 st
MiB Mem : 3866.7 total, 153.9 free, 3689.0 used, 353.2 buff/cache
MiB Swap: 3741.0 total, 1918.3 free, 1822.7 used, 177.7 avail Mem

  PID USER      PR  NI   VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 6839 root        20   0 3069408 2.2g 3036 R 91.4  57.1   0:08.85 python3

top - 20:09:22 up 30 min, 1 user, load average: 1.80, 1.22, 0.93
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.7 us, 16.8 sy, 0.0 ni, 68.6 id, 7.3 wa, 0.0 hi, 3.6 si, 0.0 st
MiB Mem : 3866.7 total, 86.9 free, 3791.0 used, 236.6 buff/cache
MiB Swap: 3741.0 total, 1217.0 free, 2524.0 used, 75.7 avail Mem

  PID USER      PR  NI   VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 6839 root        20   0 3929168 2.6g 2524 R 95.3  68.5   0:11.71 python3
```

```

top - 20:09:25 up 30 min, 1 user, load average: 1.98, 1.27, 0.95
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.7 us, 19.4 sy, 0.0 ni, 69.5 id, 5.0 wa, 0.0 hi, 2.4 si, 0.0 st
MiB Mem : 3866.7 total, 188.5 free, 3680.2 used, 243.5 buff/cache
MiB Swap: 3741.0 total, 617.6 free, 3123.4 used. 186.5 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  6839 root        20   0 4413760  2.5g 2524 R  94.7   65.2   0:14.56 python3

top - 20:09:28 up 30 min, 1 user, load average: 2.38, 1.36, 0.98
Tasks: 1 total, 0 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.4 us, 18.4 sy, 0.0 ni, 63.3 id, 9.5 wa, 0.0 hi, 3.4 si, 0.0 st
MiB Mem : 3866.7 total, 207.6 free, 3639.1 used, 300.6 buff/cache
MiB Swap: 3741.0 total, 68.5 free, 3672.5 used. 227.6 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  6839 root        20   0 4906168  2.3g 2524 D  82.4   61.7   0:17.04 python3

top - 20:09:34 up 30 min, 1 user, load average: 4.35, 1.79, 1.12
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.5 us, 26.8 sy, 0.0 ni, 27.6 id, 41.9 wa, 0.0 hi, 2.3 si, 0.0 st
MiB Mem : 3866.7 total, 1467.7 free, 2465.6 used, 141.5 buff/cache
MiB Swap: 3741.0 total, 0.0 free, 3741.0 used. 1401.1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  6839 root        20   0      0      0      0  R   79.4    0.0   0:22.17 python3

top - 20:09:37 up 30 min, 1 user, load average: 4.35, 1.79, 1.12
Tasks: 0 total, 0 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.8 us, 8.7 sy, 0.0 ni, 73.6 id, 11.7 wa, 0.0 hi, 1.1 si, 0.0 st
MiB Mem : 3866.7 total, 2520.4 free, 1316.5 used, 307.9 buff/cache
MiB Swap: 3741.0 total, 2475.6 free, 1265.4 used. 2550.2 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  6839 root        20   0      0      0      0  R   79.4    0.0   0:22.17 python3

top - 20:09:40 up 30 min, 1 user, load average: 4.24, 1.81, 1.13
Tasks: 0 total, 0 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.3 us, 2.8 sy, 0.0 ni, 86.7 id, 5.6 wa, 0.0 hi, 0.7 si, 0.0 st
MiB Mem : 3866.7 total, 2446.0 free, 1381.9 used, 317.0 buff/cache
MiB Swap: 3741.0 total, 2499.3 free, 1241.7 used. 2484.8 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  6839 root        20   0      0      0      0  R   79.4    0.0   0:22.17 python3

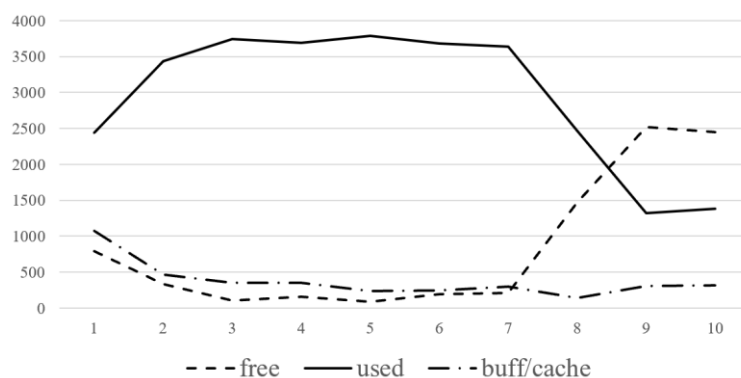
[2]+  Killed                  python3 /tmp/memory_consumer.py

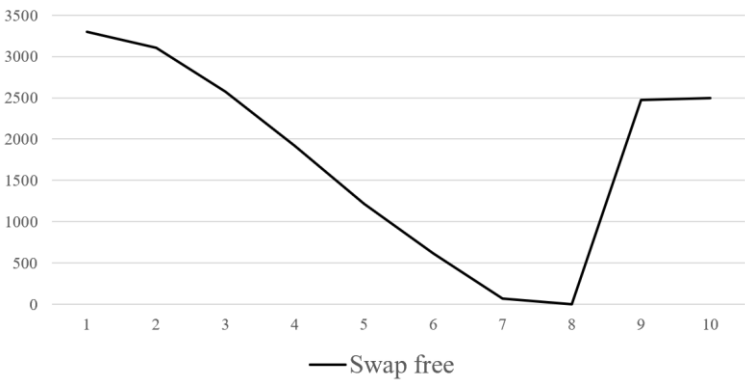
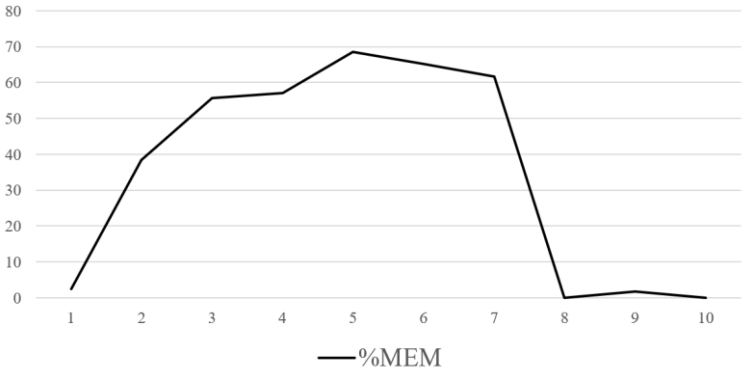
Waiting for the memory consumer program to be terminated by the system...
Final memory usage:
MemFree:        2492032 kB
Buffers:         2452 kB
Cached:          254368 kB
SwapCached:       85696 kB
SwapTotal:       3830780 kB
SwapFree:        2561356 kB

```

为方便观察 free, used, buffer/cache, swap 使用情况变化, 做下图:

MiB Mem各指标变化图



	<div>Swap free指标变化图</div>  <table border="1"><thead><tr><th>Iteration</th><th>Swap free</th></tr></thead><tbody><tr><td>1</td><td>3200</td></tr><tr><td>2</td><td>3100</td></tr><tr><td>3</td><td>2800</td></tr><tr><td>4</td><td>2200</td></tr><tr><td>5</td><td>1500</td></tr><tr><td>6</td><td>1000</td></tr><tr><td>7</td><td>500</td></tr><tr><td>8</td><td>100</td></tr><tr><td>9</td><td>2500</td></tr><tr><td>10</td><td>2500</td></tr></tbody></table> <div>%MEM指标变化图</div>  <table border="1"><thead><tr><th>Iteration</th><th>%MEM</th></tr></thead><tbody><tr><td>1</td><td>2</td></tr><tr><td>2</td><td>38</td></tr><tr><td>3</td><td>55</td></tr><tr><td>4</td><td>57</td></tr><tr><td>5</td><td>68</td></tr><tr><td>6</td><td>65</td></tr><tr><td>7</td><td>62</td></tr><tr><td>8</td><td>0</td></tr><tr><td>9</td><td>0</td></tr><tr><td>10</td><td>0</td></tr></tbody></table> <p>从图中可以看到：操作系统在第 10 次时彻底终结了程序。可以观察到 free、buff/cache 和 Swap free 指标在第 1 至 7 次逐渐减小。在第 8 次运行时 free 和 Swap free 陡然增大。此外 used 和 %MEM 在第 1 值 6 次逐渐增大，在第 7 次运行后陡然降低。由其是 %MEM 在第 8 次运行时降低为 0。</p>	Iteration	Swap free	1	3200	2	3100	3	2800	4	2200	5	1500	6	1000	7	500	8	100	9	2500	10	2500	Iteration	%MEM	1	2	2	38	3	55	4	57	5	68	6	65	7	62	8	0	9	0	10	0
Iteration	Swap free																																												
1	3200																																												
2	3100																																												
3	2800																																												
4	2200																																												
5	1500																																												
6	1000																																												
7	500																																												
8	100																																												
9	2500																																												
10	2500																																												
Iteration	%MEM																																												
1	2																																												
2	38																																												
3	55																																												
4	57																																												
5	68																																												
6	65																																												
7	62																																												
8	0																																												
9	0																																												
10	0																																												
结 论 (结 果)	<p>一、实验 1 环节</p> <p>在在第一次读取文件时，由于系统缓存中没有该文件的内容，因此需要从磁盘读取。磁盘的读取速度相对较慢，这导致了读取耗时较长，具体为 0m5.808s。随后，通过执行命令清空了系统缓存。这一操作使得系统中原本存储的文件数据被清除，为后续的文件读取操作腾出了空间。在第二次读取文件时，情况与第一次类似，系统依然需要从磁盘读取文件内容。然而，第二次读取的耗时（0m6.192s）略长于第一次。这可能是由于磁盘性能的波动、系统负载的变化，或者其他因素导致磁盘响应时间稍长。</p> <p>缓存可以显著提高文件读取速度，因为从内存中读取数据的速度比从磁盘读取快得多。当缓存被清除后，系统需要重新从磁盘加载文件内容，这使得读取速度变慢。</p> <p>二、实验 2 环节</p> <p>在第一次执行 <code>`find / -name .conf`</code> 命令时，由于系统 Buffers 中尚未缓存足够的文件系统元数据，因此系统需要从磁盘加载大量相关数据。这一过</p>																																												



	<p>程涉及大量的磁盘 I/O 操作，导致耗时较长，具体为 0m8.630s。同时，为了存储这些新加载的元数据，Buffers 的使用量显著增加。</p> <p>而在第二次执行相同命令时，情况则有所不同。此时，系统已经将大部分文件系统的元数据缓存到了 Buffers 中。因此，当再次执行该命令时，系统可以直接从内存中的 Buffers 快速读取这些元数据，而无需再次从磁盘加载。这大大减少了磁盘 I/O 操作，从而显著提高了命令的执行速度，耗时仅为 0m3.476s。同时，由于大部分必要的元数据已经缓存，Buffers 的使用量几乎没有变化。</p> <p>三、实验 3 环节</p> <p>在实验过程中，随着脚本的运行，系统内存逐渐耗尽。在前 7 次运行中，程序不断分配内存，导致系统可用内存（free）逐渐减少。同时，系统为了优化性能，会将部分未使用的内存用于缓存和缓冲区（buff/cache），因此这部分内存也会逐渐减少。当系统内存不足时，系统会尝试使用交换空间（Swap），因此 Swap free 也会逐渐减少。</p> <p>到了第 8 次运行时，系统检测到内存不足，触发了内存耗尽的保护机制。操作系统会终止占用大量内存的进程，即 memory_consumer.py，以防止系统崩溃。终止进程后，系统释放了该进程占用的内存，因此 free 和 Swap free 突然增加。由于进程被终止，used 和 %MEM 突然减少，特别是 %MEM 降低为 0，表示该进程已不再占用任何内存。</p>
小 结	通过上述实验结果可知成功完成了实验要求
指导老师 评 议	<p>成绩评定：</p> <p>指导教师签名：</p>