Project Report 3
The ms (“memory status”) Program

Team Members:
Howard Grimberg
Aditya Balasubramanian

Submitted: 5/9/2013

Purpose

The purpose of the ms program is to use the /proc virtual file system in Linux to read various statistics about the memory on the system. Cumulative totals and per process statistics were gathered and displayed in such a way to make comparisons easy.

Implementation

For the cumulative totals, the '/proc/meminfo' file was scanned and the choice bits extracted and formatted neatly. Not all statistics are available in that location, so other metrics had to be computed differently. For example, the amount of memory in use was calculated by subtracting the free memory from the total amount of memory available. The other cumulative metrics where tabulated by scanning the entire ProcessList object and summing the appropriate values.

For the individual totals, all PIDs (in use or not) were obtained by scanning the /proc directory and capturing everything that was an integer. Each PID was then given an instances of the Process class that was responsible for updating itself when asked (explained later). The Process class will scan the subdirectory of its PID in /proc and obtain the necessary info from the smaps and the cmdline file. The smaps file contains statistics for each chunk of memory allocated to that given PID. As such, a sum was taken across all of the memory blocks with some not-very-clever text processing kung-fu. Originally, I wanted the ms program to be more like top and be constantly updating rather than taking a snapshot. For that reason, every PID has its own class that updates itself. Sorting was done using the builtin in sorted() routine in python and careful massaging of the ProcessList class into tuples.

Testing

Testing was very simple. The project was tested on a personal linux computer (so permissions were not an issue). A random, large process was chosen (rhythmbox) and the output captured using ms. Top and gnome-system-monitor were used to confirm that the memory values were about the same. Memory maps can be obtained in gnome-system-monitor by right clicking on a process and selecting memory maps. The program was then run in the VM with the specified test subjects and documented. One terminal was used to run the test subject while the other terminal was used to run ms rapidly in succession (using fast keystrokes and bash history) after a 10 second warm-up time. The swap file was turned off and on using swapoff- a and swapon –a between each test to ensure no residual data was left in swap space.

Process Description

<u>Mystery Program Aether :</u>

```
Total Usage

Total      297832   kB

In Use      66244    kB

Free       231588   kB

Shared     114748   kB

Nonshared  12044    kB

ID      USS       PSS       SWAP      RES       SHR       CMD

2509           24     10270       0      20636     20612 ./aether

2507           24      6856       0      20636     20612 ./aether

2506           24      6856       0      20636     20612 ./aether

2505           24      6856       0      20636     20612 ./aether

2508           24     10270       0      20636     20612 ./aether
```

The aether process follows the below memory characteristics .Processes 2509 and 2508 share the same page size of (10270*2 = 20540 KB) and processes 2505,2506 and 2507 share the page size of (6586*3) 19758 KB.

• (d) Two buffers of the same size are allocated. Two children share one of the buffers, and the

other three children share the other. Each child requires only a small amount of unique memory.

<u>Mystery Program hemera - A</u>

```
ID      USS       PSS       SWAP      RES       SHR       CMD

2524         20504     20510       0     20644      140 ./hemera

2523         16408     16414       0     16548      140 ./hemera
```

```
2522         12312     12318        0     12452       140 ./hemera

2521          8216      8222        0      8356       140 ./hemera

2520          4120      4126        0      4260       140 ./hemera
```

The hemera process follows the below memory characteristic. The Unique memory is relatively large as compared to the Shared memory.

• (a) each child allocates their own (relatively large) buffer of unique memory. Each child requires

a different amount of unique memory. Each child shares a relatively small portion of its address

space with the other children.

<u>Mystery Program ourea-C</u>

```
Total Usage

Total      297832   kB

In Use      45844    kB

Free       251988   kB

Shared     114964   kB

Nonshared   11936    kB

PID       USS       PSS       SWAP       RES       SHR       CMD

2549            24      4124        0      20628     20604 ./ourea

2551            24      4124        0      20628     20604 ./ourea

2552            24      4124        0      20628     20604 ./ourea

2550            24      4124        0      20628     20604 ./ourea

2553            24      4124        0      20628     20604 ./ourea
```

The Mystery program ourea follows the statement C characteristic as each process has a small unique memory (USS 24) and the Shared memory buffer is large and same for all the processes which is SHR 20604

- (c) The children all share a single, large common buffer. Each child requires only a small amount of unique memory.

Mystery Program pontus-E

```
Total Usage

Total      297832   kB

In Use      86884   kB

Free       210948   kB

Shared      73996   kB

Nonshared  52892    kB

PID      USS      PSS      SWAP      RES      SHR      CMD

2564         4120     9586        0     20628     16508 ./pontus

2561        16408     17778        0     20628      4220 ./pontus

2563         8216     12316        0     20628     12412 ./pontus

2562        12312     15047        0     20628      8316 ./pontus

2565           24      6855        0     20620     20596 ./pontus
```

The Mystery Program Pontus follows the below characteristic.

As can be seen each process shares the same amount of Resident memory (RES) but has different values for Shared memory which it has shared with other children (SHR).

- (e) Each child requires about the same amount of resident memory. However, each child shares a different amount of its required memory with the other children.

Mystery Program Uranus-F

Total Usage

```
Total      297832   kB

In Use      128404   kB

Free       169428   kB

Shared     19056    kB

Nonshared  114740   kB

PID      USS      PSS      SWAP      RES      SHR      CMD

2570         20600     20775        0     22008      1408 ./uranus

2569         20600     20775        0     22008      1408 ./uranus

2571         20600     20775        0     22008      1408 ./uranus

2572         20600     20775        0     22008      1408 ./uranus

2573         20600     20775        0     22008      1408 ./uranus
```

Mystery Program Uranus follows the F description as the size of unique buffer is same for all the processes and it the Shared memory is small relatively to the amount of unique memory. The difference between this process and the other processes is that it doesn't swap out its memory when it is out of physical memory.

We ran the process with limited memory of 300MB and ran the same process in other terminals (before all the processes completed) such that the entire 300 MB RAM is exhausted. We continuously monitored the memory status by running ms each time. After a while when we tried to run the process again, we couldn't because it had run out of memory and the memory wasn't swapped out to the disk.

• (f) Each child allocates their own (relatively large) buffer of unique memory. Each child's unique buffer is about the same size as each other child. Each child shares a relatively small portion of its address space with the other children. In contrast to all of the other programs, this program uses a system call to lock all of its allocated memory in RAM. None of its allocated memory will be swapped out to disk, even when the system runs out of memory

<u>Mystery Program typhoon-B</u>

Total Usage

```
Total      297832   kB
```

```
In Use     127092   kB

Free       170740   kB

Shared     12328    kB

Nonshared  114304   kB

PID      USS       PSS       SWAP      RES       SHR       CMD

2483         20504     20509       0     20632       128 ./typhon

2486         20504     20509       0     20632       128 ./typhon

2484         20504     20509       0     20632       128 ./typhon

2485         20504     20509       0     20632       128 ./typhon

2482         20504     20509       0     20632       128 ./typhon
```

The Mystery Program typhon follows the below statement characteristic as the unique memory for each memory is same and large in this case (USS is 20504) and Shared memory is small (SHR is 128).

- (b) Each child allocates their own (relatively large) buffer of unique memory. Each child's unique buffer is about the same size as each other child. Each child shares a relatively small portion of its address space with the other children.