CS401 - ASSIGNMENT #2

Estimation of WSS (Working Set Size)

What is WSS?

WSS (Working Set Size) is the amount of physical memory used by a process in a specific amount of time to accomplish a task. WSS is a subset of RSS(Resident Set Size) which is the amount of address space present in physical memory.

How is WSS calculated?

Kernel manages memory in units of Pages. Pages are small sized chunks of bytes. It is usually of size 4KB in a 32 bit system. Page table keeps track of all the pages allotted to a process. Each process has its own page table.

The procedure I used to calculate WSS is to periodically walk through the page table entries and make certain modifications. Page table had Page Table Entries (PTE) which contains address in physical memory and some meta data associated with it. The PTE contains many bits which indicate the status of the page. The bits that are of importance here are the Valid bit / Present Bit and Protection bit. If present bit (_PAGE_PRESENT) is set, it means that the page is available in memory. If the protection bit (_PAGE_PROTNONE) is set, it means that the page is available in memory but it is not accessible for user space programs. So the procedure is to trick the program in thinking of page faults and counting them so that we know how many pages are accessed by the process.

First, we create a hook in kernel page fault handling mechanism so that The kernel module iterates over all PTEs of a process and checks if the page is present in physical memory. If it is, then we make it not present and protected. When the page fault happens, our hook inside kernel will call us and we check if it was because of our change and roll it back. Thus the process works without knowing anything and we can calculate the WSS.

High Level Algorithm

1. Get task_struct of process we want to calculate WSS
2. Initialize the hook inside kernel
3. For each VMA region inside mm_struct
    - 3.1. For each address incremented in PAGE_SIZE units
        - 3.1.1. Get Page Table Entry
        - 3.1.2. If Present Bit is set, clear it and set protected bit
        - 3.1.3. Add it to ADDR_LIST
        - 3.1.4. Repeat until end of VMA region
    - 3.2. Repeat until end of VMAs
4. When page fault happens, the hook function will be called, inside that
    - 4.1. If it is from same process
        - 4.1.1. Get PTE and check if protected and no present and Check if present inside ADDR_LIST.
        - 4.1.2. If so set present bit and clear protected bit
        - 4.1.3. WSS_count++
5. Wait for some amount of time
6. Print WSS_count
7. Repeat N times

How to verify Page Size?

It can be verified using /proc/meminfo and /proc/vmstat.

PAGE_SIZE = Mapped/nr_mapped

How Page Table is organized?

Page Tables in linux are organized as 3 levels. They are

1. Page Global Directory (PGD)
2. Page Upper Directory (PUD)
3. Page Middle Directory (PMD)

To access a page table entry (PTE), we need to go through these levels.
The base pointer of page table of a process (ie PGD) can be found in mm_struct inside task_struct (`mm_struct_pgd`). The kernel provided methods `pgd_offset(),pud_offset(),pmd_offset()` and `pte_offset()` provides access to PTE. The present bit can be checked using `_PAGE_PRESENT` flag and protected bit using `_PAGE_PROTNONE`.

Components

1. Kernel Hook in page fault handler which calls the function in our kernel module when it is inserted
2. Kernel module which run periodically to estimate WSS. It depends on the kernel hook.

How to apply patch?

The patch is to be applied to fault.c file. Assuming you are inside the kernel source root, it can be applied as

```
patch arch/x86/mm/fault.c < fault.c.patch
```

Then you need to compile and install kernel.

How to compile the module?

1. Change directory to ./wss/modules/
2. `make`
3. The file wss.ko will be generated and it is the module file

How to execute ?

1. Go to folder where wss.ko resides
2. Find PID of the process which you want to find the WSS using the command "ps -e"
3. `sudo insmod wss.ko pid=PID ntimes=N` # where PID is you process id and N is the number of times wss should be calculated in 10 second intervals

Eg: `sudo insmod wss.ko pid=4825 ntimes=4`

4. After insertion, you can unload the module using

```
sudo rmmod wss
```

NOTE :- Do NOT close program before removing module. It might cause memory problems.

How to get results?

Results are printed in system log. You can access it using the command

```
dmesg
```

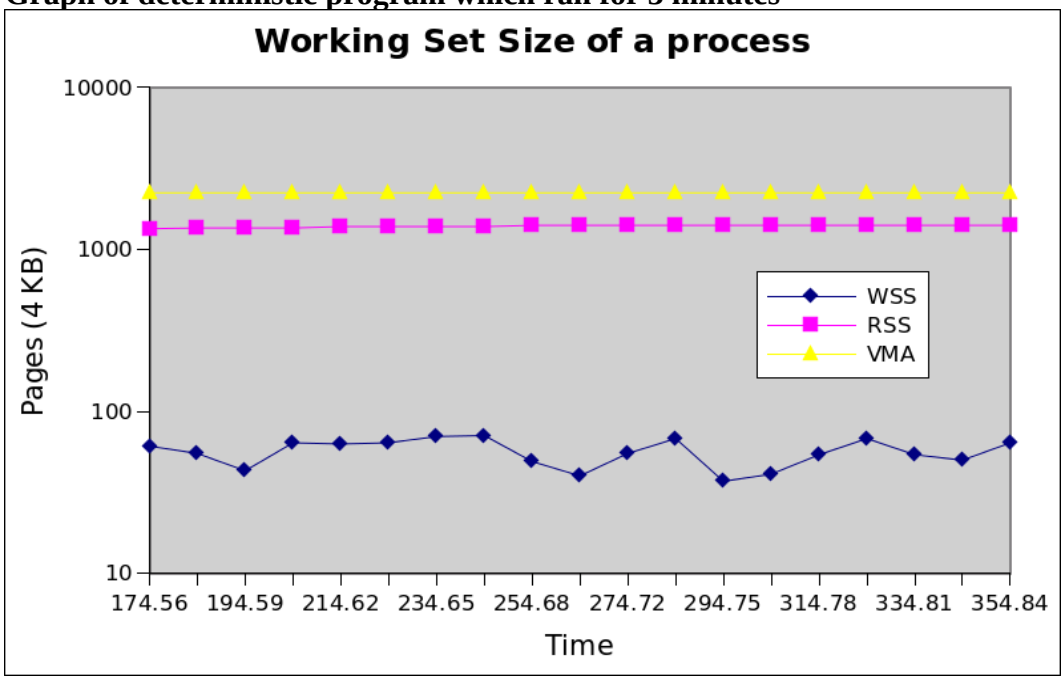
Testing Strategy

I have created a simple user space program which has an array of 4KB data items. The program access random number of array items every 10 seconds which is similar to page accesses. It shows number of items accessed. My WSS estimator module also run every 10 seconds. So we can cross check the number of page accessed.

Deterministic Outputs

[12528.314703] Module started..
[12528.314703] Page size : 4096
[12528.314712] Process name with PID 6167 is : jumper
Accessing 36 elements in array
[12538.336181] WSS : 61 RSS : 1204 VMA : 2236
Accessing 27 elements in array
[12548.353030] WSS : 55 RSS : 1216 VMA : 2236
Accessing 15 elements in array
[12558.368170] WSS : 43 RSS : 1216 VMA : 2236
Accessing 43 elements in array
[12568.384142] WSS : 64 RSS : 1216 VMA : 2236
Accessing 35 elements in array
[12578.400174] WSS : 63 RSS : 1244 VMA : 2236
Accessing 36 elements in array
[12588.416155] WSS : 64 RSS : 1244 VMA : 2236
[12588.416420] Completed.. Please remove module.

Graph of deterministic program which ran for 3 minutes



Sample OUTPUTs

[9087.546641] Module started..
[9087.546641] Page size : 4096
[9087.546649] Process name with PID 5076 is : leafpad
[9097.568144] WSS : 1764 RSS : 20724 VMA : 175504
[9107.584141] WSS : 793 RSS : 21020 VMA : 175752
[9107.585096] Completed.. Please remove module.
[9119.691414] Module exit

[12336.650860] Module started..
[12336.650860] Page size : 4096
[12336.650872] Process name with PID 5860 is : mtpaint
[12346.656170] WSS : 1790 RSS : 19524 VMA : 29584
[12356.672084] WSS : 1623 RSS : 19880 VMA : 29216
[12366.688177] WSS : 1530 RSS : 20136 VMA : 29236
[12376.704180] WSS : 1486 RSS : 20168 VMA : 29748
[12386.720180] WSS : 1392 RSS : 20516 VMA : 29612
[12386.721146] Completed.. Please remove module.
[12405.569811] Module exit