Practical Guide to AIX

**Memory**  
  
**Memory Leak:** Caused by a program that repeatedly allocates memory without freeing it.  
  
When a process exits, its working storage is freed up immediately and its associated memory frames are put back on the free list.  
However any files the process may have opened can stay in memory.  
  
AIX tries to use the maximum amount of free memory for file caching.  
  
High levels of file system cache usually means that is the way the application runs and likes it (you have to decide if this is expected by understanding the workload) or AIX can't find anything else to do with the memory and so thinks it might as well save disk I/O CPU cycles by caching - this is normal and a good idea.  
  
  
**Some notes regarding memory leak:**  
  
When a process gets busy, process will use malloc() system call (memory allocation) to get more memory, so its memory usage gets bigger.  Memory requests are satisfied by allocating portions from a large pool of memory called the heap. When the process goes idle, it uses free() system call, but that doesn't actually free up the memory from the process. It just releases the memory into the "heap area".  
  
AIX keeps a list of the pages in the heap area about the free memory pages that were used, but not used now. If there are new new malloc() requests, they will be served from heap first. Only if the heap goes to a very small size, only then will be issued new malloc() request to get new memory pages. When heap pages are not used for a long time AIX will page out these to disk.  
  
RSS size is the actual memory occupied by the process in the RAM. (RSS can be active pages or some other pages in the heap). RSS pages will be paged out only if memory is getting short. If there is free mamory, it will not page these out, becaue it maybe useful to have it in the RAM  
  
So, usually it turns out, there is no memory leak at all, just normal memory usage behaviour!!!  
  
------------------------  
  
memory:  
***topas -P***    This does not tell how much of the application is paged out but how much of the application memory is backed by paging space.  
            (things in memory (working segment) should be backed by paging space by the actual size in memory of the process.)    
***svmon -Pt15 | perl -e 'while(<>){print if($.==2||$&&&!$s++);$.=0 if(/^-+$/)}'***        top 15 processes using the most memory  
***ps aux | head -1 ; ps aux | sort -rn +3 | head -20***                                   top memory processes (the above is better)  
***ps -ef | grep -c LOCAL=NO***        shows the number of oracle client connections (each connection takes up memory, so if it is high then...)  
  
paging:  
***svmon -Pg -t 1 |grep Pid ; svmon -Pg -t 10 |grep "N"***                                top 10 processes using the most paging space  
***svmon -P -O sortseg=pgsp***                                                             shows paging space usage of processes  
  
------------------------  
  
***# ps gv | head -n 1; ps gv | egrep -v "RSS" | sort +6b -7 -n -r***  
    PID    TTY STAT  TIME PGIN  SIZE   RSS   LIM  TSIZ   TRS %CPU %MEM COMMAND  
 393428      - A    10:23 2070 54752 54840 32768    69    88  0.0  5.0 /var/opt  
 364774      - A     0:08  579 28888 28940 32768    32    52  0.0  3.0 [cimserve]  
 397542      - A     0:18  472  6468  7212    xx   526   744  0.0  1.0 /usr/sbi  
 344246      - A     0:02   44  7132  7204 32768    50    72  0.0  1.0 /opt/ibm  
  
**RSS**:    The amount of RAM used for the text and data segments per process. PID 393428 is using 54840k. (RSS:resident set size)  
**%MEM:**    The actual amount of the RSS / Total RAM. Watch for processes that consume 40-70 percent of %MEM.  
**TRS:**   The amount of RAM used for the text segment of a process in kilobytes.  
**SIZE:**    The actual amount of paging space (virtual mem. size) allocated for this process (text and data).  
  
How much big is the process in memory? It is the RSS size.  
--------------------------------------  
  
**Checking memory usage with nmon:**  
  
**nmon --> t (top processes) --> 4 (order in process size)**  
  
  PID       %CPU     Size      Res     Res      Res     Char    RAM      Paging         Command  
            Used       KB      Set     Text     Data     I/O     Use   io   other repage  
16580722     0.0   226280   322004   280640    41364        0    5%      0      0      0 oracle  
9371840      0.0   204324   300904   280640    20264        0    5%      0      0      0 oracle  
10551416     0.0   198988   305656   280640    25016        0    5%      0      0      0 oracle  
8650824      0.0   198756   305428   280640    24788        0    5%      0      0      0 oracle  
  
**Size KB:** program on disk size  
**ResSize:** Resident Set Size - how big it is in memory (excluding the pages still in the file system (like code) and some parts on paging disks)  
**ResText:** code pages of the Resident Set  
**ResData:** data and stack pages of the Resident Set             
  
--------------------------------------  
  
**regarding ORACLE:**  
***ps -ef | grep -c LOCAL=NO***  
  
This will show how many client connections we have. Each connections take up some memory, sometimes if there are memory problems too many users are logegd in causing this triouble.  
--------------------------------------  
  
**shared memory segments:**  
  
root@aix2: /root # **ipcs -bm**  
IPC status from /dev/mem as of Sat Sep 17 10:04:28 CDT 2011  
T        ID     KEY        MODE       OWNER    GROUP     SEGSZ  
Shared Memory:  
m   1048576 0x010060f0 --rw-rw-rw-     root   system       980  
m   1048577 0xffffffff D-rw-rw-rw-     root   system       944  
m   4194306 0x78000238 --rw-rw-rw-     root   system  16777216  
m   1048579 0x010060f2 --rw-rw-rw-     root   system       976  
m        12 0x0c6629c9 --rw-r-----     root   system   1663028  
m        13 0x31000002 --rw-rw-rw-     root   system    131164  
m 425721870 0x81fc461c --rw-r-----   oracle oinstall 130027520  
m        15 0x010060fa --rw-rw-rw-     root   system      1010  
m   2097168 0x849c6158 --rw-rw----  **oracle oinstall 18253647872**  
  
  
It will show our memory segments, who owns them and what their size (in bytes). It shows the maximum allocated size, that a memory segment can go to. It does not mean it is allocated, but the exception is Oracle (and DB2).  
Oracle line shows the SGA for Oracle. (This memory is allocated for Oracle. It is 18GB in this case)  
  
--------------------------------------  
  
  
**IBM script for checking what is causing paging space activity:**  
(it will run until po will be 50 then saves processes, svmon and exists)  
  
#!/usr/bin/ksh  
/usr/bin/renice -n -20 -p $$  
while [ true ]  
do  
  vmstat -I 1 1 | tail -1 | awk '{print $9}' | read po  
  if [[ $po -gt 50 ]]  
  then  
      ps -ef > ps.out &  
      svmon -G > svmon.G &  
      exit 0  
  fi  
done  
  
  
**My script for monitoring memory, paging activity:**  
  
#!/usr/bin/ksh  
/usr/bin/renice -n -20 -p $$  
  
while [ true ]; do  
echo `date` "-->" `svmon -G | head -2 | tail -1` "-->" `vmstat -v | grep numperm` >> svmon.out &  
echo `date` "-->" `svmon -G | head -3 | tail -1` >> paging.out &  
echo `vmstat -Iwt 1 1 | tail -1` >> vmstat.out &  
sleep 60  
done