**HSM RECALLS**

When someone gets the I/O error in their SAS log, please have them do the following before submitting a ticket:

1. Check the status of the file using the dsmls command:  dsmls *filename*
   1. Open a unix window.  Go to the directory where the file is located.  Type the command dsmls *filename*
2. If the ‘file state’ is “m”, do a “cat *filename* |more”, then open another Unix window to monitor the status of the file using the dsmls command.  If it works, the file will eventually show a status of “p”.
   1. You can also use the “dsmrecall *filename*” command but the cat command is preferable.
   2. Note:  if the file restore works, issuing the “dsmls filename” command repeatedly one will see the resident file size increasing until eventually the file status is “p”.
3. If the file gets restored properly, go back to the first window where you issued the “cat” command and type ctrl-Z.  This will kill the cat command.
   1. If the file does not get restored after a reasonable amount of time, then submit a help desk ticket.  OR, if the file gets recalled using the above method (and the file state changes to ‘p’) and the SAS job still crashes because of an I/O error, then also open a ticket with the helpdesk.

NOTE:  you can also do the above before submitting a program to try to avoid the I/O error in the first place.  That is, check the file status first, then use the cat or dsmrecall command to restore the file, then submit the SAS job.  This is especially true if the file is very large and/or the file has not been accessed in a long time.

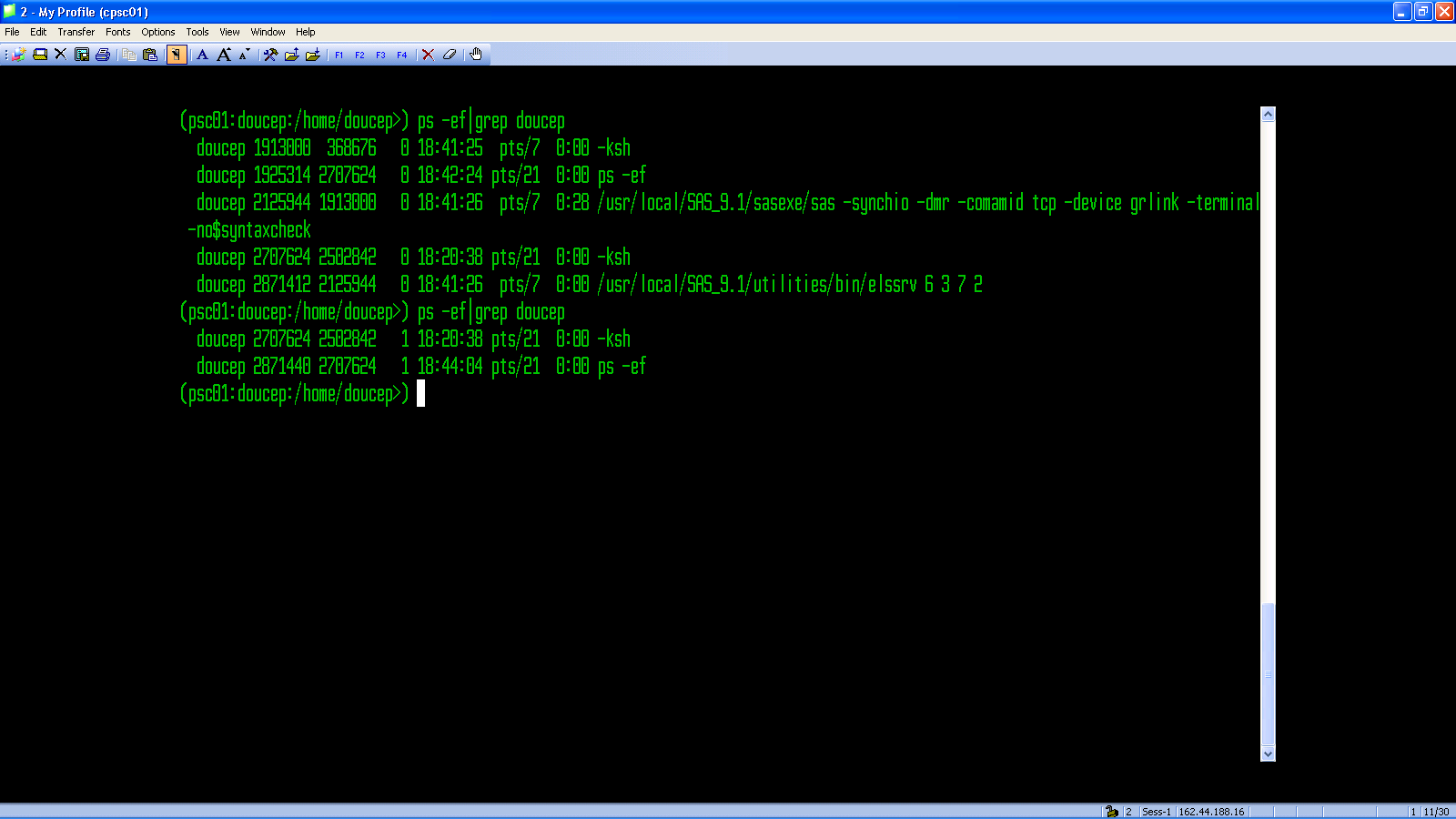
**CHECKING PROCESSES/JOBS RUNNING ON UNIX**

At a Unix command prompt, type ps –ef |grep *username*

This will list all of the active processes for *username*.

The first number after the username is the primary, or parent, process id. The second number is a subordinate, or child, process id. The last column of information is the program name.

When submitting a SAS program using remote submit via SAS on a laptop or blade/powerpc, there will usually be 3 related processes. The primary process is the sasexe process. The process id of the primary process will be the child process for the secondary process. The second process id of the primary (sasexe) process will be the primary process id of a third process. Batch sas jobs usually usually spawn two processes.



Tertiaryprocess

Primary process

Secondary process

Use the “kill” command to delete/remove/kill any old processes. Example: **kill -9 2125944** should kill all 3 related processes.

**CHECKING FOR OLD SAS TEMP/WORK DIRECTORIES**

SAS create two temporary or ‘work’ directories during execution of a program, a util directory and a work directory. If a SAS job ends abnormally, these can be left behind and can consume valuable diskspace. There is a nightly utility which is supposed to clean these up but it doesn’t catch them all.

To find old util/work directories, go to /fs427.1/sastmp and use the ll|more (that’s two lowercase L’s with the vertical bar) and look for your username and the dates of the directories. If you have no jobs running, then there should be no util/work directories. If you find any, delete the contents of the directories using the *rm \*.\** command (making sure you’re in the directory first) , then delete the directory with the *rmdir dirname* command.