Why is /proc/kcore so big?

* Document ID:7004153
* Creation Date:11-Aug-2009
* Modified Date:30-Apr-2012
  + Micro Focus Products:

Open Enterprise Server

* + SUSE Products:

SUSE Linux Enterprise Desktop

SUSE Linux Enterprise Point of Service

SUSE Linux Enterprise Real Time Extension

SUSE Linux Enterprise Server

Environment

SUSE Linux

Situation

When you browse through the file system you see a huge file named kcore. What is this file for?

Resolution

Like all other files below /proc the kcore file is only a virtual file. It contains the RAM the kernel can allocate. Therefore this should not be touched or read. It is nothing to worry about. This file doesn't use actual disk space and only exists virtually.  
  
Note: On 64-bit systems the size of /proc/kcore is even 128TB because that's the absolute limit of what 64-bit systems can allocate.

Additional Information

Some services like NTP have these files below their respective directories as well. For example /var/lib/ntp/proc/kcore  
  
Certain services run in a so-called change root (short chroot) environment. This is done mainly for security reasons. The chroot environment means that the service only sees the files below a certain directory. For NTP this is /var/lib/ntp. NTPs root in this case is /var/lib/ntp. Since it might need access to certain files below /proc/ it mounts /proc below its own root file system again. You can see that when you use the command 'mount' and you will see a line like this:  
  
/proc on /var/lib/ntp/proc type proc (ro)

Find The Execution Time Of A Command Or Process In Linux

To measure the execution time of a command/program, just run.

$ /usr/bin/time -p ls

Or,

$ time ls

Sample output:

dir1 dir2 file1 file2 mcelog

real 0m0.007s

user 0m0.001s

sys 0m0.004s

$ time ls -a

. .bash\_logout dir1 file2 mcelog .sudo\_as\_admin\_successful

.. .bashrc dir2 .gnupg .profile .wget-hsts

.bash\_history .cache file1 .local .stack

real 0m0.008s

user 0m0.001s

sys 0m0.005s

The above commands displays the total execution time of **‘ls’** command. Replace “ls” with any command/process of your choice to find the total execution time.

Here,

1. **real** -refers the total time taken by command/program,
2. **user** – refers the time taken by the program in user mode,
3. **sys** – refers the time taken by the program in kernel mode.

We can also limit the command to run only for a certain time as well. Refer the following guide for more details.