

NAME

getcore-drop core of running process

SYNOPSIS

getcore pid outfile
getcore [-s] offset size outfile

DESCRIPTION

When called as **getcore offset size outfile** this program opens /dev/mem, seeks to offset number of cliques (64 bytes) and reads size number of cliques into its data space. Then it writes the data just read into outfile which it created. If the size is bigger than what fits into its data buffer, then it seats in a loop reading a buffer full at a time and writing a buffer full at a time. Since this process can not be as fast as reading a complete core at one buffer full an attempt is made to make the buffer as big as possible so the most cores can fit into a single buffer full. If the -s option is used the /dev/swap is opened instead, then a seek is made to offset block and size cliques are captured and written into outfile.

When called as **getcore pid outfile** the program will search the symbol table of /unix to find the address of the process table in the operating system. Then it will open /dev/mem and look in the process table of the running operating system for the occurrence of the given pid. If it finds it, it will use the size and offset found in the process table and will capture a core image of the process in memory or swapped according to the information in the process table that indicates where the process is.

BUGS

There is always a possibility although remote that a process moves from the time the information is obtained from the process table to the time that the core copy is made or also as the core copy is made the process can move around from under you. To minimize those possibilities this process is run at nice(-10) but it can still happen. Hence if your core makes absolutely no sense drop another one. If the second one doesn't make better sense I would suspect that you have a bug in your program which is causing the stack or something else to be overwritten. What I am saying is that the chances of it moving on you two times in a row are almost negligible. If you want to know if you got the core you wanted you can look at the end of the core (e.g. tail -150c outfile ^bd). This is the top of the user stack which contains the arguments to main unless your program has overwritten the stack. The first argv in main is by convention the name of the program.

FILES

/unix
/dev/mem
/dev/swap