NAME

protect - protect processes from being killed when swap space is exhausted

SYNOPSIS

protect [**-i**] *command*
protect [**-cdi**] **-g** *pgrp*
protect [**-cdi**] **-p** *pid*

DESCRIPTION

The **protect** command is used to mark processes as protected. The kernel does not kill protected processes when swap space is exhausted. Note that this protected state is not inherited by child processes by default.

The options are:

-c   Remove protection from the specified processes.

-d   Apply the operation to all current children of the specified processes.

-i   Apply the operation to all future children of the specified processes.

-g *pgrp*   Apply the operation to all processes in the specified process group.

-p *pid*   Apply the operation to the specified process.

command Execute *command* as a protected process.

Note that only one of the **-p** or **-g** flags may be specified when adjusting the state of existing processes.

Daemons can be protected on startup using *<name>_oomprotect* option from rc.conf(5).

EXIT STATUS

The **protect** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

Mark the Xorg server as protected:

    pgrep Xorg | xargs protect -p

Protect all ssh sessions and their child processes:

    pgrep sshd | xargs protect -dip

  Remove protection from all current and future processes:

    protect -cdi -p 1

  Using ps(1) to check if the protect flag has been applied to the process:

    ps -O flags,flags2 -p 64430

     PID   F   F2 TT STAT   TIME COMMAND
    64430 10104002 00000001 5 S+   0:00.00 ./main
         ^P   ^PI

  In the above example **P** points at the protected flag and **PI** points at the inheritance flag. The process is
  protected if **P** bit is set to 1. All children of this process will also be protected if **PI** bit is set to 1.

DIAGNOSTICS

  protect: procctl: Operation not permitted The **protect** command does not have the required permissions
  to protect selected processes. There are many reasons why this could be the case, e.g.:

  -   **protect** is not executed by root.

  -   **protect** is executed inside a jail(8), which is not supported at the moment.

SEE ALSO

  ps(1), procctl(2), rc.conf(5)

BUGS

  If you protect a runaway process that allocates all memory the system will deadlock.