NAME

plock – lock process or text in memory

SYNOPSIS

plock (operation)

DESCRIPTION

A process can lock in memory its complete process image or just its text image, and thereby, have it immune to all routine swapping. The caller must be super-user. The argument *operation* specifies:

0 – PUNLOCK – remove locked status on process

1 - PROCLOCK – lock process text & data in memory

2 - TXTLOCK – lock only the text in memory

3 - TUNLOCK – remove locked status on text

Locked processes and texts are shuffled down to the lowest possible address in user swappable memory. *Locks* are not inherited by children across *forks* and *execs* by locked processes are illegal. *Locked* processes can still be swapped under certain circumstances such as those in the following warning.

WARNING:

A great deal of swapping, including the swapping of other locked processes, occurs whenever a process locks its text and/or data; or a locked process grows its data or stack, exits, or is the last locking process to free a locked text. Consequently, processes performing locking should possess long term stability. If the application of locking is to improve real time response, then the careless use of it will do more harm than good.

FILES

/usr/include/sys/lock.h

DIAGNOSTICS

The error bit (c-bit) is set if the proper lock-unlock sequence is not performed in order. i.e. locking the text then locking the process before unlocking the text is illegal.

ASSEMBLER

(syscb = 45.; lock = 3.)
(lock in r1)
sys syscb; operation