;;;

;;; Stub functions for DLX traps.

;;;

;;; Ethan L. Miller, 1999. Released to the public domain.

;;;

;;; The traps such as Open and Close

;;; (note the capital letters) are for use by user programs. The traps

;;; with names such as open and close (lower case) are for use by the OS.

;;;

.text

.align 2

;;; The following are the traps to be used by user-level programs

;;;

.text

.align 2

.proc \_Open

.global \_Open

\_Open:

;;; Note that trap #0x213 actually causes a trap vector in the OS of

;;; 0x1213. The same is true for other user traps. Thus, a user trap

;;; with trap #0x240 will result in a trap vector of 0x1240 in the OS.

trap #0x213

jr r31

nop

.endproc \_Open

.proc \_Close

.global \_Close

\_Close:

trap #0x214

jr r31

nop

.endproc \_Close

.proc \_Read

.global \_Read

\_Read:

trap #0x210

jr r31

nop

.endproc \_Read

.proc \_Write

.global \_Write

\_Write:

trap #0x211

jr r31

nop

.endproc \_Write

.proc \_Lseek

.global \_Lseek

\_Lseek:

trap #0x212

jr r31

nop

.endproc \_Lseek

.proc \_Putchar

.global \_Putchar

\_Putchar:

trap #0x280

jr r31

nop

.endproc \_Putchar

.proc \_Printf

.global \_Printf

\_Printf:

trap #0x201

jr r31

nop

.endproc \_Printf

.proc \_getpid

.global \_getpid

\_getpid:

trap #0x431

jr r31

nop

.endproc \_getpid

.proc \_process\_create

.global \_process\_create

\_process\_create:

trap #0x432

jr r31

nop

.endproc \_process\_create

.proc \_shmget

.global \_shmget

\_shmget:

trap #0x440

jr r31

nop

.endproc \_shmget

.proc \_shmat

.global \_shmat

\_shmat:

trap #0x441

jr r31

nop

.endproc \_shmat

.proc \_sem\_create

.global \_sem\_create

\_sem\_create:

trap #0x450

jr r31

nop

.endproc \_sem\_create

.proc \_sem\_wait

.global \_sem\_wait

\_sem\_wait:

trap #0x451

jr r31

nop

.endproc \_sem\_wait

.proc \_sem\_signal

.global \_sem\_signal

\_sem\_signal:

trap #0x452

jr r31

nop

.endproc \_sem\_signal

.proc \_lock\_create

.global \_lock\_create

\_lock\_create:

trap #0x453

jr r31

nop

.endproc \_lock\_create

.proc \_lock\_acquire

.global \_lock\_acquire

\_lock\_acquire:

trap #0x454

jr r31

nop

.endproc \_lock\_acquire

.proc \_lock\_release

.global \_lock\_release

\_lock\_release:

trap #0x455

jr r31

nop

.endproc \_lock\_release

.proc \_cond\_create

.global \_cond\_create

\_cond\_create:

trap #0x456

jr r31

nop

.endproc \_cond\_create

.proc \_cond\_wait

.global \_cond\_wait

\_cond\_wait:

trap #0x457

jr r31

nop

.endproc \_cond\_wait

.proc \_cond\_signal

.global \_cond\_signal

\_cond\_signal:

trap #0x458

jr r31

nop

.endproc \_cond\_signal

.proc \_cond\_broadcast

.global \_cond\_broadcast

\_cond\_broadcast:

trap #0x459

jr r31

nop

.endproc \_cond\_broadcast

.proc \_mbox\_create

.global \_mbox\_create

\_mbox\_create:

trap #0x460

jr r31

nop

.endproc \_mbox\_create

.proc \_mbox\_open

.global \_mbox\_open

\_mbox\_open:

trap #0x461

jr r31

nop

.endproc \_mbox\_open

.proc \_mbox\_close

.global \_mbox\_close

\_mbox\_close:

trap #0x462

jr r31

nop

.endproc \_mbox\_close

.proc \_mbox\_send

.global \_mbox\_send

\_mbox\_send:

trap #0x463

jr r31

nop

.endproc \_mbox\_send

.proc \_mbox\_recv

.global \_mbox\_recv

\_mbox\_recv:

trap #0x464

jr r31

nop

.endproc \_mbox\_recv

.proc \_sleep

.global \_sleep

\_sleep:

trap #0x465

jr r31

nop

.endproc \_sleep

.proc \_yield

.global \_yield

\_yield:

trap #0x466

jr r31

nop

.endproc \_yield

.proc \_Exit

.global \_Exit

\_Exit:

trap #0x500

jr r31

nop

.endproc \_Exit

.proc \_malloc

.global \_malloc

\_malloc:

trap #0x467

jr r31

.endproc \_malloc

.proc \_mfree

.global \_mfree

\_mfree:

trap #0x468

jr r31

.endproc \_mfree