```
/* -*- linux-c -*- ------------------------------------------------------- *
 *
 *   Copyright (C) 1991, 1992 Linus Torvalds
 *   Copyright 2007-2008 rPath, Inc. - All Rights Reserved
 *   Copyright 2009 Intel Corporation; author H. Peter Anvin
 *
 *   This file is part of the Linux kernel, and is made available under
 *   the terms of the GNU General Public License version 2.
 *
 * ----------------------------------------------------------------------- */

/*
 * Enable A20 gate (return -1 on failure)
 */

#include "boot.h"

#define MAX_8042_LOOPS      100000
#define MAX_8042_FF32

static int empty_8042(void)
{
        u8 status;
        int loops = MAX_8042_LOOPS;
        int ffs   = MAX_8042_FF;

        while (loops--) {
                io_delay();

                status = inb(0x64);
                if (status == 0xff) {
                        /* FF is a plausible, but very unlikely status */
                        if (!--ffs)
                                return -1; /* Assume no KBC present */
                }
                if (status & 1) {
                        /* Read and discard input data */
                        io_delay();
                        (void)inb(0x60);
                } else if (!(status & 2)) {
                        /* Buffers empty, finished! */
                        return 0;
                }
        }

        return -1;
}

/* Returns nonzero if the A20 line is enabled.  The memory address
   used as a test is the int $0x80 vector, which should be safe. */

#define A20_TEST_ADDR       (4*0x80)
#define A20_TEST_SHORT  32
#define A20_TEST_LONG       2097152   /* 2^21 */

static int a20_test(int loops)
{
        int ok = 0;
        int saved, ctr;

        set_fs(0x0000);
        set_gs(0xffff);

        saved = ctr = rdfs32(A20_TEST_ADDR);

        while (loops--) {
                wrfs32(++ctr, A20_TEST_ADDR);
                io_delay();        /* Serialize and make delay constant */
                ok = rdgs32(A20_TEST_ADDR+0x10) ^ ctr;
                if (ok)
                        break;
        }

        wrfs32(saved, A20_TEST_ADDR);
        return ok;
}

/* Quick test to see if A20 is already enabled */
static int a20_test_short(void)
{
        return a20_test(A20_TEST_SHORT);
}

```

```
/* Longer test that actually waits for A20 to come on line; this
   is useful when dealing with the KBC or other slow external circuitry. */
static int a20 test long(void)
{
          return a20_test(A20_TEST_LONG);
}

static void enable a20 bios(void)
{
          struct biosregs ireg;

          initregs(&ireg);
          ireg.ax = 0x2401;
          intcall(0x15, &ireg, NULL);
}

static void enable_a20_kbc(void)
{
          empty 8042();

          outb(0xd1, 0x64);  /* Command write */
          empty_8042();

          outb(0xdf, 0x60);  /* A20 on */
          empty 8042();

          outb(0xff, 0x64);  /* Null command, but UHCI wants it */
          empty_8042();
}

static void enable a20 fast(void)
{
          u8 port_a;

          port_a = inb(0x92);/* Configuration port A */
          port a |=  0x02;   /* Enable A20 */
          port a &= ~0x01;   /* Do not reset machine */
          outb(port_a, 0x92);
}

/*
 * Actual routine to enable A20; return 0 on ok, -1 on failure
 */

#define A20_ENABLE_LOOPS 255 /* Number of times to try */

int enable a20(void)
{
       int loops = A20_ENABLE_LOOPS;
       int kbc_err;

       while (loops--) {
                /* First, check to see if A20 is already enabled
                     (legacy free, etc.) */
                if (a20_test_short())
                          return 0;

                /* Next, try the BIOS (INT 0x15, AX=0x2401) */
                enable a20 bios();
                if (a20_test_short())
                          return 0;

                /* Try enabling A20 through the keyboard controller */
                kbc err = empty 8042();

                if (a20_test_short())
                          return 0; /* BIOS worked, but with delayed reaction */

                if (!kbc_err) {
                          enable a20 kbc();
                          if (a20_test_long())
                                    return 0;
                }

                /* Finally, try enabling the "fast A20 gate" */
                enable a20 fast();
                if (a20_test_long())
                          return 0;
       }

       return -1;
}
```