```
/*
 * File:   alarmdelay.c
 * Author: Michael Kepple
 * Date:   21 Feb 2013
 *
*/
#include "ring.h"

void alarmDelay(int seconds)
{
    sigset_t blockSet;
    struct sigaction action;
    action.sa_handler = timer;
    sigemptyset(&action.sa_mask);
    if (sigaction(SIGALRM, &action, NULL) < 0)
    {
            perror("sigaction");
            exit(EXIT_FAILURE);
    }
    sigfillset(&blockSet);
    sigdelset(&blockSet, SIGALRM);
    alarm(seconds);
    sigsuspend(&blockSet);
    printf("RING!!!");
}

// Empty non-lethal SIGALRM handler
void timer() {}
#
# Makefile for ring
# Author: Michael Kepple
# Date:   21 Feb 2013
#
ring: ring.c ring.h alarmdelay.c msleepdelay.c
/*
 * File:   msleepdelay.c
 * Author: Michael Kepple
 * Date:   21 Feb 2013
 * Note:   usleep() deprecated POSIX.1-2001
*/
#include "ring.h"
#define MIL2NANO 1000000
#define MIL2SECS 1000

void msleepDelay(int milliseconds)
{
    struct timespec time;
    sigset_t set;
    int seconds = milliseconds / MIL2SECS;
    milliseconds %= MIL2SECS;
    sigfillset(&set);
    // Block all signals while sleeping
    sigprocmask(SIG_BLOCK, &set, NULL);
    time.tv_sec = seconds;
    time.tv_nsec = milliseconds * MIL2NANO;
    if (nanosleep(&time, NULL) < 0)
    {
        perror("nanosleep");
        exit(EXIT_FAILURE);
    }
    printf("RING!!!");
}
/*
```

```
 * File:   ring.c
 * Author: Michael Kepple
 * Date:   21 Feb 2013
 * Note:   Input value assumed to be valid.
*/
#include "ring.h"
#define ASCIIOFFSET 48
#define BASE 10

int main(int argc, char* argv[])
{
    int flag = getopt(argc, argv, "s:a:m:");
    int value = ad2b(argv[2]);
    switch (flag)
    {
        case 's':
            sleep(value);
            printf("RING!!!");
            break;
        case 'a':
            alarmDelay(value);
            break;
        case 'm':
            msleepDelay(value);
            break;
        default:
            fprintf(stderr, "Usage: ring -s|-a|-m <delay>\n");
            exit(EXIT_FAILURE);
    }
}

/*
 * Function: ad2b
 * Description: converts ASCII digits into binary/
 * Params: Numeric C string to convert.
 * Returns: Int representation of string param.
 * Modifies: Nothing.
*/
int ad2b(char* value)
{
    int sum = 0, digit = 0;
    char* ascii = value;
    while (*ascii)
    {
        sum *= BASE;
        digit = *ascii++ - ASCIIOFFSET;
        sum += digit;
    }
    return sum;
}
/*
 * File:   ring.h
 * Author: Michael Kepple
 * Date:   21 Feb 2013
*/
#ifndef RING_H
#define RING_H
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <signal.h>
#include <time.h>
typedef int bool;
```

```
    #define true 1
    #define false 0
    #define MAX_INT 2147483647
    int ad2b(char* value);
    void alarmDelay(int seconds);
    void timer();
    #endif
    #
    # Test file for ring program
    #
    $ time ring -s 3
    RING!!!
    real    0m3.002s
    user    0m0.000s
    sys     0m0.000s

    $ time ring -a 4
    RING!!!
    real    0m4.002s
    user    0m0.000s
    sys     0m0.000s

    $ time ring -m 1234
    RING!!!
    real    0m1.237s
    user    0m0.000s
    sys     0m0.000s
```