```
/*
 *
 * aclock.c
 *
 * Author: Michael Kepple
 * Date:   28 Mar 2013
 * Description: Ncurses-based analog clock
 *
*/
#include "aclock.h"

int main()
{
    initscr();
    int cenX = COLS/2;
    int cenY = LINES/2;
    int maxRadius = MIN(LINES/2, COLS/2)-2;
    curs_set(CURSOR_OFF);
    time_t curTime;
    struct tm *localTime;
    int hourPos, minPos, secPos;
    for (;;)
    {
        time(&curTime);
        localTime = localtime(&curTime);
        hourPos = (localTime->tm_hour*HOURVAL)+(localTime->tm_min/MINUTEVAL);
        minPos = localTime->tm_min;
        secPos = localTime->tm_sec;
        clear();
        drawClock(cenX, cenY, maxRadius);
        drawHand(HOUR, hourPos, cenX, cenY, maxRadius);
        drawHand(MINUTE, minPos, cenX, cenY, maxRadius);
        drawHand(SECOND, secPos, cenX, cenY, maxRadius);
        refresh();
        sleep(1);
    }
}

/*
 * Function: drawClock
 * Description: draws the initial clock layout.
 * Params: center x-coordinte, center y-coordinate, maximum radius for screen.
 * Returns: Nothing.
 * Modifies: screen.
*/
void drawClock(int cenX, int cenY, int maxRadius)
{
    char * marks[] = {"3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "1",
    "2"};
    int i, posX, posY;
    for (i=0; i<DEGREES; i+=FREQ)
    {
        double angle = DEG2RAD(i);
        posX = cos(angle)*maxRadius*HALFOFFSET+cenX;
        posY = sin(angle)*maxRadius+cenY;
        if (i % HOURVAL == 0)
        {
            static int place = 0;
            mvprintw(posY, posX, marks[place%HOURS]);
            place++;
        }
        else
            mvwaddch(stdscr, posY, posX, ’-’);
    }
}

/*
 * Function: drawHand
 * Description: draws an individual clock hand based off arguments.
 * Params: int representing which hand caller wants to draw, int representing
 *   the value the hand should point towards, center x-coordinate, center
 *   y-coordinate, maximum radius for screen.
 * Returns: Nothing.
 * Modifies: screen.
*/
void drawHand(int hand, int position, int cenX, int cenY, int maxRadius)
{
    float radians = DEG2RAD(position-QUARTEROFFSET)*FREQ;
    int i, x, y, length;
    char display;
    switch (hand)
    {
        case HOUR:
            length = maxRadius*HOUR_LENGTH;
            display = '@';
            break;
        case MINUTE:
            length = maxRadius*MIN_LENGTH;
            display = '$';
            break;
        case SECOND:
            length = maxRadius;
            display = '*';
            break;
    }
    for (i=1; i<length; i++)
    {
        x = cos(radians)*i*HALFOFFSET+cenX;
        y = sin(radians)*i+cenY;
        mvwaddch(stdscr, y, x, display);
    }
}
/*
 * File:   aclock.h
 * Author: Michael Kepple
 * Date:   28 Mar 2013
 *
*/
#ifndef ACLOCK_H
#define ACLOCK_H
#include <unistd.h>
#include <curses.h>
#include <math.h>
#include <time.h>
#include <stdlib.h>
#define HOUR_LENGTH .5
#define MIN_LENGTH .75
#define CURSOR_OFF 0
#define SECOND 0
#define MINUTE 1
#define HOUR 2
#define HOURS 12
#define FREQ 6
#define HOURVAL 5
#define MINUTEVAL 10
#define DEGREES 360
```

```
#define QUARTEROFFSET 15
#define HALFOFFSET 2
#define PI 3.14159265358979323846
#define MIN(a,b) (((a)<(b))?(a):(b))
#define DEG2RAD(Deg) ((Deg) / 180.0 * PI)
void drawClock(int cenX, int cenY, int maxRadius);
void drawHand(int hand, int value, int cenX, int cenY, int maxRadius);
#endif

#
# Makefile for aclock
# Author: Michael Kepple
# Date:   28 Mar 2013
#
clock: aclock.h aclock.c
	gcc aclock.h aclock.c -lncurses -lm -o aclock
```