```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

/* user-defined parameters */
#define CtrFile "counter.dat"      /* file name to store counter info */
#define HistLen 30                 /* history length, in days         */
#define MesgStr "You are visitor number <b>%ld</b> to this page, number %ld in the past month\n"
                                   /* any string permitted, but must  */
                                   /* have two %ld references, first  */
                                   /* one to total hits, second to    */
                                   /* hits in history period          */

/* user-passable constants */

/* NoUpdate     - will not change counter values when called */
/* PrintTotal   - print total number of hits                 */
/* PrintHistory - print number of hits in history, not total */


/* remainder not customizable */
#define NameLen 64
#define BufLen  (NameLen + (HistLen+2)*sizeof(long))
#define min(a,b) (((a)<(b)) ? (a) : (b))


int main(int argc, char **argv)
{
  struct timeval *tp;
  long fpos, d, user_ctr, user_period, ctr_period, *ctr_data;
  char fname[NameLen+1], buf[BufLen+1];
  int h, i, cgimode, done, matched;
  char *dn;

  dn = getenv( "DOCUMENT_NAME" );

  cgimode = 1;
  gettimeofday( tp, NULL );
  d = tp->tv_sec / 86400;

  user_ctr = user_period = -1;
  if (argc > 1) {
    printf( "Today's date: %d\n\n", d );
    cgimode = 0; dn = argv[1];
    if ((strlen(dn)==1) && (dn[0]=='-')) {
      dn="";
    } else {
      if (argc > 2) user_ctr = atoi( argv[2] );
      if (argc > 3) user_period = atoi( argv[3] );
    }
  }
  else {
    printf("Content-type: text/html%c%c",10,10);
  }

  if (dn != NULL) {
    h = open( CtrFile, O_RDWR | O_CREAT, S_IREAD | S_IWRITE );
    if (h < 0) {
      if (cgimode == 0) printf( "Unable to open counter file.\n" );
      exit(0);
```

```
    }
    done = 0; matched = 0;

    /* have counter file and file name to search */
    ctr_data = (long *) (buf+NameLen);

    while (!done) {
      fpos = lseek( h, 0, SEEK_CUR );
      if (read( h, buf, BufLen ) < BufLen) {
        done = 1;
      }
      else {
        if (strncmp(dn,buf, min(strlen(dn),NameLen-2) ) == 0) {
          /* have a match */
          matched = 1;

#ifndef NoUpdate
          lseek( h, fpos, SEEK_SET );

          /* update info in array */
          if (cgimode == 1) {
            if (d>ctr_data[HistLen+1])
              for (i=0; i<HistLen; i++)
                ctr_data[i] = (i+d-ctr_data[HistLen+1]<HistLen) ? ctr_data[i+d-ctr_data[HistLen+1]] :
0;
            ctr_data[HistLen-1]++;
            ctr_data[HistLen+1] = d;
            ctr_data[HistLen]++;
          }
          else {
            if (user_ctr >= 0) ctr_data[HistLen] = user_ctr;
            if (user_period >= 0)
              for (i=0; i<HistLen; i++)
                ctr_data[i] = user_period / HistLen;
          }
          for (i=0,ctr_period=0; i<HistLen; i++) ctr_period += ctr_data[i];

          write( h, buf, BufLen );
#endif

          /* report to caller */
#ifdef PrintTotal
          printf( "%ld\n", ctr_data[HistLen] );
#else
#ifdef PrintHist
          printf( "%ld\n", ctr_period );
#else
          if (cgimode == 1)
            printf( MesgStr, ctr_data[HistLen], ctr_period );
          /* print info in regular mode */
          if (cgimode == 0)
            printf( "For file %s:\n    Days Since Update: %4ld    Past Month Hits: %5ld     Total
Hits: %8ld\n", buf, d-ctr_data[HistLen+1], ctr_period, ctr_data[HistLen] );
#endif
#endif
        }
      }
    }

    /* if nothing found, write new entry to file */
#ifndef NoUpdate
    if (matched == 0) {
      if (cgimode == 0)
        printf( "No matching entries found.\n" );
```

```
      if (strlen(dn)>0) {
        if (cgimode == 0) printf( "Entry created.\n" );

        lseek( h, 0, SEEK_END );
        strncpy( buf, dn, NameLen-2 ); buf[NameLen-1] = 0;
        for (i=0; i<HistLen; i++) {
          ctr_data[i] = (user_period >= 0) ? user_period/(HistLen-i) : 0;
          user_period -= ctr_data[i];
        }
        ctr_data[HistLen-1] = (user_period >= 0) ? user_period/HistLen : 1;
        ctr_data[HistLen] = (user_ctr >= 0) ? user_ctr : 1;
        ctr_data[HistLen+1] = d;

        write( h, buf, BufLen );
      }
    }
#endif

    close(h);
  }

  exit(0);
}
```