CSI 402 – Systems Programming – Handout 11.2

A Program Example using `struct stat`

Note: This example is taken from pages 55–56 of the text by Haviland et al. The function given below displays information about the file given by the parameter.

```
#include <stdio.h>
#include <sys/stat.h>

#define  SIZE  9

int  filedata (const char *pathname) {

  /* pathname: Path name of the file for which information must be printed. */
  /* The function prints the file size, the user and group id's of the file */
  /* and the file's  permissions in the form "rwxrwxrwx".                    */
  /* Function returns 0 if successful and -1 otherwise.                      */

  /* Permission bits for user, group and others. */
  short  octarray[SIZE] = {0400, 0200, 0100, 0040, 0020, 0010,
                           0004, 0002, 0001};

  char  perms[SIZE+1] = "rwxrwxrwx"; /* To get readable form of permissions. */
  char  description[SIZE+1]; /* Will contain readable form of file permissions. */
  struct stat  statbuf; int j;  /* Temporaries. */

  /* Obtain stat information. */
  if (stat(pathname, &statbuf) == -1) {
     fprintf(stderr, "Could not get stat on file %s\n", pathname);
     return (-1);
  }

  /* Put file permissions in readable form. */
  for (j = 0; j < SIZE; j++) {
    /* Test whether permission is set using bitwise AND. */
    if (statbuf.st_mode & octarray[j])
       description[j] = perms[j];
    else
       description[j] = '-';
  }
  description[SIZE] = '\0';  /* Terminate the string. */

  /* Display information. */
  printf("File: %s\n", pathname); printf("Size: %ld  bytes\n", statbuf.st_size);
  printf("User Id: %d,  Group Id: %d\n", statbuf.st_uid, statbuf.st_gid);
  printf("Permissions: %s\n", description);
  return  (0);
} /* End of filedata */
```