

ΕΡΓΑΣΤΗΡΙΟ 5

Αρχεία και κατάλογοι

Να πληκτρολογηθεί και να εκτελεστεί ο κώδικας των επόμενων παραδειγμάτων που επιδεικνύουν τη χρήση των συναρτήσεων διαχείρισης αρχείων και καταλόγων.

Παράδειγμα 1. Χρήση της `stat` για την εμφάνιση των πληροφοριών του i-node και για αρχείο που διαβιβάζεται ως όρισμα στη γραμμή εντολών ([αρχείο filestat.c](#)).

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

int main(int argc, char *argv[]) {
    struct stat sb;
    if (argc != 2) {
        fprintf(stderr, "Usage: %s <pathname>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    if (stat(argv[1], &sb) == -1) {
        perror("stat");
        exit(EXIT_FAILURE);
    }

    printf("File type:          ");
    switch (sb.st_mode & S_IFMT) {
        case S_IFBLK: printf("block device\n"); break;
        case S_IFCHR: printf("character device\n"); break;
        case S_IFDIR: printf("directory\n"); break;
        case S_IFIFO: printf("FIFO/pipe\n"); break;
        case S_IFLNK: printf("symlink\n"); break;
        case S_IFREG: printf("regular file\n"); break;
        case S_IFSOCK: printf("socket\n"); break;
        default:      printf("unknown?\n"); break;
    }

    printf("I-node number:      %ld\n", (long) sb.st_ino);
    printf("Mode:              %lo (octal)\n", (unsigned long) sb.st_mode);
    printf("Link count:        %ld\n", (long) sb.st_nlink);
    printf("Ownership:         UID=%ld  GID=%ld\n", (long) sb.st_uid, (long) sb.st_gid);
    printf("Preferred I/O block size: %ld bytes\n", (long) sb.st_blksize);
    printf("File size:          %lld bytes\n", (long long) sb.st_size);
    printf("Blocks allocated:    %lld\n", (long long) sb.st_blocks);
    printf("Last status change:   %s", ctime(&sb.st_ctime));
    printf("Last file access:     %s", ctime(&sb.st_atime));
    printf("Last file modification: %s", ctime(&sb.st_mtime));
    exit(EXIT_SUCCESS);
}
```

Παράδειγμα 2. Έλεγχος της ύπαρξης και των δικαιωμάτων πρόσβασης του αρχείου με τη χρήση της access (αρχείο testAccess.c).

```
#include <errno.h>
#include <stdio.h>
#include <unistd.h>

int main (int argc, char* argv[]) {
    char* path = argv[1];
    int rval;

    /* Check file existence. */
    rval = access (path, F_OK);
    if (rval == 0)
        printf ("%s exists\n", path);
    else {
        if (errno == ENOENT)
            printf ("%s does not exist\n", path);
        else if (errno == EACCES)
            printf ("%s is not accessible\n", path);
        return 0; }

    /* Check read access. */
    rval = access (path, R_OK);
    if (rval == 0) printf ("%s is readable\n", path);
    else printf ("%s is not readable (access denied)\n", path);

    /* Check write access. */
    rval = access (path, W_OK);
    if (rval == 0) printf ("%s is writable\n", path);
    else if (errno == EACCES) printf ("%s is not writable (access denied)\n", path);
    else if (errno == EROFS) printf ("%s is not writable (read-only filesystem)\n", path);
    return 0; }
```

Παράδειγμα 3. Αλλαγή δικαιωμάτων πρόσβασης του αρχείου με την εντολή chmod (αρχείο testChmod.c).

```
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <errno.h>

int main (int argc, char * argv[]) {
    int fd;
    struct stat info;
    if (argc!=2) {
        printf ("Usage testChmod pathname\n");
        return (-1); }
    fd = access(argv[1], F_OK);
```

```
If (fd == -1){
    printf("Error Number : %d\n", errno);
    perror("Error Description");
    return (-2); }
else {
    stat(argv[1], &info);
    printf("Original file permissions were: %08o\n", info.st_mode);
    if (chmod(argv[1], S_IRWXU|S_IRWXG) != 0) perror("chmod() error");
    else {
        stat(argv[1], &info);
        printf("After chmod(), file permissions are: %08o\n", info.st_mode); }}
return (0); }
```

Παράδειγμα 4. Αλλαγή κατόχου και ομάδας κατόχου αρχείου με την εντολή chown (αρχείο testChown.c).

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

int main (int argc, char * argv[]) {
    int retVal;
    if (argc!=2) {
        printf ("Usage testChown pathname\n");
        return (-1); }
    retVal = chown (argv[1], (uid_t)3, (gid_t)3);
    if(retVal == -1){
        printf("Error Number : %d\n", errno);
        perror("Error Description");
        return (-2); }
    printf ("Assignment to %s of user SYS and group SYS
                                                    has been performed succesfully\n", argv[1]);

    return (0); }
```

Παράδειγμα 5. Υπολογισμός μεγέθους αρχείου με την lseek (αρχείο fsize1.c).

```
#include <unistd.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/types.h>
#include <errno.h>

int main(int argc, char * argv[]) {
    int fd;
    off_t filelength;
```

```
if (argc !=2) {
    printf ("Usage: flength pathname\n");
    return (-1); }
fd = open(argv[1], O_RDONLY );
if (fd < 0) {
    printf("Error Number : %d\n", errno);
    perror("Error Description"); }
filelength = lseek (fd, 0, SEEK_END);
if (filelength < 0) {
    printf("Error Number : %d\n", errno);
    perror("Error Description"); }
else printf ("Size of file %s is %d bytes\n", argv[1], (int)filelength);
close (fd);
return (0); }
```

Παράδειγμα 6. Υπολογισμός μεγέθους αρχείου με την read (αρχείο [fsize2.c](#)).

```
#include <unistd.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/types.h>
#include <errno.h>

#define BUFSIZE 100

int main (int argc, char *argv[]) {
    int fd, nread, total=0;
    char buf[BUFSIZE];
    if (argc !=2) {
        printf ("Usage: flength pathname\n");
        return (-1); }
    fd = open(argv[1], O_RDONLY);
    while (1) {
        nread = read(fd,buf,BUFSIZE-1);
        if (nread == 0){
            break; }
        /* buf[nread] = '\0';*/
        total += nread; }
    printf("%d bytes total\n",total);
    close(fd);
    return 0; }
```

Παράδειγμα 7. Αντιγραφή αρχείου με τις read και write (αρχείο [myCopy.c](#)).

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <fcntl.h>
#include <errno.h>
```

```
#include <stdlib.h>
#define BUFSIZE 512

int main(int argc, char * argv[]) {
    char buffer[BUFSIZE];
    int src, dest, errno;
    int nread, nwritten, n;
    int totalr=0, totalw=0;
    if (argc!=3) {
        printf ("Usage: flength source destination\n");
        return (-1); }
    src = open(argv[1], O_RDONLY);
    if (src < 0) {
        perror("Source file could not be opened!");
        return (-1); }
    dest = open(argv[2], O_CREAT | O_WRONLY | O_TRUNC, S_IRUSR | S_IWUSR);
    if (dest < 0) {
        perror("Target file could not be opened!");
        return (-2); }
    while ((nread = read(src, buffer, BUFSIZE))>0) {
        nwritten=0;
        do {
            n = write (dest, &buffer[nwritten], nread-nwritten);
            nwritten += n;
        } while (nwritten<nread); }
    close (src); close (dest);
    return (0); }
```

Παράδειγμα 8. Αντιγραφή αρχείου με τις συναρτήσεις της C ([αρχείο myCopy.c](#)).

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <fcntl.h>
#include <errno.h>
#include <stdlib.h>
#define BUFSIZE 512

int main(int argc, char * argv[]) {
    char cTemp;
    FILE * src, * dest;
    if (argc!=3) {
        printf ("Usage: flength source destination\n");
        return (-1); }

    src = fopen(argv[1], "rb");
    if (!src) {
        printf ("Source File could not be opened! Aborting...");
        return (-2); }
```

```
dest = fopen(argv[2], "wb");
if (!src) {
    printf ("Destination file could not be opened! Aborting...");
    return (-3); }

while(fread(&cTemp, 1, 1, src) == 1) {
    fwrite(&cTemp, 1, 1, dest); }

fclose(src);
fclose(dest);
return (0); }
```

Παράδειγμα 9. Δημιουργία και διαγραφή καταλόγου [\(αρχείο dirFun.c\)](#).

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <fcntl.h>
#include <errno.h>
#include <stdlib.h>
#include <dirent.h>

int main(int argc, char * argv[]) {
    long size; int retVal; DIR * dir;
    char * buf, * ptr, arg [20]="ls -l | grep ";
    if (argc!=2) {
        printf ("Usage: dirFun dirname\n");
        return (-1); }
    dir = opendir(argv[1]);
    if (dir) {
        printf ("Directory exists. Aborting...\n");
        closedir(dir);
        return (-2); }
    size = pathconf(".", _PC_PATH_MAX);
    if ((buf = (char *)malloc((size_t)size)) != NULL)
        ptr = getcwd(buf, (size_t)size);
    printf ("Current Working Directory is %s\n", buf);
    printf ("Creating directory %s inside directory %s\n", argv[1], buf);
    free(buf);
    retVal = mkdir (argv[1],0755);
    if (retVal==-1) {
        printf("Error Number : %d\n", errno);
        perror("Error Description");
        return (-2); }
    strcat (arg, argv[1]);
    system (arg);
    printf ("Removing directory %s\n", argv[1]);
    rmdir (argv[1]);
    return (0); }
```

Παράδειγμα 10. Εκτύπωση περιεχομένων καταλόγου (αρχείο myLS.c).

```
#include <stdio.h>
#include <sys/types.h>
#include <errno.h>
#include <dirent.h>

int main(int argc, char * argv[]) {
    DIR * dip;
    struct dirent * dit;
    int files = 0;
    if (argc!=2) {
        printf ("Usage: myLs dirname\n");
        return (-1); }
    if ((dip = opendir (argv[1]))==NULL) {
        printf("Error Number : %d\n", errno);
        perror("Error Description");
        return (-2); }
    printf ("List of contents of directory %s\n", argv[1]);
    while ((dit = readdir(dip))!=NULL) {
        files++;
        printf("%s\n", dit->d_name); }
    printf ("Directory %s contains %d file(s)\n", argv[1], files);
    return (0); }
```