



Εργαστήριο 8

Ασκήσεις: Files and Directories in C

Να υλοποιήσετε σε C και με χρήση κλήσεων συστήματος (system calls) ένα πρόγραμμα, έστω `lsdir`, το οποίο παίρνει ως παράμετρο ένα μονοπάτι (pathname) και εκτυπώνει αναδρομικά (με τη λογική της διάσχισης κατά βάθος, depth-first-search) όλους τους καταλόγους και αρχεία τα οποία συναντά.

Χρησιμοποιείστε τις συναρτήσεις:

```
opendir, readdir, closedir, chdir
```

και τις δομές

```
struct dirent
struct stat
```

Η κλήση του προγράμματος θα γίνεται όπως πιο κάτω, όπου το πρώτο όρισμα μετά το όνομα της εντολής να είναι το μονοπάτι από το οποίο να ξεκινά η διάσχιση/εκτύπωση.

π.χ. `./lsdir ~/public_html/courses/epl421`

Παράρτημα – Χρήσιμες εντολές

```
/* opens a directory stream corresponding to the directory
 * name, and returns a pointer to the directory stream
 */
```

```
DIR *opendir(const char *name);
```

```
/* changes the current working directory of the calling
 * process to the directory specified in path
 */
```

```
int chdir(const char *path);
```

```
/* returns a pointer to a dirent structure representing the
 * next directory entry in the directory stream pointed to by
 * dirp
 */
```

```
struct dirent *readdir(DIR *dirp);
```

```
struct dirent {
    ino_t          d_ino;          /* Inode number */
    off_t          d_off;         /* Not an offset; see below */
    unsigned short d_reclen;     /* Length of this record */
    unsigned char  d_type;       /* Type of file; not supported
                                by all filesystem types */
    char           d_name[256];  /* Null-terminated filename */
};
```



```
/* returns information about a file specified in path.
 * Information is returned within struct stat
 */
int lstat(const char *path, struct stat *buf);

struct stat {
    dev_t      st_dev;      /* ID of device containing file */
    ino_t      st_ino;     /* inode number */
    mode_t     st_mode;    /* File type and mode: for more
                           information please see here */
    nlink_t    st_nlink;   /* number of hard links */
    uid_t      st_uid;     /* user ID of owner */
    gid_t      st_gid;     /* group ID of owner */
    dev_t      st_rdev;    /* device ID (if special file) */
    off_t      st_size;    /* total size, in bytes */
    blksize_t  st_blksize; /* blocksize for file system I/O */
    blkcnt_t   st_blocks;  /* number of 512B blocks allocated */
    time_t     st_atime;   /* time of last access */
    time_t     st_mtime;   /* time of last modification */
    time_t     st_ctime;   /* time of last status change */
};
```