



# **B1 - Unix System Programming**

---

B-PSU-100

## **Bootstrap**

---

my\_ls



2.1



# Bootstrap

binary name: info\_file

language: C

compilation: via Makefile, including re, clean and fclean rules



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

You must develop a program that takes a file name as parameter and displays the following information:

- File name
- File type
- Inode
- Number of hard links
- File size
- Allocated space on hard drive
- Minor (if file is "device")
- Major (if file is "device")
- Owner's UID
- Owner's GID

## AUTHORIZED FUNCTIONS

The only system calls allowed are the following ones:

- opendir, readdir, closedir, stat, lstat, write
- getpwuid, getgrgid, time, ctime, readlink,
- malloc, free, perror, strerror, exit.



Read the manual first



## EXAMPLES

---

```
Terminal
~/B-PSU-100> info_file /bin/ls
Name: ls
Type: f
Inode: 424242
Hard Link: 2
Size: 110080
Allocated space: 112000 0
Minor: N/A
Major: N/A
UID: 0
GID: 0
```

```
Terminal
~/B-PSU-100> info_file /dev/tty
Name: tty
Type: c
Inode: 17
Hard Link: 1
Size: 0
Allocated space: 0
Minor: 0
Major: 5
UID: 0
GID: 5
```



```
Terminal
~/B-PSU-100> ./info_file /dev/toto
File not found
~/B-PSU-100> echo $?
84
```

```
Terminal
~/B-PSU-100> ./info_file path/to/a_symbolic_link
Name: a_symbolic_link
Type: l
Inode: 22546188
Hard Link: 1
Size: 7
Allocated space: 4096
Minor: N/A
Major: N/A
UID: 1000
GID: 1000
```