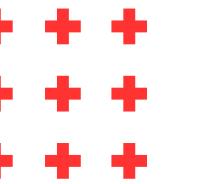


LINUX FILESYSTEM

"Code Your Embedded Future"

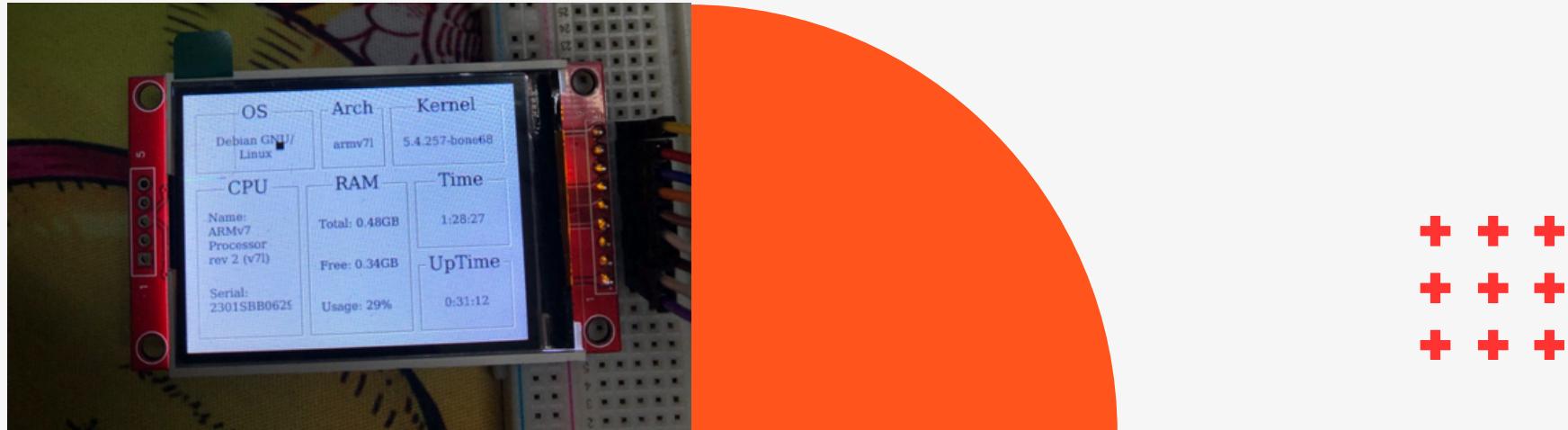


 <https://devlinux.vn>

 Cộng đồng lập trình Nhúng & Vi Mạch

 **SUBSCRIBE** @devlinux097

AGENDA



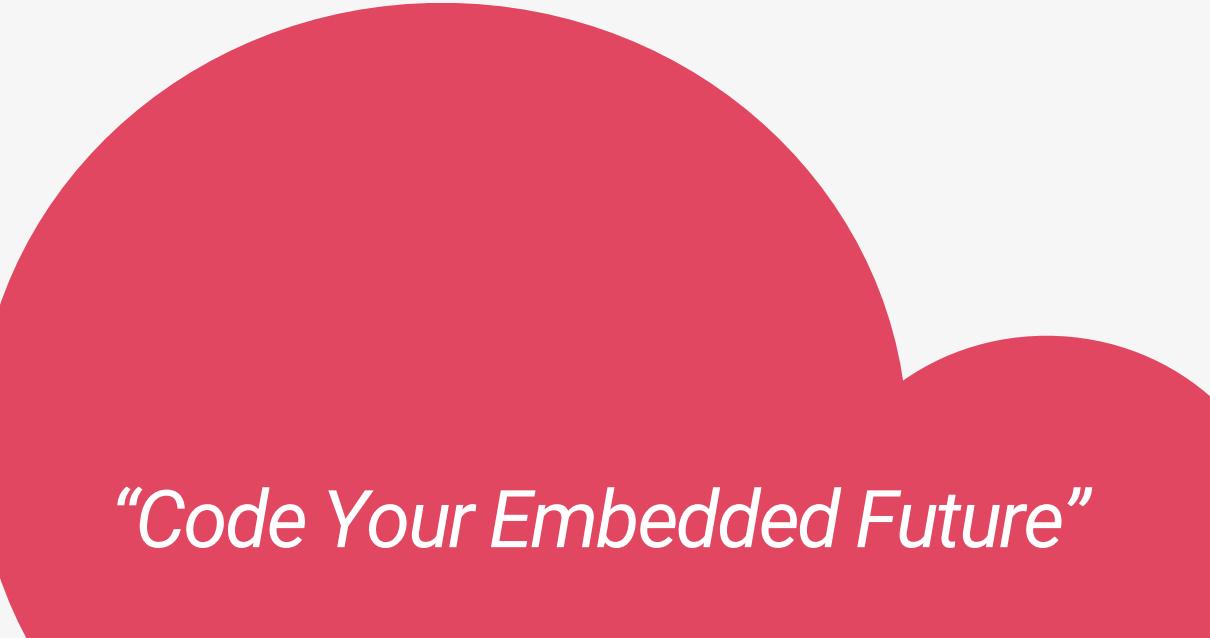
-  I. Introduction
-  II. Operations on File
-  III. File Management
-  IV. File locking

INTRODUCTION

 <https://devlinux.vn>

 Cộng đồng lập trình Nhúng & Vi Mạch

 **SUBSCRIBE** @devlinux097



“Code Your Embedded Future”

Overview

The Philosophy of Linux: Everything in Linux is a file.

Types of files in Linux:

- **Regular file:** Standard files such as text files or executable files.
- **Directory file:** A file that contains a list of other files (i.e., a folder).
- **Character device file:** A special file that represents character-based devices.
- **Block device file:** A special file that represents block-based devices.
- **Link file:** A file that represents (or points to) another file.
- **Socket file:** A file that represents a socket.
- **Pipe file:** A file that represents a pipe (used for inter-process communication).

phong@slayer:~\$ ls -l							
total 488							
drwxr-xr-x	3	phong	phong	4096	Th03 22	2021	Desktop
drwxr-xr-x	2	phong	phong	4096	Th03 17	2021	Documents
drwxr-xr-x	4	phong	phong	4096	Th09 20	18:38	Downloads
-rw-r--r--	1	phong	phong	8980	Th03 17	2021	examples.desktop
-rw-r--r--	1	root	root	70	Th04 24	2021	minicom.log
drwxr-xr-x	2	phong	phong	4096	Th03 17	2021	Music
drwxr-xr-x	2	phong	phong	4096	Th03 17	2021	Pictures
drwxr-xr-x	2	phong	phong	4096	Th03 17	2021	Public
drwxrwxr-x	3	phong	phong	4096	Th03 29	2021	SimplicityStudio
drwxr-xr-x	2	phong	phong	4096	Th03 17	2021	Templates
-rw-rw-r--	1	phong	phong	441592	Th10 16	21:25	u-boot.bin
drwxr-xr-x	2	phong	phong	4096	Th03 17	2021	Videos
drwxrwxr-x	9	phong	phong	4096	Th11 16	23:41	Working_Space

Tên File
Thời điểm edit file lần cuối
Kích thước file (bytes)
Tên group
Tên User
Số hardlink của file
Loại file, quyền của file

<https://devlinux.vn>

Cộng đồng lập trình Nhúng & Vị Mạch

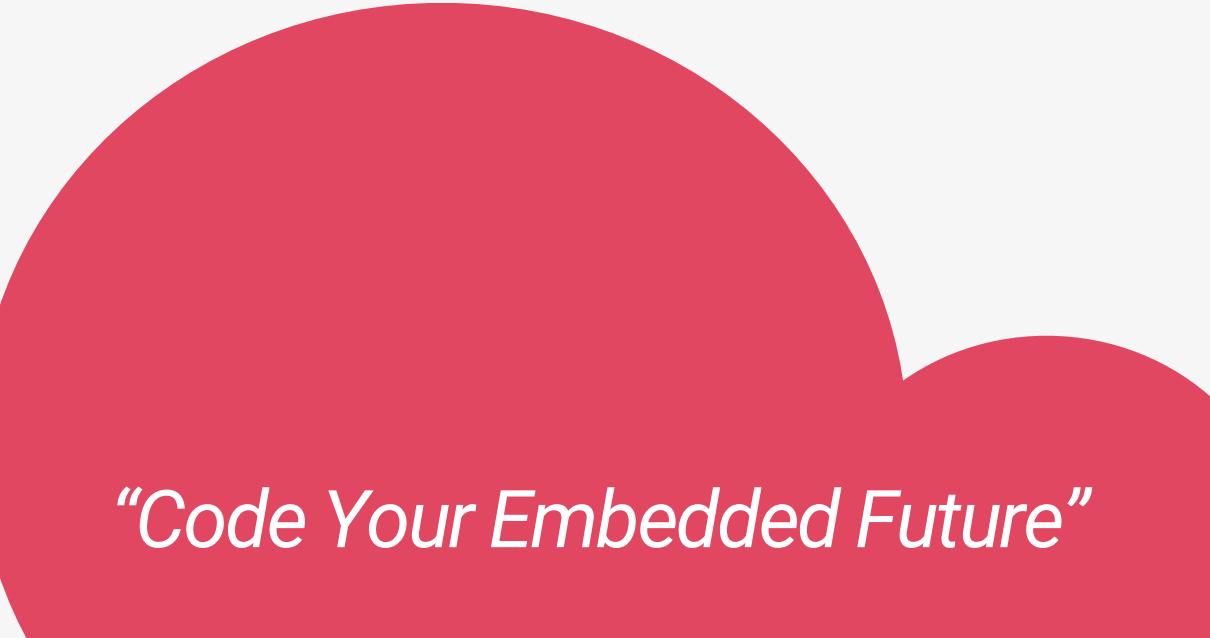
SUBSCRIBE @devlinux097

OPERATIONS ON FILE

 <https://devlinux.vn>

 Cộng đồng lập trình Nhúng & Vි Mạch

 **SUBSCRIBE** @devlinux097



“Code Your Embedded Future”

System Call

The kernel provides system calls to perform reading, writing, and other necessary operations on files.

- open()
- read()
- write()
- lseek()
- close()

```
int open(const char *pathname, int flags, mode_t mode);
```

```
ssize_t read(int fd, void *buffer, size_t count);
```

```
ssize_t write(int fd, void *buffer, size_t count);
```

```
off_t lseek(int fd, off_t offset, int whence);
```

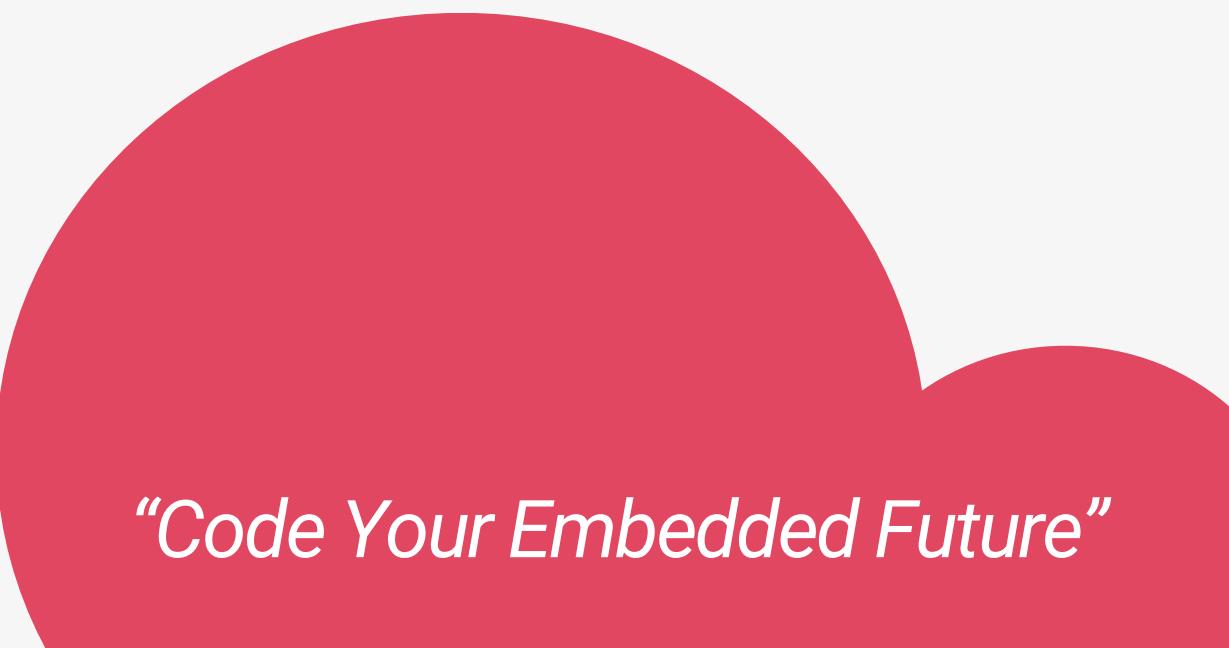
```
int close(int fd);
```

FILE MANAGEMENT

 <https://devlinux.vn>

 Cộng đồng lập trình Nhúng & Vi Mạch

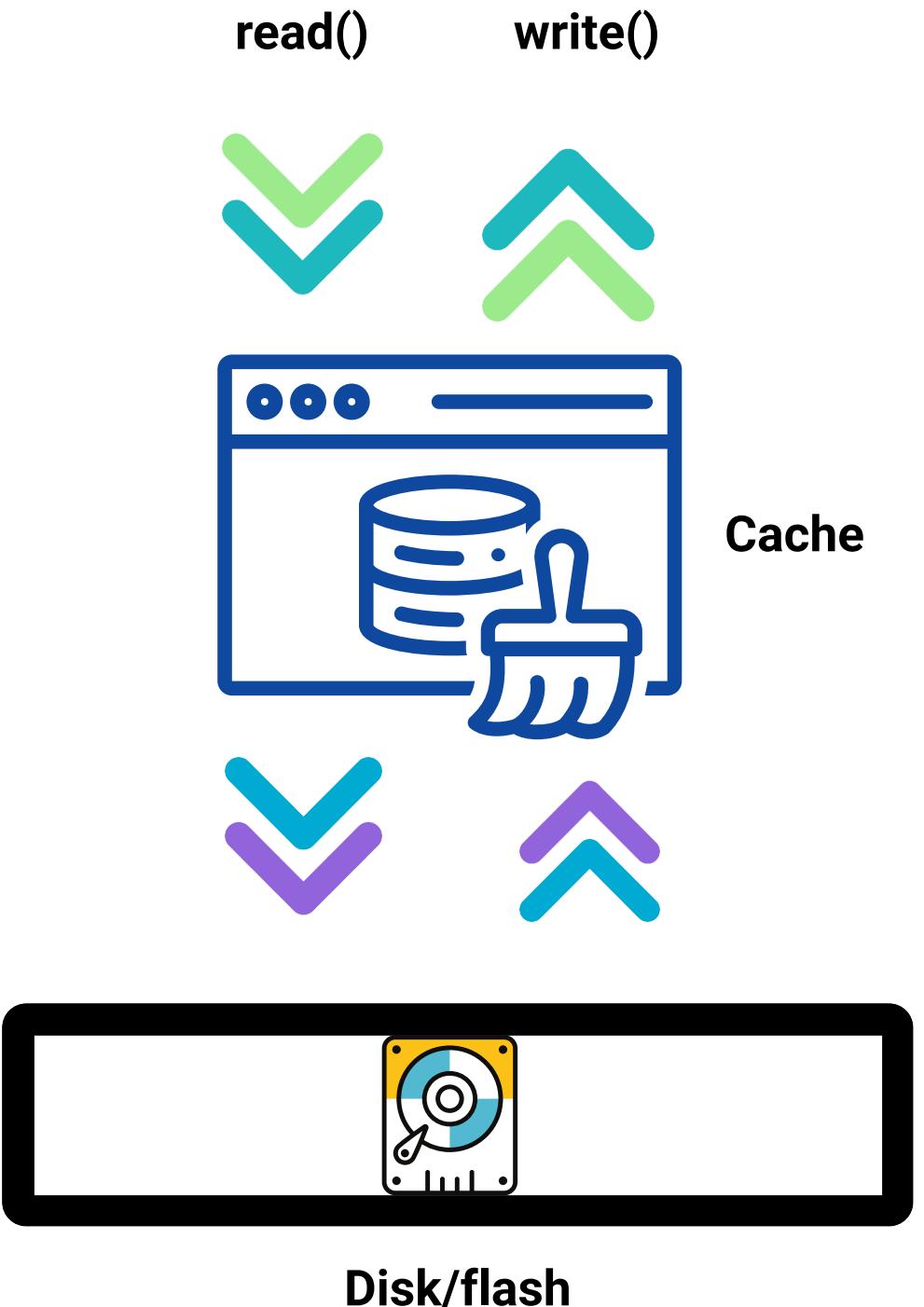
 **SUBSCRIBE** @devlinux097



“Code Your Embedded Future”

Page Cache

1. The kernel identifies the page to be read.
2. The kernel reads from the page cache.
3. If the page is already in the cache, the data will be read directly.
4. If the page is not in the cache, the system will read it from physical memory into the cache, then copy it to user space.



1. The kernel writes the page contents to the page cache.
2. The page cache will be flushed to physical memory periodically or when commands like sync() or fsync() are called.

I-node Table

I-node number	Type	Permissions	Size	Hard Link Count	Locks	Data Block Pointers
142	Regular	-rw-r--r-	4096 bytes (4 KB)	1	Locked (by Process ID 1234)	234, 125, 253
-	-	-	-	-	-	-
1145	Regular	-rw-rw-rw	1024 bytes (1KB)	3	Unlocked	5356, 9839
-	-	-	-	-	-	-
5632	5632	-rw---r-	2048 bytes (2KB)	1	Unlocked	821, 2929

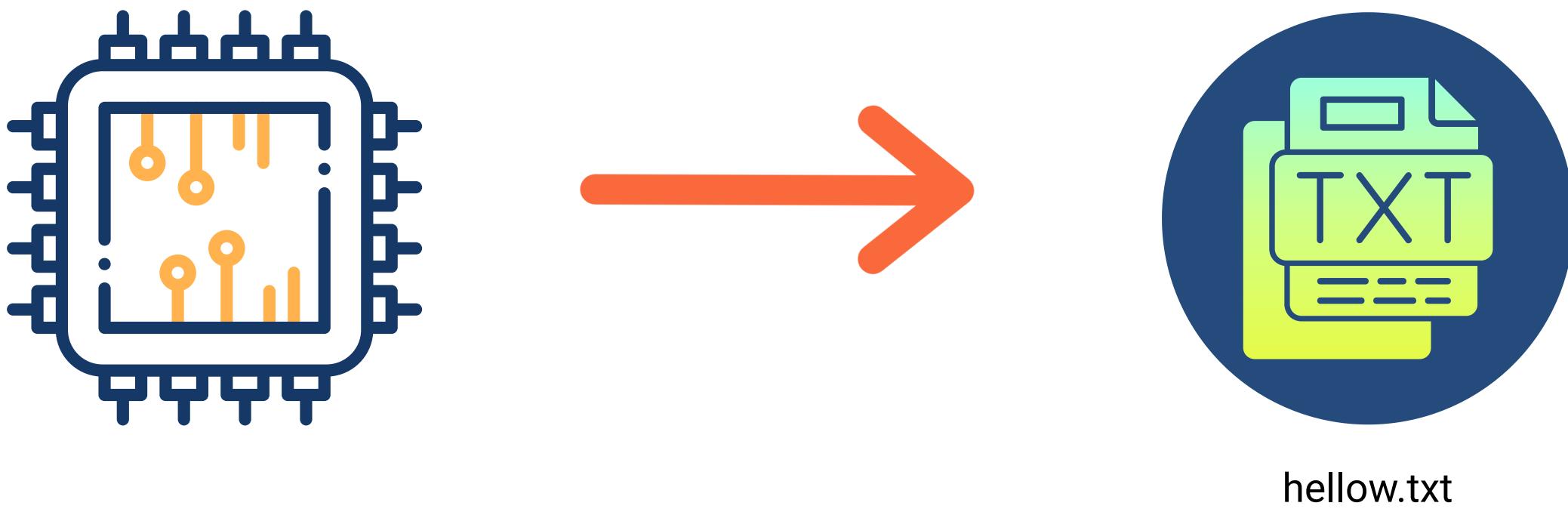
Open File Table

File Offset	Status Flags	I-node pointer
0	O_RDWR O_CREATE	1145
-	-	-
16	O_RDONLY	5632
-	-	-
100	O_RDWR	142

File Descriptor Table

File Descriptor	Fd Flag	File Ptr
fd0	O_RDWR	100
fd1	O_RDONLY	16
fd2	O_RDWR O_CREATE	0
-	-	-

File Opening Process



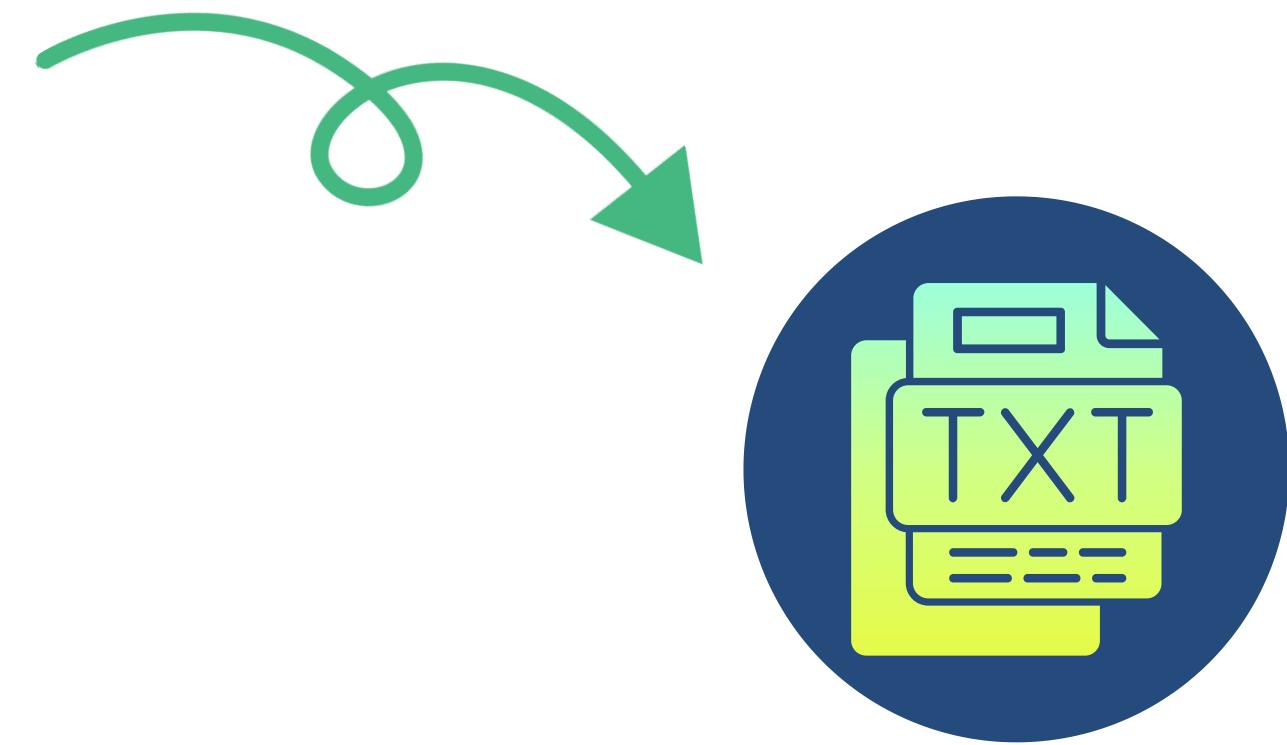
<https://devlinux.vn>

Cộng đồng lập trình Nhúng & Vි Mạch

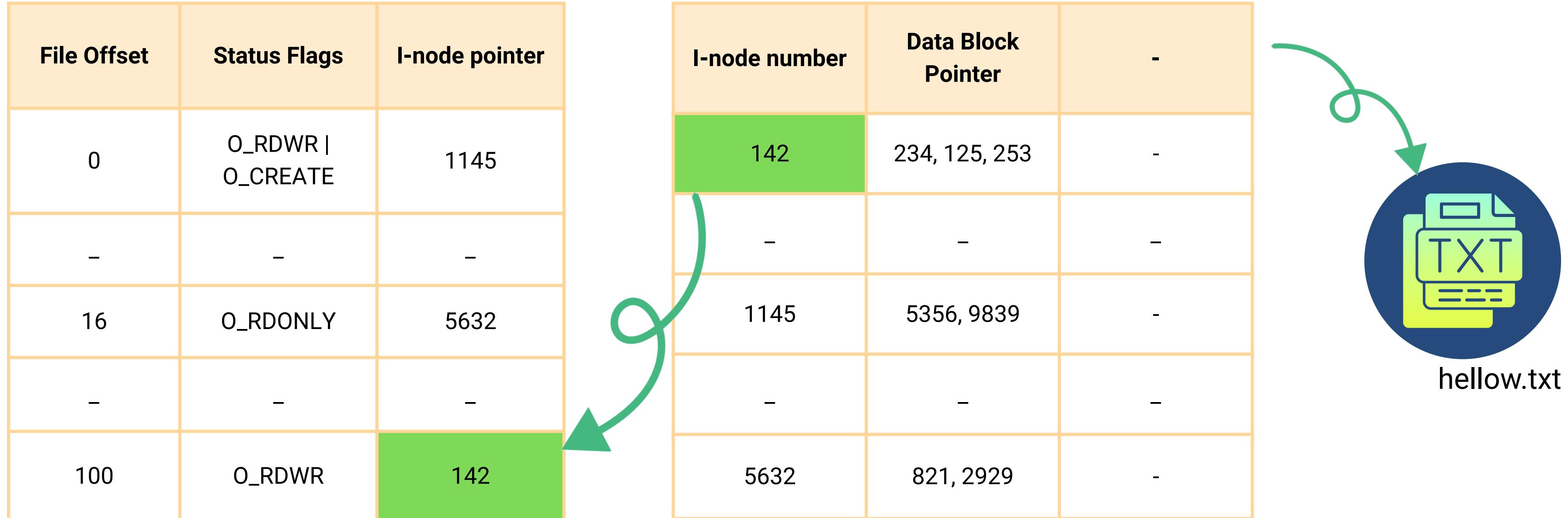
 SUBSCRIBE @devlinux097

File Opening Process

I-node number	Data Block Pointer	-
142	234, 125, 253	-
-	-	-
1145	5356, 9839	-
-	-	-
5632	821, 2929	-



File Opening Process



File Opening Process

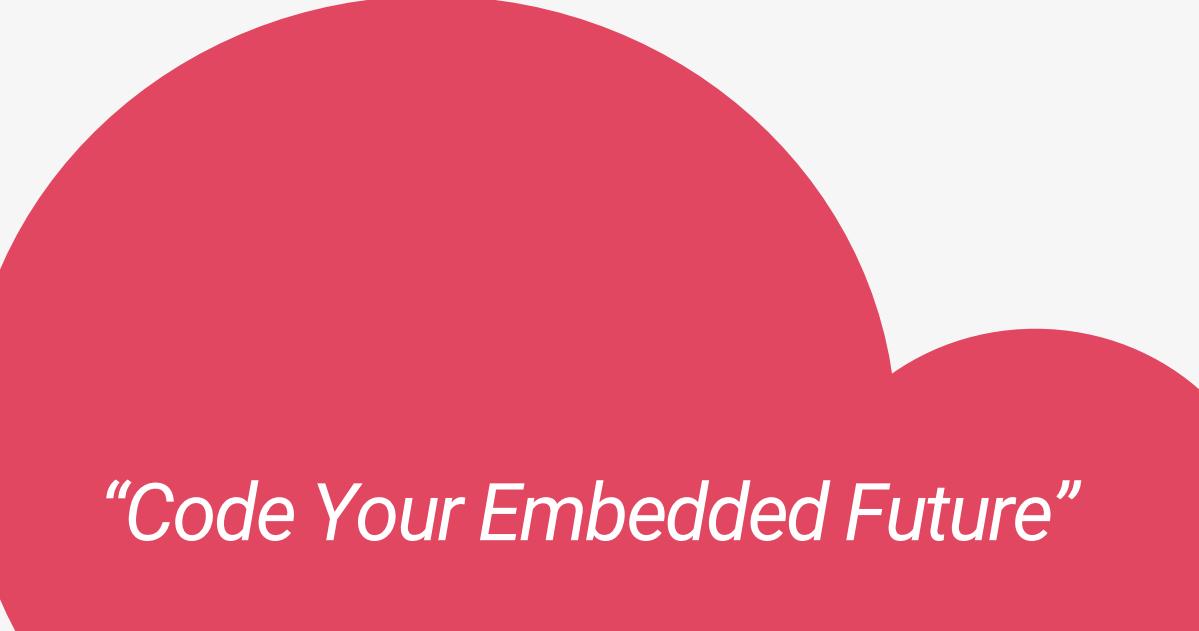


FILE LOCKING

 <https://devlinux.vn>

 Cộng đồng lập trình Nhúng & Vි Mạch

 **SUBSCRIBE** @devlinux097



“Code Your Embedded Future”

File Locking

Feature	flock()	fcntl()
Usage Scope	Applies to the entire file, simple and easy to use	Can lock the entire file or specific byte ranges (more fine-grained control)
Lock Types	Shared (read) lock or Exclusive (write) lock	Shared (read) lock, Exclusive (write) lock, Advisory lock, Mandatory lock
Lock Granularity	Only supports whole-file locking	Supports byte-range locking
Concurrency Handling	Can be used between multiple processes, but simpler and less powerful than fcntl()	Supports locking between and within processes, better control over concurrent access

int flock(fd, operation)

flock() uses the file descriptor information to set the lock state in the i-node table.

Parameters:

- **fd**: The file descriptor of the file to be locked.
- **operation**: The lock operation to be applied.
 - **LOCK_SH** (Shared Lock)
 - Allows multiple processes to read the file simultaneously.
 - Used when processes only need to read the file without modifying its contents.
 - **LOCK_EX** (Exclusive Lock)
 - Ensures that only one process can write to the file; prevents other processes from reading or writing.
 - Used when a process needs to write to the file and must ensure no other process reads/writes during that time.

int fcntl(fd, cmd, &flockstr)

fcntl() is more flexible than flock(). The fcntl() function allows locking specific parts of a file (even down to individual bytes). Lock information stored in the i-node table includes: process ID, lock status, and the locked region.

Parameters:

- **fd**: The file descriptor of the file to be locked.
- **cmd**: The action to be performed.
 - **F_SETLK**: Set or release a lock
 - **F_GETLK**: Retrieve lock information
- **flockstr**: A structure containing lock information (including lock status, the region to lock, and the process holding the lock)



Thanks for Your Attention!

Liên hệ với chúng tôi

 <https://devlinux.vn>

 Cộng đồng lập trình Nhúng & Vi Mạch

 **SUBSCRIBE** @devlinux097

