EMBEDDED SYSTEMS COACHING BLOG

This blog will contains the embedded systems tutorilas and interview questions and examples and so on.....

Blog Archive

- 2013 (78)
  - April (3)
  - July (3)
  - October (21)
  - November (38)
    - c and c++ Programs Compilation Process
    - Modules Build Log
    - Deleted module Log
    - Interrupts Handling Interview Question
    - What is use struct stat
    - Camera notes
    - VIDIOC_S_FMT error 22, Invalid argument
    - error: curl/curl.h: No such file or directory
    - Interrupts in Linux kernel
    - Liner search
    - Binary Search
    - drivers/usb/gadget/r8a66597-udc.c:1585: error: imp...
    - Storage classes
    - How to find kernel source version ?
    - Understanding ARM Processors / CPU in your Smartph...
    - More Questions on Linux kernel Modules
    - Openssl ppt
    - Character device drivers development or tutorial
    - Linux device drivers interview questions and

Wednesday, 13 November 2013

Linux Device driver interview questions and answers

What is mknod and it's usage ?

mknod is a command which used create the device file (or) node in Linux file system.

In unix or linux we will represent everything as a file .

syntax: **mknod** Name { **b** | **c** } Major Minor

Name : name of the device file

{ **b** | **c** } : type of device (ex; char or block device)

Major : Major number of the device file

Minor : Minor number of the device file

ex : $ mknod /dev/rama c 12 5

MKDEV(int major, int minor);

In how many ways we can allocate device number ?

In 2 ways we can allocate device numbers
1)statically
2)dynamically

How can we allocate device number statically ?

Ans :register_chrdev_region() function will statically allocate device numbers. which is declared in <linux/fs.h>
int register_chrdev_region(dev_t first, unsigned int count, char *name);
Return values : In case of success "0" will return , In case of failure "-1 or negative value " will return
Here

1. first is the beginning device number of the range you would like to allocate. The minor number portion of first is often 0.
2. count is the total number of contiguous device numbers you are requesting.
3. name is the name of the device that should be associated with this number range. it will appear in /proc/devices and sysfs.

How can we allocate device number dynamically ?

alloc_chrdev_region()will dynamically allocate device numbers.
int alloc_chrdev_region(dev_t *dev, unsigned int firstminor,
unsigned int count, char *name);
Here

1. dev is an output-only parameter that will, on successful completion, hold the first number in your allocated range.
2. firstminor should be the requested first minor number to use; it is usually 0

Pages

Home

Homepage

handb...

Picture resolutions site

Linux Device driver interview questions and answer...

Embedded C Interview Questions

implicit declaration of function copy_from_user

Sample USB driver or Pen drive code

How to set adb path in ubuntu

Openmax AL ppt

How to install java on UBuntu

How to set ndk path in ubuntu

How to Put JNI logs

Basic Linux commands required for device drivers

My interview Experienc with HCL

How do I install Oracle JDK 6?

I2C Protocol and Devicr Driver Interview Questions...

USB device driver and Protocol interview Questions...

V4L2 Interview Questions and Answers

Camera (or) Sensor device driver Interview Questio...

Interview Question asked in MNC Companies HCL,Robe...

Mutex Locks interview questions

► December (13)

► 2014 (2)

3. count is the total number of contiguous device numbers you are requesting.
4. name is the name of the device that should be associated with this number range. it will appear in /proc/devices and sysfs.

How can we Free Device Numbers ?

void unregister_chrdev_region(dev_t first, unsigned int count);

What is Major number and it's usage ?

It's an integer number mainly used to provide the association between the device driver and device file . this number is used by kernel .

(or)

The major number tells you which driver handles which device file.

Can we have same major number for more than one device file ?

yes . we can have .

What is minor number and it's usage ?

The minor number is used only by the driver itself to differentiate which device it's operating on, just in case the driver handles more than one device.

(or)

one driver can control more than one device .minor will be used to distinguish the one device from other devices .

What is range of major and minor numbers?

0-255

What is use of dev_t type ?

This is used to hold device numbers—both the major and minor parts.

How to retrieve major and minor number from dev_t type ?

To obtain the major or minor number of a dev_t, use:

MAJOR(dev_t dev); // to obtain major number

MINOR(dev_t dev); // to obtain minor number

int major=MAJOR(dev_t dev);

int minor =MINOR(dev_t dev);

How can i use my own major and minor number for a device file ?

if you have the major and minor numbers and need to turn them into a dev_t, use:

register_chrdev_region works well if you know ahead of time exactly which device numbers you want. Often, however, you will not know which major numbers your device will use; there is a constant effort within the Linux kernel development community to move over to the use of dynamically-allocated device numbers.

How to see statically assigned major numbers ?

Some major device numbers are statically assigned to the most common devices. A list of those devices can be found in Documentation/devices.txt within the kernel source tree.

Note : for new drivers, we strongly suggest that you use dynamic allocation to obtain your major device number, rather than choosing a number randomly from the ones that are currently free. In other words, your drivers

should almost certainly be using alloc_chrdev_region rather thanregister_chrdev_region.

What is the disadvantage of dynamic device number assignment ?

The disadvantage of dynamic assignment is that you can't create the device nodes in advance, because the major number assigned to your module will vary.

Where can we write allocation and freeing of device number's code ?

allocation : init function of a module
freeing : cleanup function of a module

Posted by Ramesh rama at 22:53

No comments:

Post a Comment

Links to this post

Create a Link

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

About Me

Ramesh rama

Follow 29

View my complete profile