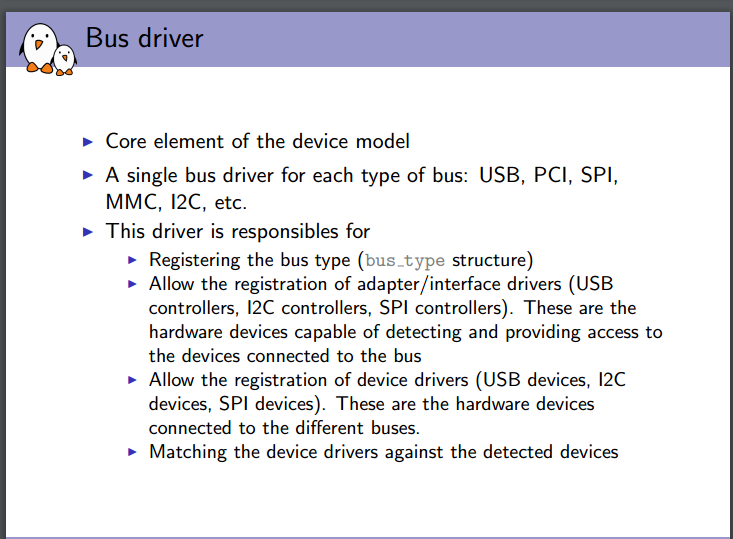
không cần interrupt vì chỉ có 1 thiết bị kết nối với PC trong 1 thời điểm

ioctl(): set speed of serial port (baundrate)



những struct có sẵn trong kernel

* usb\_device
* usb\_interface
* usb\_endpoint\_descriptor
* usb\_cdc\_line\_coding
* us\_cdc\_parsed\_header
* urb
* work\_struct
* async\_icount
* tty\_driver
* tty\_port
* tty\_struct
* file
* serial\_struct
* serial\_icounter\_struct

~~throttled, unthrottled ???~~

1. Look up an ACM structure by minor. If found and not disconected,

Increment

1. Find available minor number, if found, associate it with acm
2. Release the minor number associated with acm
3. ACM control messages ( **functions**)
4. Write buffer management
5. Finish write. Caller must hold **acm->write\_lock**
6. poke write. The caller is responsible for locking
7. data interface wrote those outgoing bytes
8. TTY handlers (tty\_**install, tty\_open, port\_activate, port\_destruct,port\_shutdown, tty\_cleanup, tty\_hangup, tty\_close, tty\_write,)**

Get\_serial\_info

Set\_serial\_info

Wait\_serial\_change

Acm\_tty\_**ioctl**

**Acm\_tty\_set\_termios**

1. USB probe and disconnect routines

Write\_buffers\_free

Read\_buffer\_free

Wite\_buffers\_alloc

Acm\_probe

….

Acm\_disconnect

1. USB structure

Nokia blabla

1. TTY driver structures

.install

.open

.close

….

1. Init / exit

\_\_init acm\_init

\_\_exit acm\_exit

**Module\_init(acm\_init)**

**Module\_exit(acm\_exit)**