

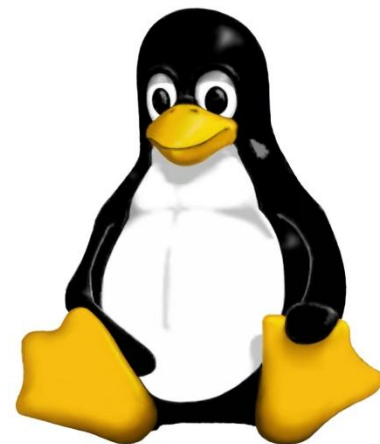


Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχ. και Μηχανικών Υπολογιστών  
Εργαστήριο Υπολογιστικών Συστημάτων

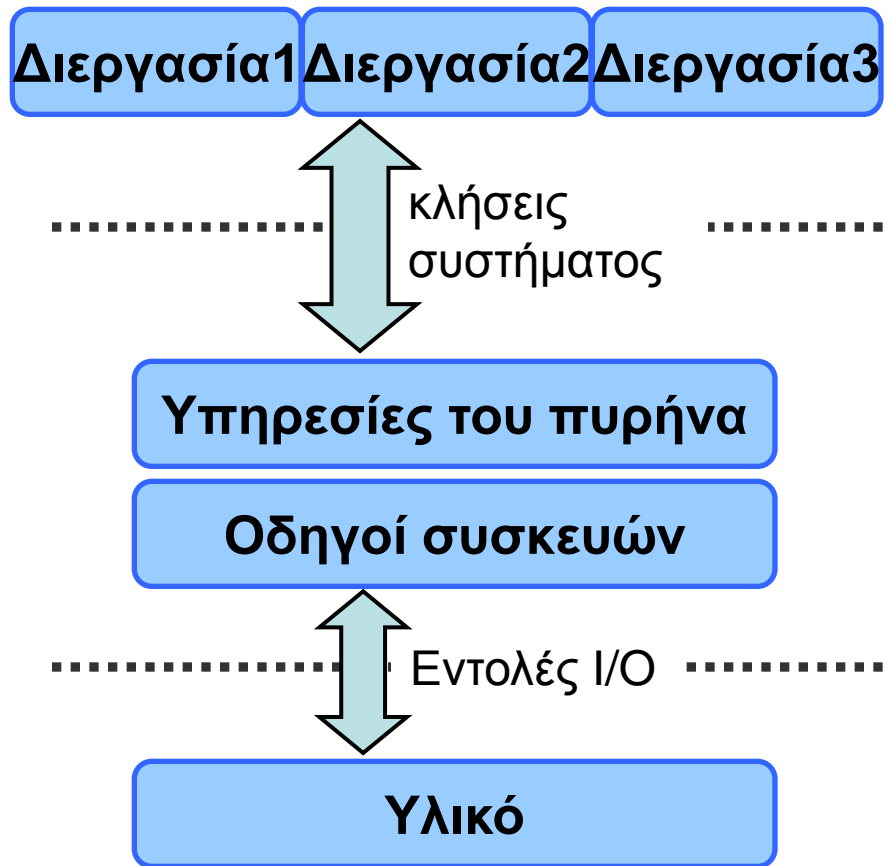
# Οδηγός Ασύρματου Δικτύου Αισθητήρων στο Λειτουργικό Σύστημα Linux

Εργαστήριο Λειτουργικών Συστημάτων  
2η εργαστηριακή άσκηση

Οκτώβριος 2021



# Οργάνωση ενός σύγχρονου Λ.Σ.



## ◆ Χώρος χρήστη

- ➔ απομονωμένη εκτέλεση
- ➔ χρήση κλήσεων συστήματος

## ◆ Χώρος πυρήνα

- ➔ απεριόριστη πρόσβαση στη μνήμη
- ➔ ελεύθερη επικοινωνία με το υλικό

# Ο πυρήνας του Linux



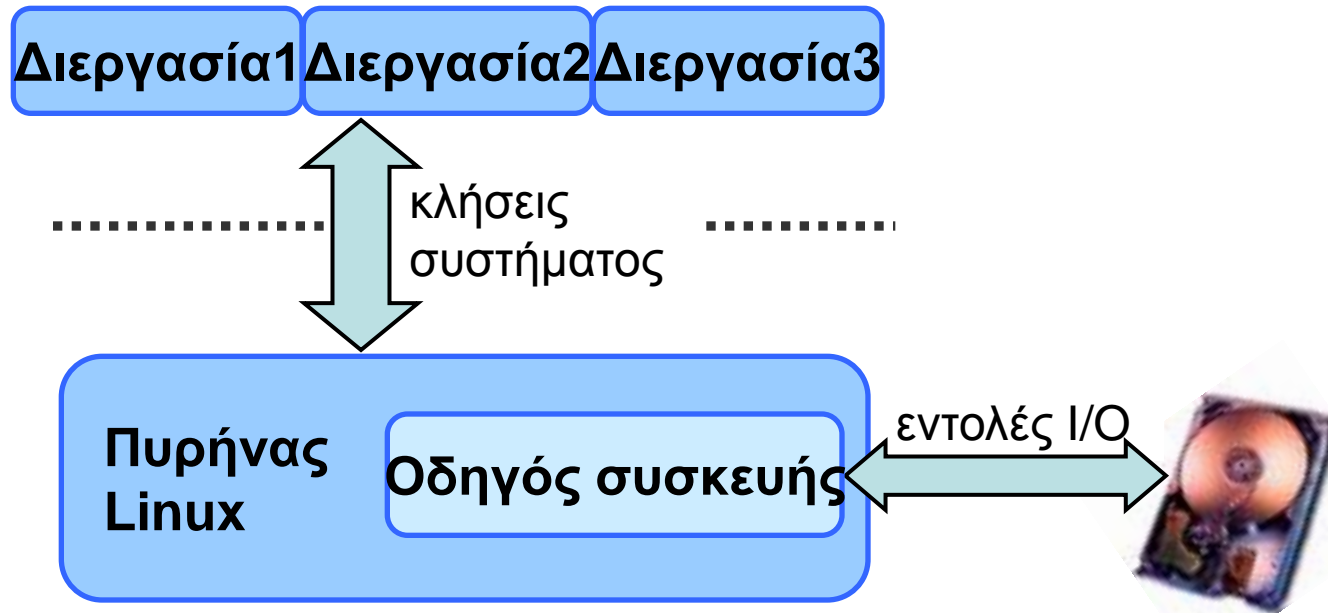
- ◆ Διαχείριση διεργασιών
  - ➔ χρονοδρομολόγηση σε CPUs
- ◆ Διαχείριση μνήμης
- ◆ Διαδικτύωση (TCP/IP)
- ◆ Υποστήριξη συστημάτων αρχείων
- ◆ Διαχείριση συσκευών (Device Drivers)

# Kernel modules



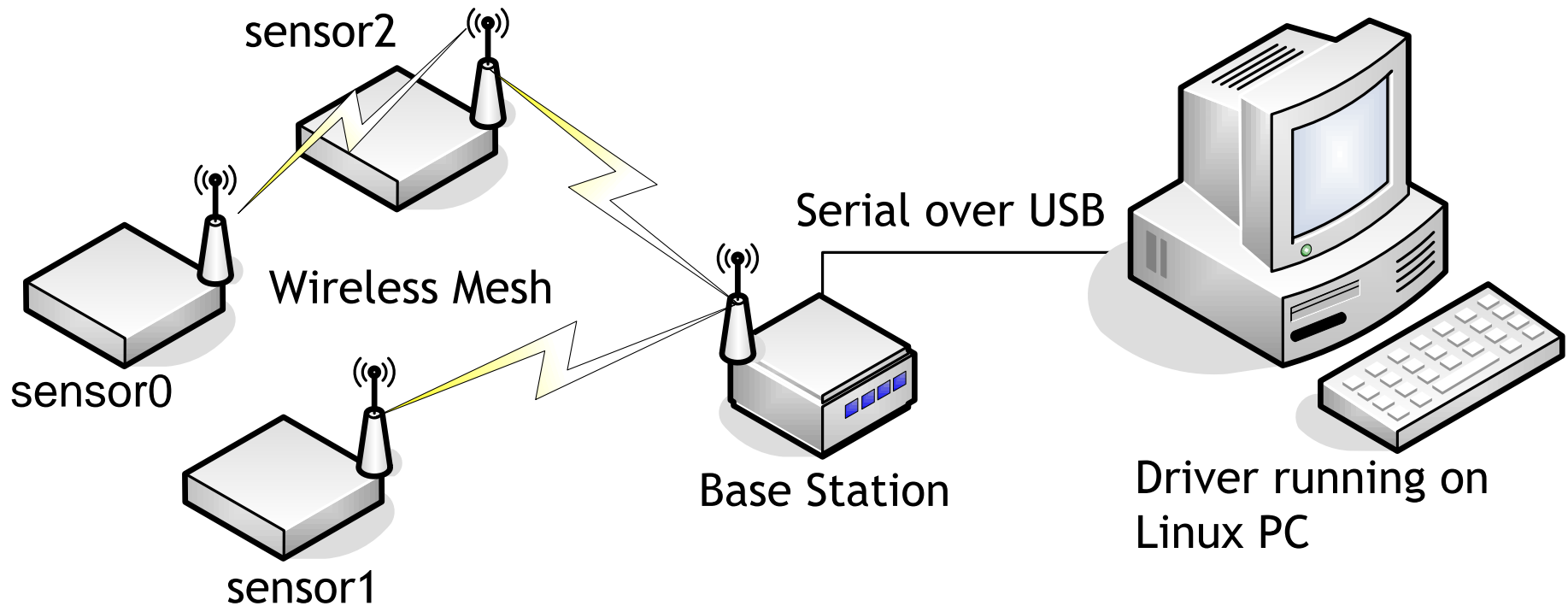
- ◆ Τμήματα κώδικα
- ◆ Επέκταση των δυνατοτήτων του πυρήνα
  - ➔ Οδηγοί συσκευής, συστήματα αρχείων
- ◆ Δυναμική εισαγωγή / αφαίρεση
  - ➔ Ενώ ο πυρήνας εκτελείται
  - ➔ Απαιτούνται δικαιώματα root
- ◆ Εκτέλεση σε kernelspace

# Οδηγοί συσκευής

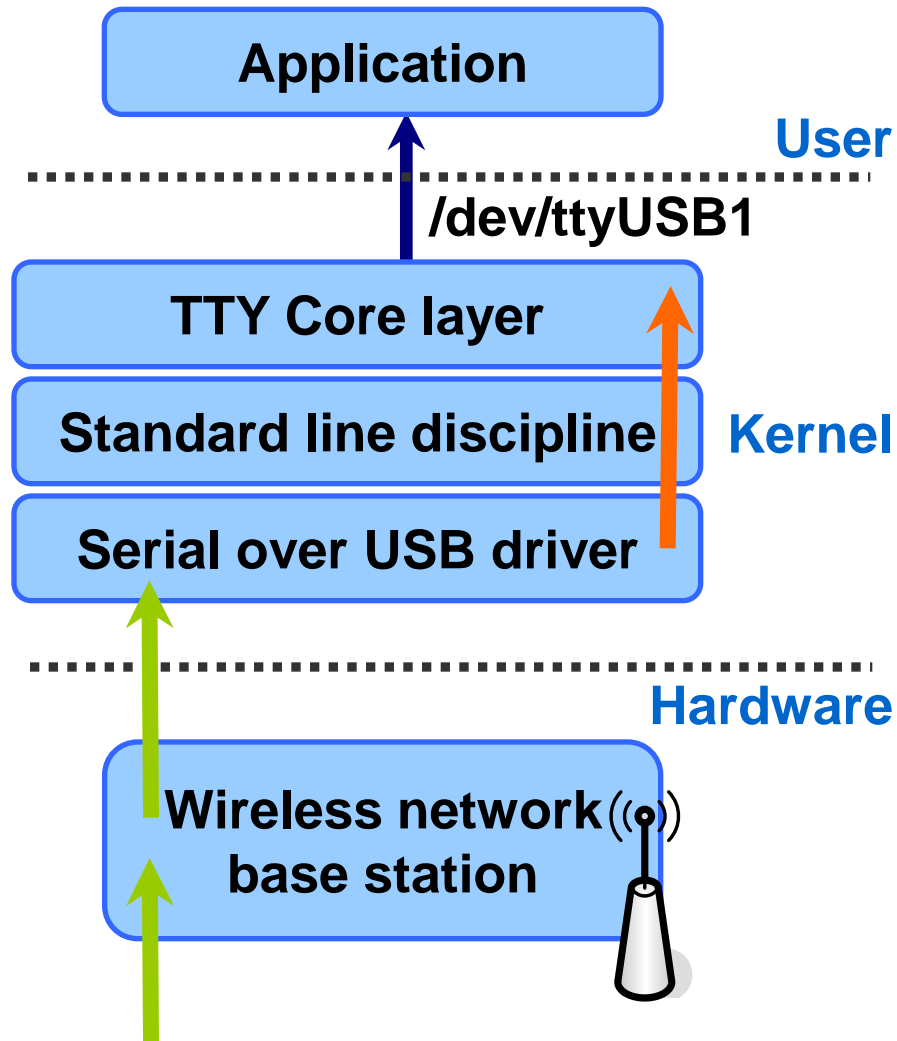


- ◆ Απόκρυψη των λεπτομερειών
- ◆ Καθορισμένο interface με τον πυρήνα
- ◆ Δυνατότητα δυναμικής εισαγωγής ως module
- ◆ Εφαρμογές ανεξάρτητες από το hardware

# Driver για ένα πραγματικό σύστημα



# Μα υπάρχει ήδη driver!



# Ναι, αλλά...



- ◆ Όλα τα πακέτα δεδομένων, χωρίς διάκριση, στο /dev/ttyUSB1
- ◆ Για μόνο μία διεργασία
- ◆ Δεν υπάρχει μηχανισμός για λεπτομερή ανάθεση δικαιωμάτων πρόσβασης
  - ➔ ανά χρήστη, ανά αισθητήρα, ανά μέτρηση
  - ➔ ανάγκη για διακριτή αντιμετώπιση



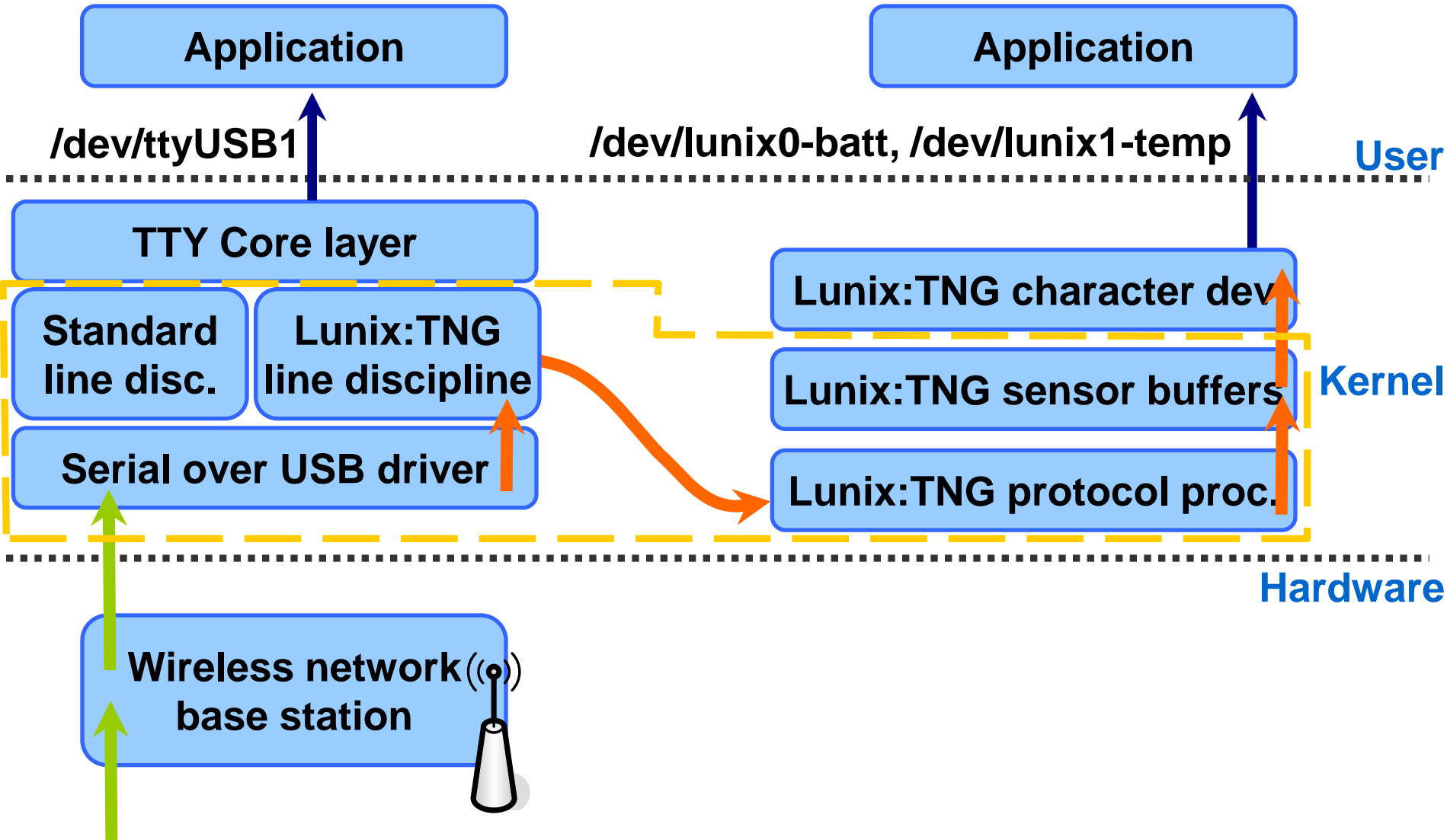
# Οπότε...



## ◆ Lunix:TNG

- ➔ driver για τον πυρήνα του Linux
- ➔ Χωριστό αρχείο συσκευής ανά αισθητήρα και μέτρηση
  - /dev/lunix0- {batt, temp, light}
  - /dev/lunix1- {batt, temp, light}
- ➔ Ταυτόχρονη πρόσβαση από πολλές διεργασίες
- ➔ Περιορισμός δικαιωμάτων ανά αρχείο

# Υπό κατασκευή σύστημα



# Οδηγοί συσκευών στο Linux



## ◆ Συσκευές χαρκτήρων

- ➔ τερματικά, σειριακές/παράλληλες θύρες, κάρτες ήχου

## ◆ Συσκευές block

- ➔ Σκληροί δίσκοι, CD-ROM, δισκέτες

## ◆ Συσκευές δικτύου

- ➔ Κάρτες δικτύου, PPP

# Παράδειγμα χρήσης οδηγού (1)

```
user@utopia:~/linux-tng$ make
gcc -Wall -o mk_lookup_tables mk_lookup_tables.c -lm
./mk_lookup_tables >linux-lookup.h
make -C /lib/modules/2.6.25/build M=/home/user/linux-tng
modules
make[1]: Entering directory `/usr/src/linux-source-2.6.25'
CC [M] /home/user/linux-tng/linux-module.o
CC [M] /home/user/linux-tng/linux-chrdev.o
CC [M] /home/user/linux-tng/linux-ldisc.o
CC [M] /home/user/linux-tng/linux-protocol.o
CC [M] /home/user/linux-tng/linux-sensors.o
LD [M] /home/user/linux-tng/linux.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/user/linux-tng/linux.mod.o
LD [M] /home/user/linux-tng/linux.ko
make[1]: Leaving directory `/usr/src/linux-source-2.6.25'
gcc -Wall -o linux-attach linux-attach.c
user@utopia:~/linux-tng$ make
```

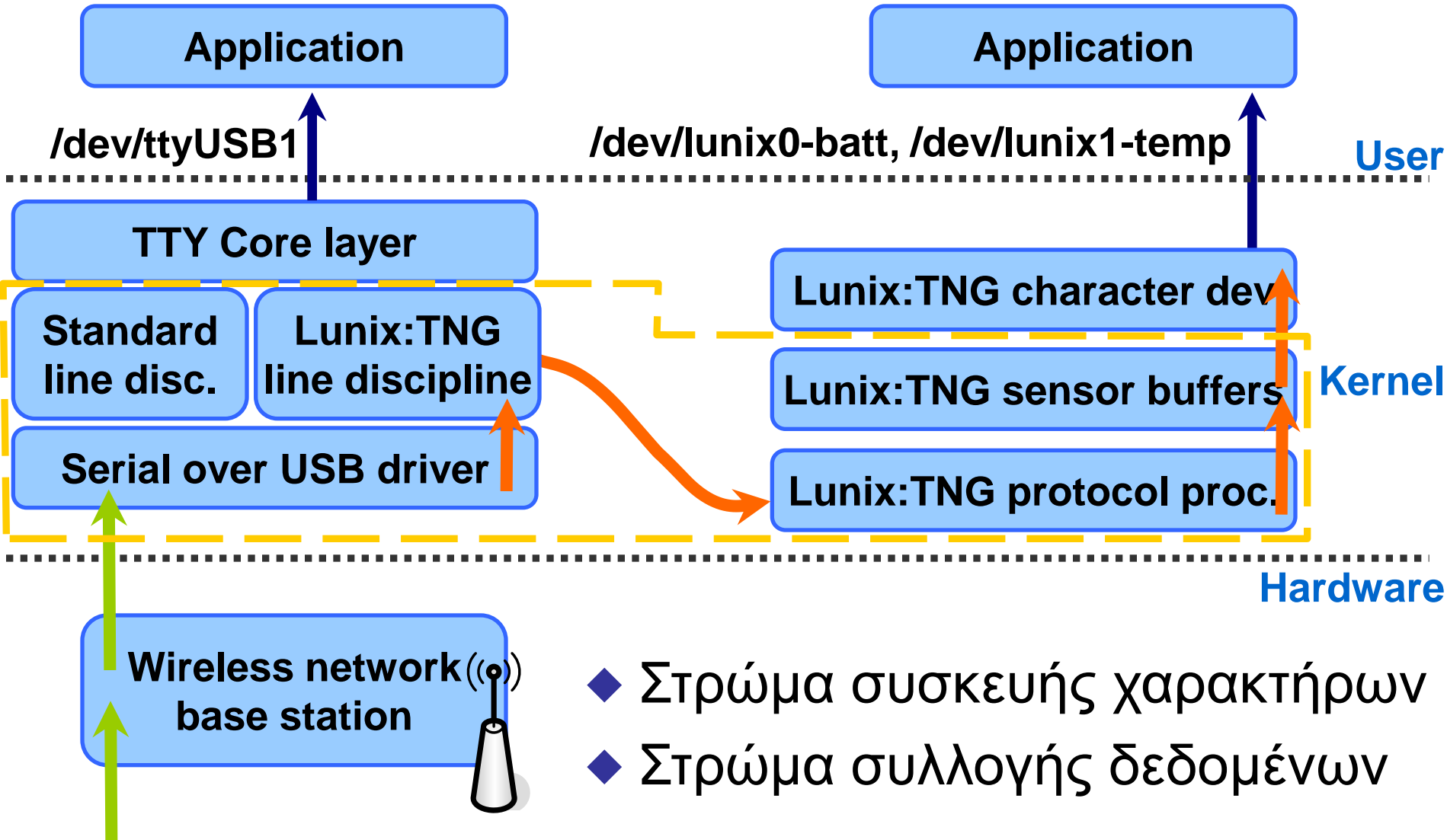
# Παράδειγμα χρήσης οδηγού (2)

```
user@utopia:~/linux-tng$ su -  
Password:  
utopia:~# cd ~user/linux-tng  
utopia:/home/user/linux-tng# insmod ./linux.ko  
utopia:/home/user/linux-tng# ./linux_dev_nodes.sh  
utopia:/home/user/linux-tng# ./linux-attach /dev/ttyUSB1  
tty_open: looking for lock  
tty_open: trying to open /dev/ttyUSB1  
tty_open: /dev/ttyUSB1 (fd=3) Line discipline set on  
/dev/ttyUSB1, press ^C to release the TTY...
```

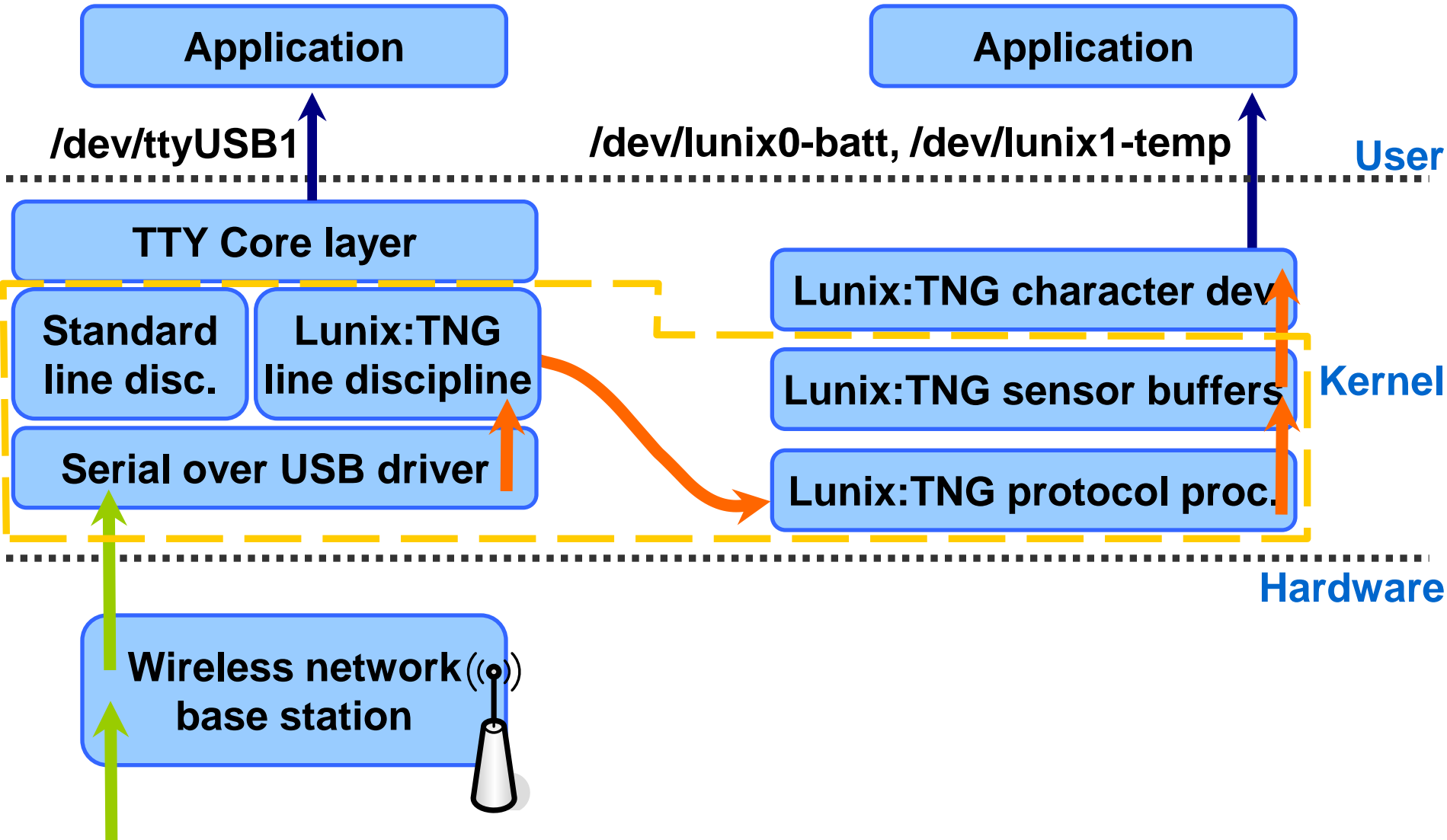
Οπότε μετά:

```
user@utopia:~$ cat /dev/linux2-temp  
27.791  
27.791  
27.693  
27.791  
^C
```

# Υπό κατασκευή σύστημα



# Στρώμα συσκευής χαρκτήρων



# Τα πάντα είναι αρχεία

```
$ ls -l /dev
```

```
crw-rw----  1 root  audio      14,   3 Jul  5  2000 /dev/dsp
crw-rw-rw-   1 root   root        1,   3 Jul 21  2002 /dev/null
crw-----  1 root   root        4,   1 Apr 22 18:10 /dev/tty1
crw-rw----  1 root  dialout    4,  64 Jul  5  2000 /dev/ttyS0
crw-rw-rw-   1 root   root        1,   5 Jul 21  2002 /dev/zero
brw-rw----  1 root   disk       3,    0 Jul  5  2000 /dev/hda
brw-rw----  1 root  floppy     2,    0 Jul  5  2000 /dev/fd0
```

- ◆ Ειδικά αρχεία, πύλες πρόσβασης σε συσκευές
  - ➔ Κατάλογος /dev
- ◆ Κλήσεις συστήματος στο ειδικό αρχείο
  - ➔ open, close, read, write, lseek, ioctl



# Διεπαφή με τον πυρήνα του Linux (1)

```
$ ls -l /dev
```

crw-rw----	1	root	audio	14,	3	Jul 5 2000	/dev/dsp
crw-rw-rw-	1	root	root	1,	3	Jul 21 2002	/dev/null
crw-----	1	root	root	4,	1	Apr 22 18:10	/dev/tty1
crw-rw----	1	root	dialout	4,	64	Jul 5 2000	/dev/ttyS0
crw-rw-rw-	1	root	root	1,	5	Jul 21 2002	/dev/zero
brw-rw----	1	root	disk	3,	0	Jul 5 2000	/dev/hda
brw-rw----	1	root	floppy	2,	0	Jul 5 2000	/dev/fd0

## ◆ Major / minor numbers

- ➔ major: Είδος συσκευής, καθορίζει τον οδηγό
- ➔ minor: Καθορίζει τη συσκευή, πχ. αριθμός σειριακής θύρας

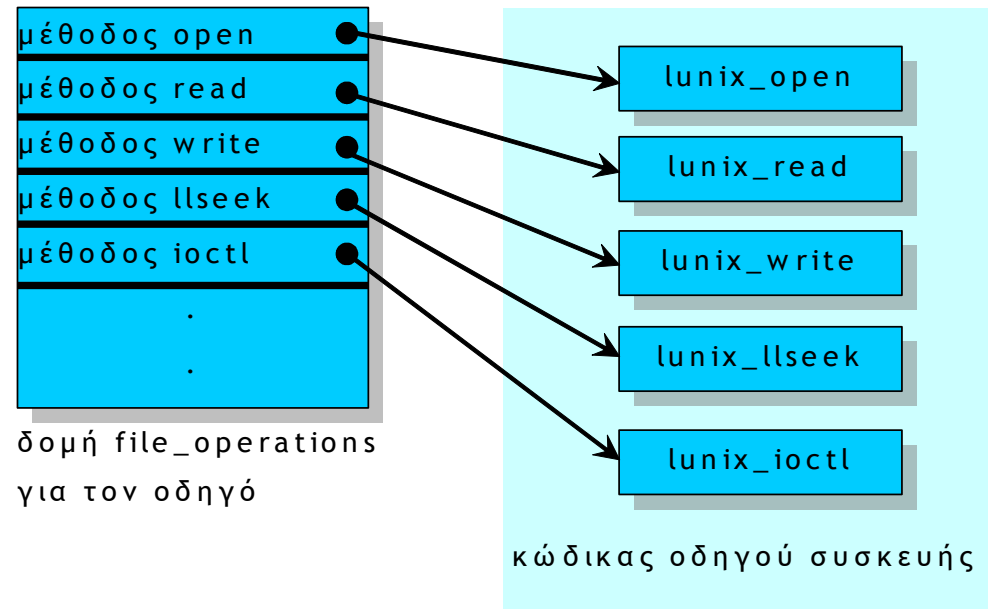
## ◆ Δέσμευση major number με register\_chrdev\_region

# Διεπαφή με τον πυρήνα του Linux (2)

```
int (*open) (struct inode *, struct file *);  
void (*release) (struct inode *, struct file *);  
ssize_t (*read) (struct file *, char *, size_t, loff_t);  
ssize_t (*write) (struct file *, const char *, size_t, loff_t);  
loff_t (*llseek) (struct file *, loff_t, int);  
int (*unlocked_ioctl) (struct file *, unsigned int, unsigned long);
```

- ◆ Υπηρεσίες οδηγού συσκευής χαρακτήρων
- ◆ Υλοποιούν κλήσεις συστήματος
- ◆ Εκτελούνται από τον πυρήνα
  - ➔ Κατά την πρόσβαση στο ειδικό αρχείο
  - ➔ Μέσω της δομής file\_operations

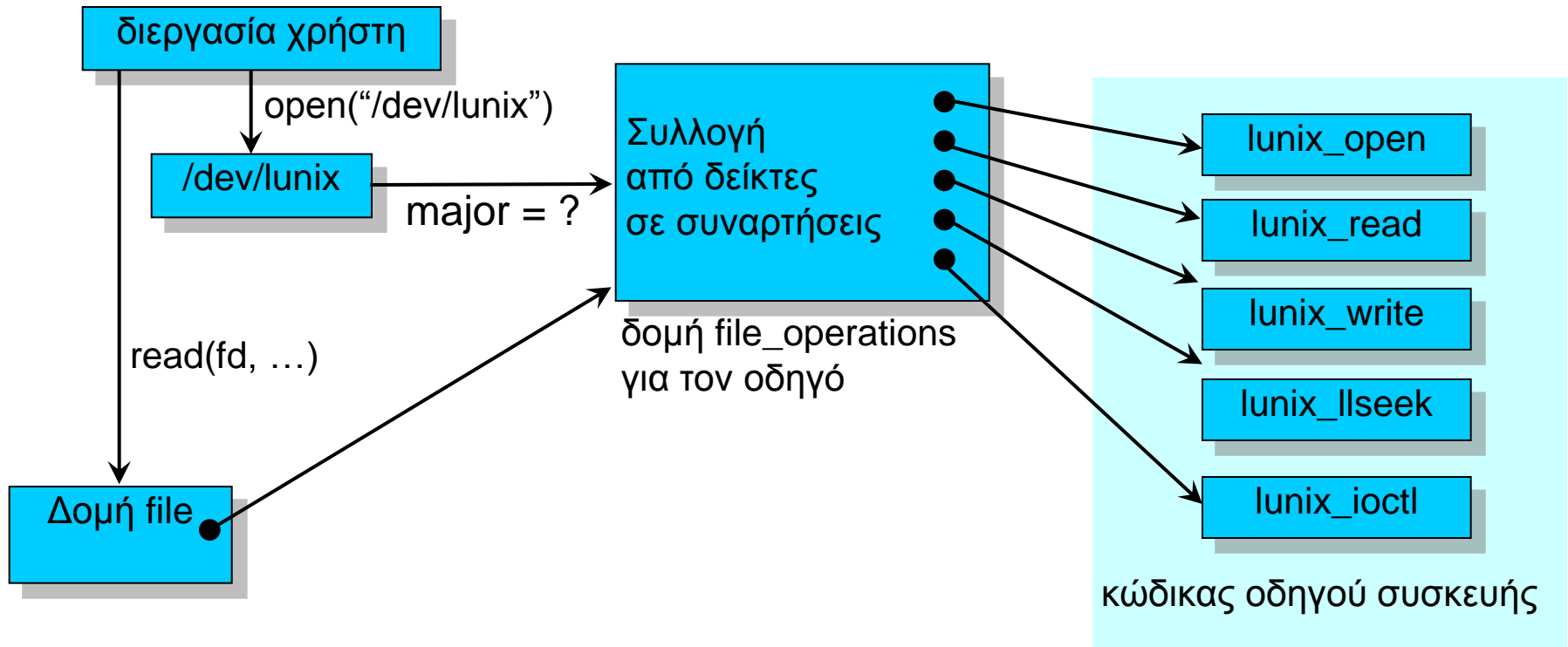
# Διεπαφή με τον πυρήνα του Linux (3)



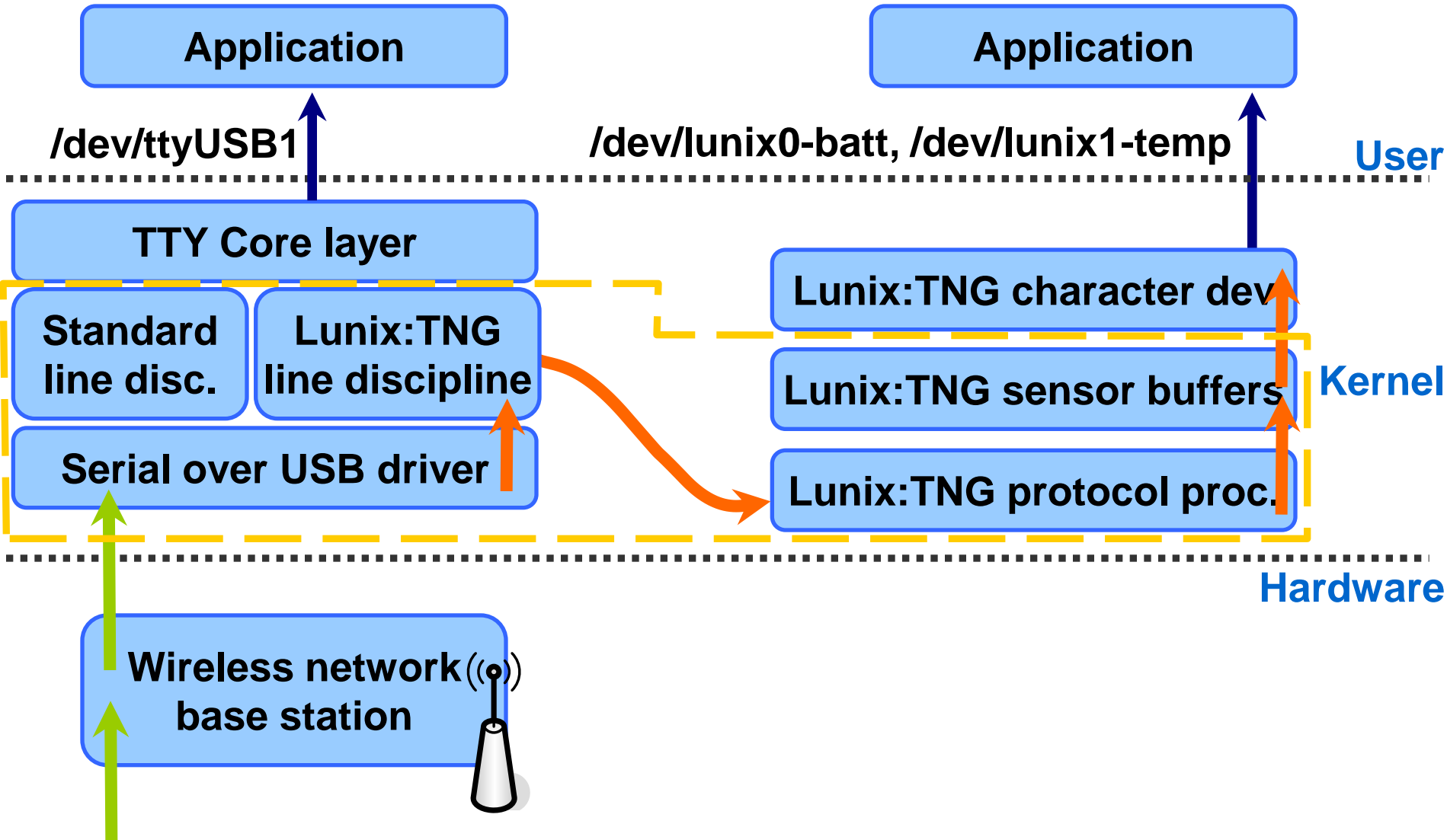
## ◆ Δομή `file_operations`

- ➔ Για κάθε λειτουργία, δείκτης σε αντίστοιχη συνάρτηση
- ➔ Μία για κάθε major number
- ➔ Δήλωση κατά την καταχώρηση νέας συσκευής χαρακτήρων (`cdev_add`)

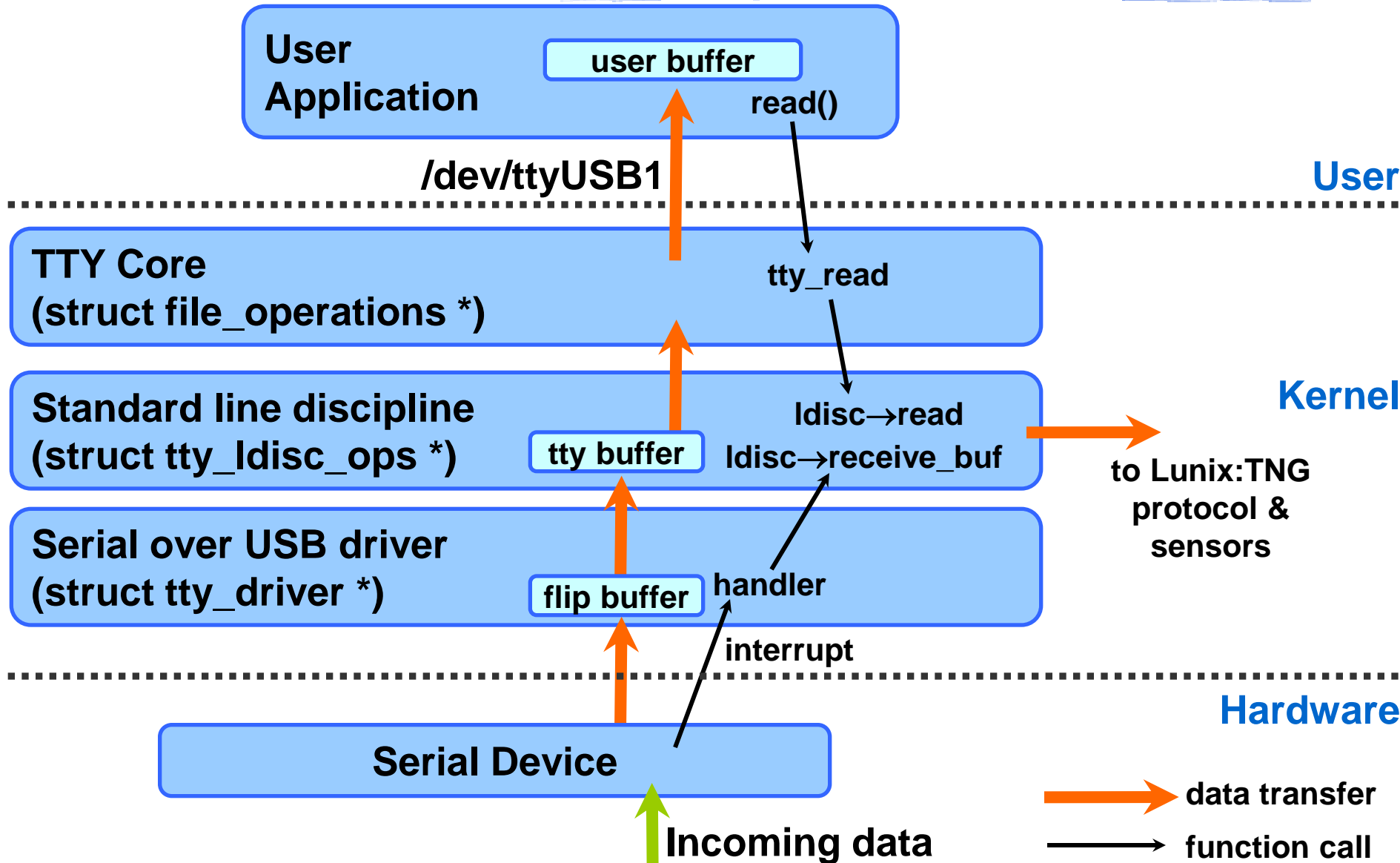
# Διεπαφή με τον πυρήνα του Linux (4)



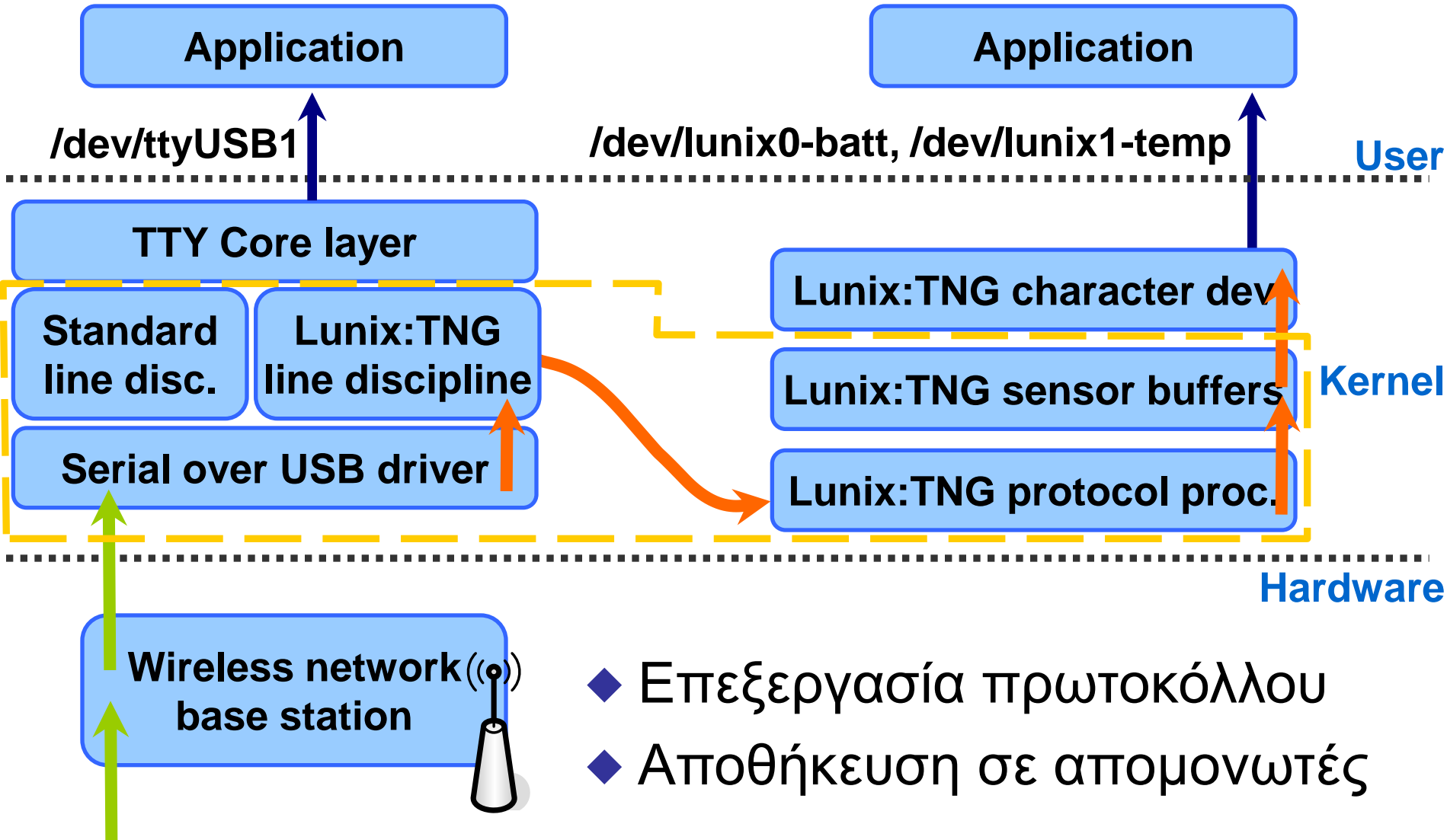
# Στρώμα συλλογής δεδομένων



# Lunix:TNG line discipline



# Στρώμα συλλογής δεδομένων (2)



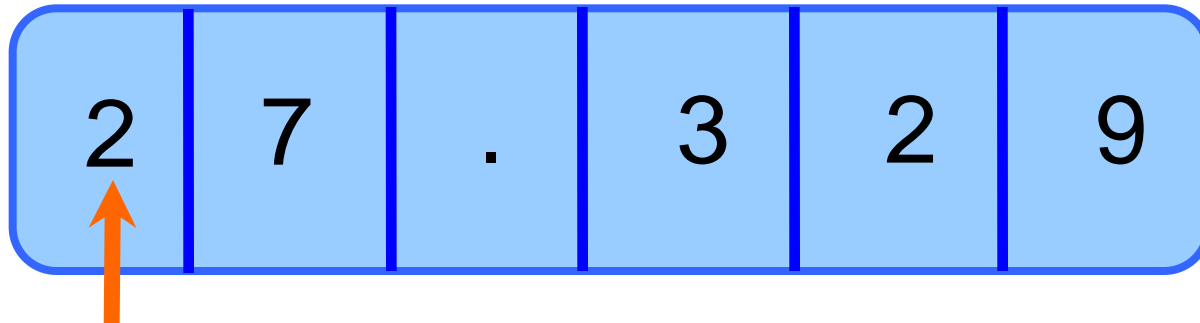
# Ζητήματα Υλοποίησης (1)

- ◆ Πρόσβαση σε δεδομένα χώρου χρήστη
  - ➔ `copy_to_user()`, `copy_from_user()`
- ◆ Πώς πέφτουν οι διεργασίες για ύπνο;
  - ➔ Μπλοκάρισμα διεργασιών σε ουρές αναμονής
  - ➔ Δεν είναι έτοιμες προς εκτέλεση, δεν καταναλώνουν χρόνο CPU
- ◆ Κρίσιμα τμήματα στον κώδικα του πυρήνα
  - ➔ Σημαφόροι, spinlocks

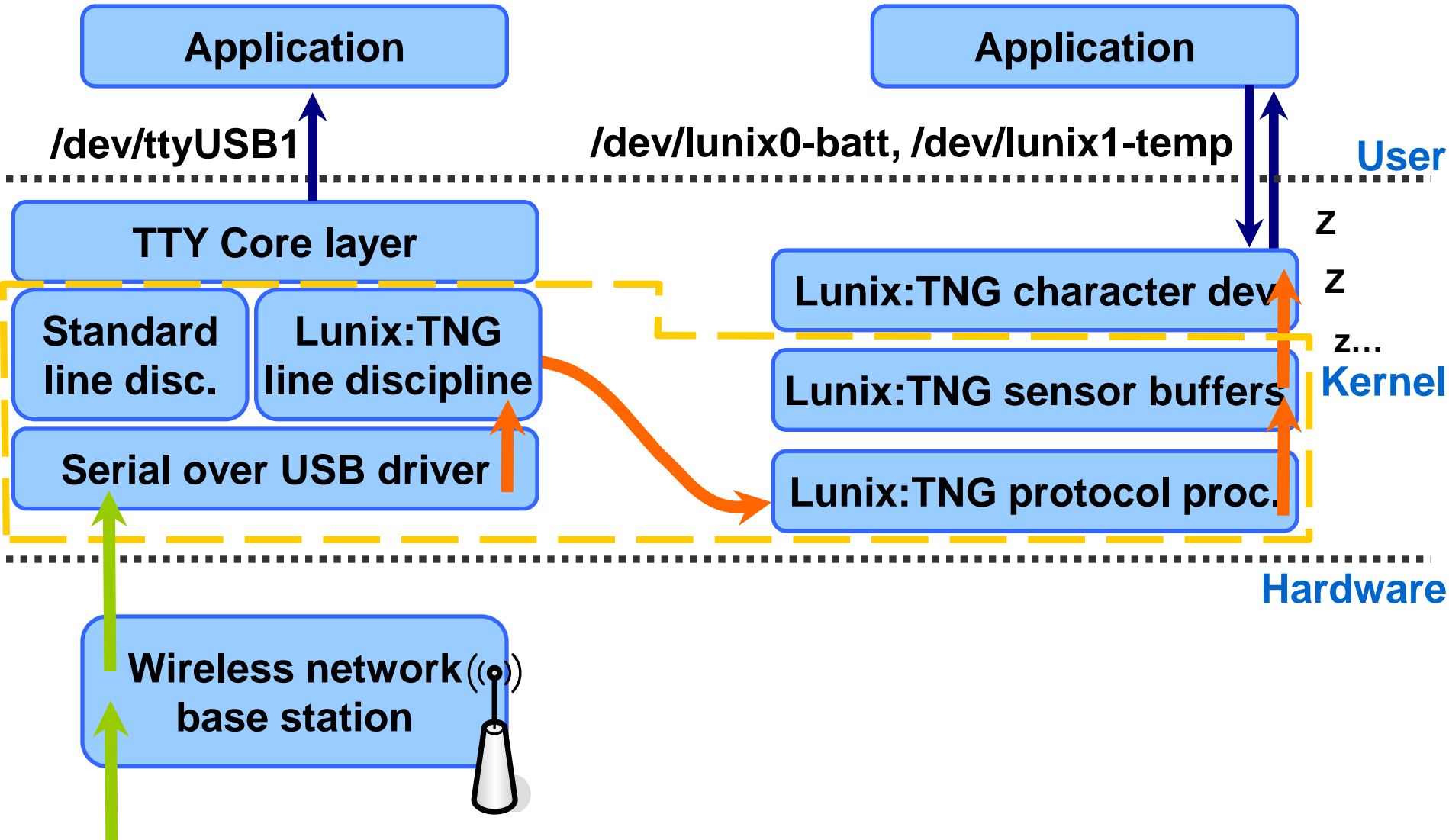


# Ζητήματα Υλοποίησης (2)

- ◆ Η `read()` είναι το κέντρο της υλοποίησης
  - ➔ Κρατάει την μορφοποιημένη τιμή σε προσωρινό χώρο μνήμης
  - ➔ Δίνει από εκεί bytes ανάλογα με το `f_pos`
  - ➔ Όταν το `f_pos` είναι μηδέν τι γίνεται;
    - Υπάρχουν φρέσκα δεδομένα; Χρήση timestamps
    - Αν δεν υπάρχουν πρέπει να πέφτει για ύπνο
  - ➔ Όταν έρθουν ξυπνάει και τα επιστρέφει



# Ζητήματα Υλοποίησης (3)



# Πρακτικά Ζητήματα

- ◆ Υλοποίηση μέσα σε εικονική μηχανή
  - ➔ root privilege σε Debian, σε μηχανή QEMU-KVM
- ◆ Από πού ξεκινάω;
  - ➔ make στον έτοιμο κώδικα, ανάγνωση του linux-sensors.c, προσθήκες στα linux-chrdev.{h,c}
  - ➔ Σας δίνεται σκελετός με σχόλια
- ◆ Kernel logging, printk, /var/log/kern.log
- ◆ Linux Cross Reference (LXR)

# Έλεγχος και πιθανές επεκτάσεις

## ◆ Έλεγχος σωστής λειτουργίας

- ➔ Από userspace, με ταυτόχρονα **cat**
- ➔ και με άλλα εργαλεία, π.χ. **dd**

## ◆ Πιθανές επεκτάσεις

- ➔ Υποστήριξη `ioctl()` για raw / cooked modes
- ➔ Υποστήριξη memory-mapped I/O με `mmap()`
- ➔ Userspace πρόγραμμα για καταγραφή και παρουσίαση μετρήσεων

# Ιστοσελίδα – λίστα του μαθήματος



[http://www.cslab.ece.ntua.gr/courses/  
compsyslab/](http://www.cslab.ece.ntua.gr/courses/compsyslab/)

[os-lab@lists.cslab.ece.ntua.gr](mailto:os-lab@lists.cslab.ece.ntua.gr)

# Βιβλιογραφία

- ◆ “Linux Device Drivers”, 3rd Edition, Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman.

<http://lwn.net/Kernel/LD3>

