

# Basics of embedded programming

Fabrice.Kordon@lip6.fr



# As an introduction...

## Embedded programming?

- Constraints
  - ▶ Energy, memory, CPU
- Cross compilation (often)

## Programming on mobile platforms?

- Embedded programming
- User interface centered programming
  - ▶ MVC model
- Reactive programming

# Cross compilation



## Conditions

- Create a program for an architecture different from the target one

## Two steps test & debug

- On an emulator
- On the terminal

# Programming constraints

## Limited resources

### Memory

- ▶ Management is delicate
- ▶ Android : garbage collector (Java)
- ▶ iOS : reference counters
  - Now ARC (Automatic reference Counting)
  - Objective-C (can be deactivated)
  - Swift (built-in)

### Energy consumption

- ▶ Caution with peripheral (GPS, cameras/HDR/Flash, etc.)

### CPU (energy related)

- ▶ Better choose efficient algorithms

## Emergency events

-  No memory or energy left
-  Phone call (smartphones... and some tablets)

# Other programming issues

## Ergonomics

- Ease to use
- Respect of the «look and feel»
  - ▶ Graphic chart
  - ▶ Dedicated mechanisms

## Reliability

- Avoid crashes (typically memory troubles)

## Security

- Protection of personnel data
  - ▶ Dedicated API (access to devices, etc.)

## Performance

- Beware of algorithm complexity
  - ▶ Concerns often integrated in the framework

# As a conclusion...

-  Care about all these concerns when programming
-  This is the key for your Apps success
-  You need «the good idea» too of course  

**In short, many compromises...**

...Be careful, Apple may find reasons to reject Apps from your mistakes

