

# Programma

---

- legOS
- geometria, meccanica, sensori ed attuatori Mindstorm

# legOS

---

- sistema operativo *embedded*
- open source
- preemptive multitasking
- caricamento dinamico di programmi
- gestione dinamica della memoria
- IR networking

# RCX

---

- CPU (Hitachi H8/3292)
  - indirizzamento a 16 bit
  - 16 registri 8 bit (accesso a 8 registri da 16 bit)
- 16 kB ROM (programma per lanciare il firmware)
- 32 kB RAM esterna
  - firmware, programmi utente
  - memoria LCD, registri porte I/O, vettore interrupt

# RCX (2)

---

- sottosistemi
  - timer a 16 bit (OS timing)
  - timer a 8 bit a 2 canali (timing speaker e IR comm.)
  - convertitore A/D (campionamento sensori)
  - porte di I/O (tasti, motori, ecc.)

# kernel legOS

---

- monolitico (niente moduli)
- interfaccia con programmi utente via script di caricamento dinamico
- start
  - single- o multi-task
  - 3 task:
    - idle (risparmio energia)
    - packet\_consumer (monitor porta IR)
    - key\_handler (monitor tasti)

# Timing

---

- polling dei sottosistemi ogni millisecondo
  - motori
  - suono
  - check timeout networking
  - tasti
  - batterie
  - LCD
- context switch? (per default timeslice di 20 ms)

# Gestione dei task

---

- programma: immagine binaria in RAM  
esecuzione di un programma per volta
- task/processi: thread in esecuzione  
condivisione memoria
- preemption (timesharing con priorità)
- stati: Sleeping, Running, Waiting, Zombie, Dead  
wake-up su richiesta dello scheduler  
problema per eventi rapidi

# Gestione dei task (2)

---

- creazione di nuovi processi tramite `execi`  
funzione da eseguire, suoi argomenti, priorità
- priorità: da 20 (alta) ad 1 (bassa)



# Scheduler

---

- scheduler
  - attivazione al termine di una timeslice o su wait
  - controllo del semaforo dei task
  - rimozione eventuale zombie
  - check dei task per priorità
  - wait: wake-up?
  - idle non è mai waiting

# Gestione della memoria

---

- allocazione continua (risorsa scarsa!)
- organizzazione in blocchi
- allocazione first-fit con eventuale suddivisione  
frammentazione interna vs. esterna
- fusione alla deallocazione
- accesso a tutta la memoria  
comunicazione tra processi :)  
sovrascrittura :(
- allocazione dinamica: rischio out of memory

# Comunicazione tra processi

---

- memoria condivisa
- semafori
- IR networking
  - libreria per applicazioni su host

# Periferiche

---

- Motori
- Sensori
- Suono
- LCD
- Tasti

---

- Funzionamento del differenziale

# Apparato locomotore

---

- Differential drive
  - Simple
  - Dual
- Ruote sterzanti
- Synchro drive

- 
- aritmetica intera
    - approssimazioni
    - overflow
    - underflow
  - isteresi

# Orientamento

---

- riferimenti esterni
- riferimenti interni



