

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

Interfaccia

Pratica

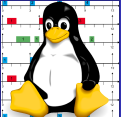
# The SCHED\_DEADLINE Linux Kernel Scheduler and its Usage

Alessio Balsini

[a.balsini@sssup.it](mailto:a.balsini@sssup.it)



Linux Day Pisa - 24/10/2015



# Sommario

Alessio Balsini

Linux Day  
2015

Introduzione

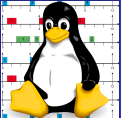
FCFS  
RR  
– Real-Time  
RM  
EDF  
– Reservation  
CBS

Linux  
Scheduler

SCHED\_DL

Teoria  
Interfaccia  
Pratica

- Introduzione allo scheduling
  - Nozioni di base
  - Multiprogrammazione
  - Scheduling
  - Nomenclatura
- Semplici algoritmi
  - FCFS
  - RR
- Real-Time
  - Task periodici
  - RM
  - EDF
- Isolation
  - CBS
- Scheduling su Linux
  - Struttura e classi
- SCHED\_DEADLINE
  - Teoria
  - Interfaccia
  - Pratica



# Sistemi Multiprogrammati: Concorrenza

Alessio Balsini

Linux Day  
2015

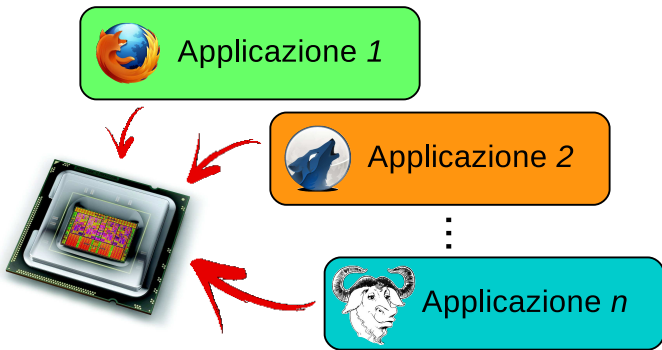
## Introduzione

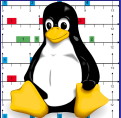
- FCFS
- RR
  - Real-Time
- RM
- EDF
  - Reservation
- CBS

Linux  
Scheduler

## SCHED\_DL

- Teoria
- Interfaccia
- Pratica





# Scheduler

Alessio Balsini

Linux Day  
2015

## Introduzione

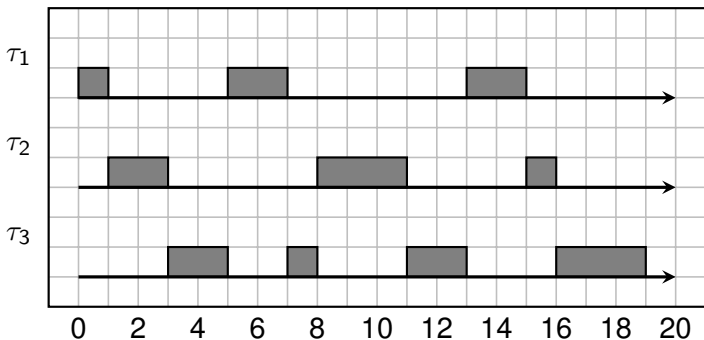
- FCFS
- RR
- Real-Time
- RM
- EDF
- Reservation
- CBS

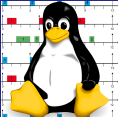
Linux  
Scheduler

## SCHED\_DL

- Teoria
- Interfaccia
- Pratica

Scheduler: componente del sistema operativo che decide quale task debba essere eseguito.





# Rappresentazione e Nomenclatura

Alessio Balsini

Linux Day  
2015

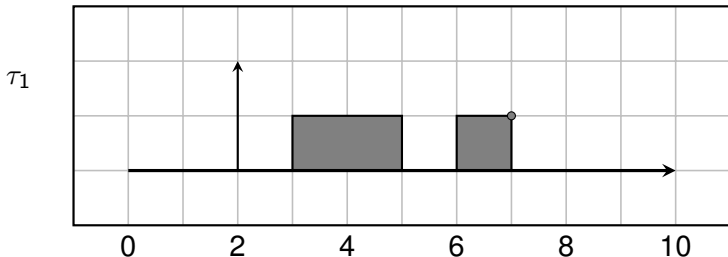
Introduzione

FCFS  
RR  
– Real-Time  
RM  
EDF  
– Reservation  
CBS

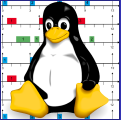
Linux  
Scheduler

SCHED\_DL

Teoria  
Interfaccia  
Pratica



- $C_1 = 3$ : *computation time*, tempo di esecuzione richiesto.
- $a_1, t = 2$ : *activation time*, task pronto per l'esecuzione.
- $s_1, t = 3$ : *start time*, task messo in esecuzione.
- $t = 5$ : *preemption*, esecuzione revocata.
- $f_1, t = 7$ : *finishing time*, task completato.



# Algoritmi di Scheduling: FCFS

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

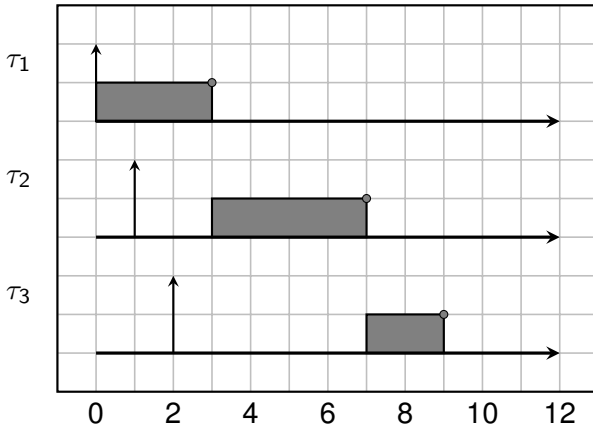
Teoria

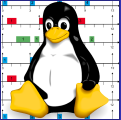
Interfaccia

Pratica

## FCFS (*First Come First Served*)

- di tipo FIFO (*Fist In First Out*).





# Algoritmi di Scheduling: RR

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

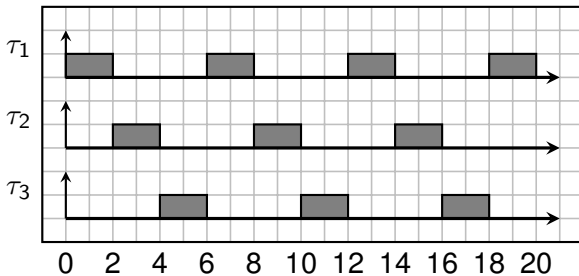
Teoria

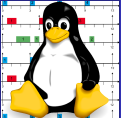
Interfaccia

Pratica

## RR (*Round Robin*)

- di tipo FIFO, con preemption se superato il *time quantum*.





# Real-Time

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

Interfaccia

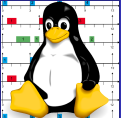
Pratica

Applicazioni *time critical*.



La correttezza dell'output dipende anche dall'istante in cui il risultato è prodotto.





# Real-Time

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

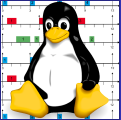
Interfaccia

Pratica

## Applicazioni multimediali.



Le deadline miss si percepiscono come un peggioramento della qualità del servizio (QoS).



# Real-Time

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

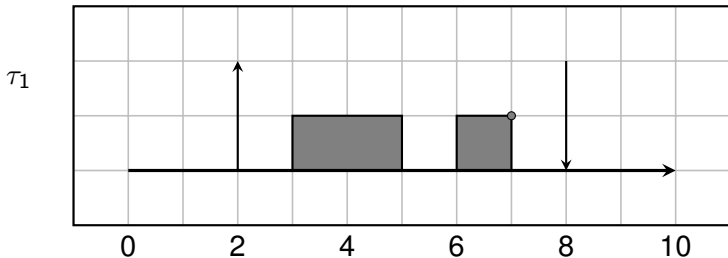
Teoria

Interfaccia

Pratica

La correttezza dell'output dipende anche dall'istante in cui il risultato è prodotto.

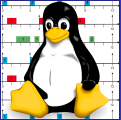
- Introduce il concetto di *deadline*.



- $d_1, t = 8$ : *absolute deadline*, scadenza del task.
- $D_1 = 6$ : *relative deadline*, scadenza rispetto all'istante di attivazione.

Real-time: **garantire** l'esecuzione dei task entro le deadline:

- no *deadline miss*.



# Task Periodici

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

Interfaccia

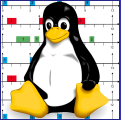
Pratica

## Tipico task di controllo.

```
void * my_control_task(void * task_attributes)
{
    /* initialization */

    while (condition) {
        data = get_sensor_data();
        control = control_algorithm(data);
        send_control_signal_to_actuators(control);

        wait_for_next_activation(task_attributes);
    }
}
```



# Task Periodici

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

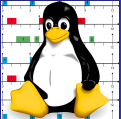
Interfaccia

Pratica

## Su Arduino.

```
void setup()
{
    /* Data structures & I/O initialization */
}

void loop()
{
    /* Do something */
    delay(...);
}
```



# Task Periodici

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

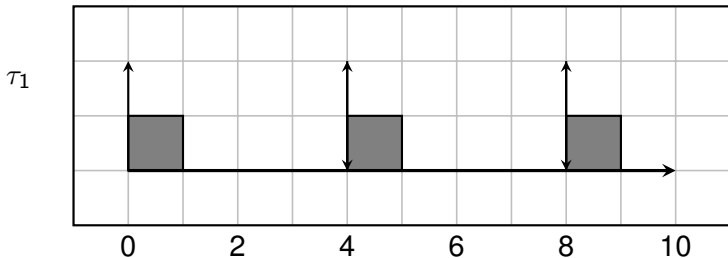
Teoria

Interfaccia

Pratica

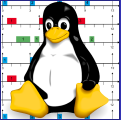
Task periodico: task attivato periodicamente.

- Periodo:  $T_i$ .
- $a_{i,k} = a_{i,0} + kT_i$ .



Job: ciò che un task esegue periodicamente.

- Spesso  $D_i = a_{i+1}$ .



# Algoritmi di Scheduling RT

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

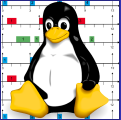
Teoria

Interfaccia

Pratica

- Garantire (se possibile) le deadline.
- Esistono molti algoritmi e tanta letteratura.
  - Si può verificare se un taskset (insieme di task) sia schedulabile (no deadline miss) con metodi specifici per ciascun algoritmo.

Il Worst Case Execution Time (*WCET*) identifica la durata massima di un job.



# Algoritmi di Scheduling RT: RM

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

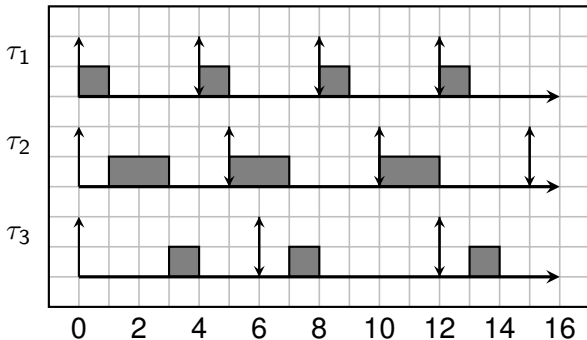
Teoria

Interfaccia

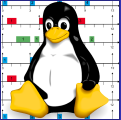
Pratica

## RM (*Rate Monotonic*)

- Priorità inversamente proporzionale alla relative deadline.



- $C_1 = 1$  ,  $D_1 = T_1 = 4$ ;
- $C_2 = 2$  ,  $D_2 = T_2 = 5$ ;
- $C_3 = 1$  ,  $D_3 = T_3 = 6$ .



# Algoritmi di Scheduling RT: EDF

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

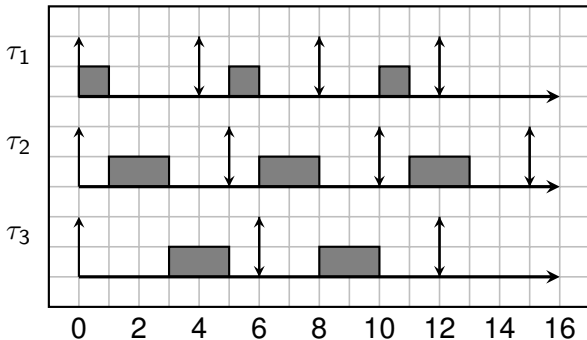
Teoria

Interfaccia

Pratica

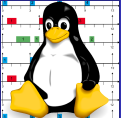
## EDF (*Earliest Deadline First*)

- Priorità inversamente proporzionale alla absolute deadline.



- $C_1 = 1$  ,  $D_1 = T_1 = 4$ ;
- $C_2 = 2$  ,  $D_2 = T_2 = 5$ ;
- $C_3 = 2$  ,  $D_3 = T_3 = 6$ .





# Algoritmi di Scheduling RT: EDF

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

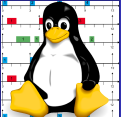
Interfaccia

Pratica

EDF è ottimo: se esiste un algoritmo in grado di schedulare un dato taskset, allora il taskset è schedulabile con EDF.

- Ma l'attuale implementazione di EDF ha complessità  $O(\log(n))$ .

E... Cosa succede se un task ha un  $C_i$  più grande di quello dichiarato?



# Algoritmi di Scheduling RT: EDF

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

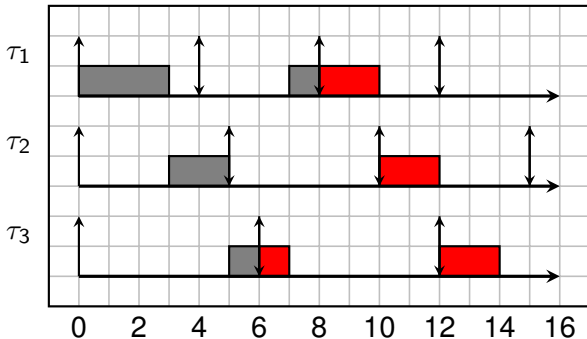
Teoria

Interfaccia

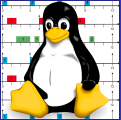
Pratica

Overrun:

- Esecuzione maggiore di quella dichiarata.



- $C_1 = 1^3$ ,  $D_1 = T_1 = 4$ ;
- $C_2 = 2$ ,  $D_2 = T_2 = 5$ ;
- $C_3 = 2$ ,  $D_3 = T_3 = 6$ .



# Algoritmi di Scheduling RT: Reservation

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

Interfaccia

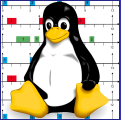
Pratica

I task sono associati a “server”.



Ogni server ha un “budget”:

- il budget si consuma durante l'esecuzione dei suoi task;
- quando il budget finisce, nessuno dei suoi task può eseguire.



# Algoritmi di Scheduling RT: Reservation

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

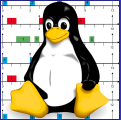
Interfaccia

Pratica

Come si gestisce il budget?

- Quanto vale il budget massimo?
- Quando avviene il ripristino (*replenishment*) del budget?

Ci sono tanti algoritmi per la gestione dei budget.



# Algoritmi di Scheduling RT: CBS

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS  
RR  
– Real-Time  
RM  
EDF  
– Reservation  
CBS

Linux  
Scheduler

SCHED\_DL

Teoria  
Interfaccia  
Pratica

## CBS: Constant Bandwidth Server

- Meccanismo di server che si appoggia ad EDF.

---

Ogni server ha associati:

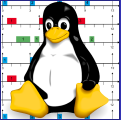
- $T_i$  period;
- $Q_i$  budget massimo;
- $d_i$  deadline (serve a EDF);
- $c_i$  budget;

Quando il budget finisce ( $c_i = 0$ ):

- $c_i = Q_i$  (il replenishment avviene subito);
- $d_i = d_i + T_s$  (deadline posticipata: diminuisce la priorità).

Quando arriva un nuovo task e il server non stava eseguendo,  
se  $c_i \geq (d_i - r_i)Q_i/T_i$ :

- $d_i = a_i + T_i$
- $c_i = Q_i$



# Struttura dello scheduler in Linux

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

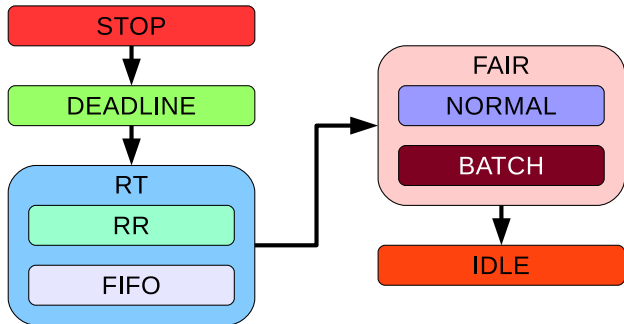
Teoria

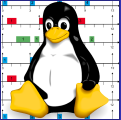
Interfaccia

Pratica

Tanti algoritmi di scheduling, organizzati secondo una lista di *sched\_class*.

L'ordine della lista definisce la priorità.





# SCHED\_DEADLINE: Teoria

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

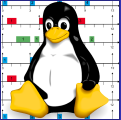
Teoria

Interfaccia

Pratica

Implementa l'algoritmo *Hard-CBS*, un'evoluzione del CBS.

- Il replenishment avviene all'istante della deadline (non è istantaneo come nel CBS).
- Serve ad evitare il fenomeno di *deadline aging*.



# SCHED\_DEADLINE: Interfaccia

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

Interfaccia

Pratica

I parametri che si possono definire per ogni task sono:

- Runtime: massimo budget del task;
- Deadline: deadline del task;
- Period: periodo per il replenishment.

Tutti unsigned a 64 bit che esprimono valori in nanosecondi.

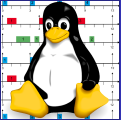
*Admission control:* SCHED\_DEADLINE non accetta nuovi task se non è verificato:

$$\sum_i \frac{Q_i}{T_i} \leq 0.95 \cdot m$$

dove:

- $m$  è il numero di core;
- 0.95 è un numero ottenuto statisticamente.





# SCHED\_DEADLINE: Pratica

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS  
RR  
– Real-Time  
RM  
EDF  
– Reservation  
CBS

Linux

Scheduler

SCHED\_DL

Teoria  
Interfaccia  
Pratica

La libreria *sched.h* non supporta SCHED\_DEADLINE. Si può creare una libreria (*dl.h*) che chiami direttamente le *syscall*:

```
#include <linux/types.h>

#define SCHED_DEADLINE      6

#ifdef __x86_64__
#define __NR_sched_setattr 314
#define __NR_sched_getattr 315

#elif __i386__
#define __NR_sched_setattr 351
#define __NR_sched_getattr 352

#elif __arm__
#define __NR_sched_setattr 380
#define __NR_sched_getattr 381
#endif

struct sched_attr {
    __u32 size;

    __u32 sched_policy;
    __u64 sched_flags;

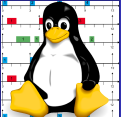
    /* SCHED_NORMAL, SCHED_BATCH */
    __s32 sched_nice;
```

```
/* SCHED_FIFO, SCHED_RR */
__u32 sched_priority;

/* SCHED_DEADLINE (nsec) */
__u64 sched_runtime;
__u64 sched_deadline;
__u64 sched_period;
};

int sched_setattr(pid_t tid,
    const struct sched_attr *attr,
    unsigned int flags)
{
    return syscall(__NR_sched_setattr,
        tid, attr, flags);
}

int sched_getattr(pid_t tid,
    struct sched_attr *attr,
    unsigned int size,
    unsigned int flags)
{
    return syscall(__NR_sched_getattr,
        tid, attr, size, flags);
}
```



# SCHED\_DEADLINE: Pratica

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

Interfaccia

Pratica

```
#include <stdio.h>
#include <stdlib.h>

#include "dl.h"

int main()
{
    struct sched_attr attr;
    unsigned int flags = 0;

    attr.size = sizeof(attr);
    attr.sched_flags = attr.sched_nice
        = attr.sched_priority = 0;

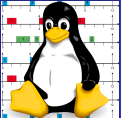
    attr.sched_policy = SCHED_DEADLINE;
    attr.sched_runtime = 10 * 1000 *
        1000;
    attr.sched_period = attr.
        sched_deadline = 1000 * 1000
        * 1000;

    if (sched_setattr(0, &attr, flags)
        < 0) {
        perror("sched_setattr");
        exit(-1);
    }

    for (;;) {
        job();

        sched_yield();
    }

    return 0;
}
```



# SCHED\_DEADLINE: Sessione Dimostrativa

Alessio Balsini

Linux Day  
2015

Introduzione

FCFS

RR

– Real-Time

RM

EDF

– Reservation

CBS

Linux

Scheduler

SCHED\_DL

Teoria

Interfaccia

Pratica



Alessio Balsini

[ [a.balsini@sssup.it](mailto:a.balsini@sssup.it) ]

