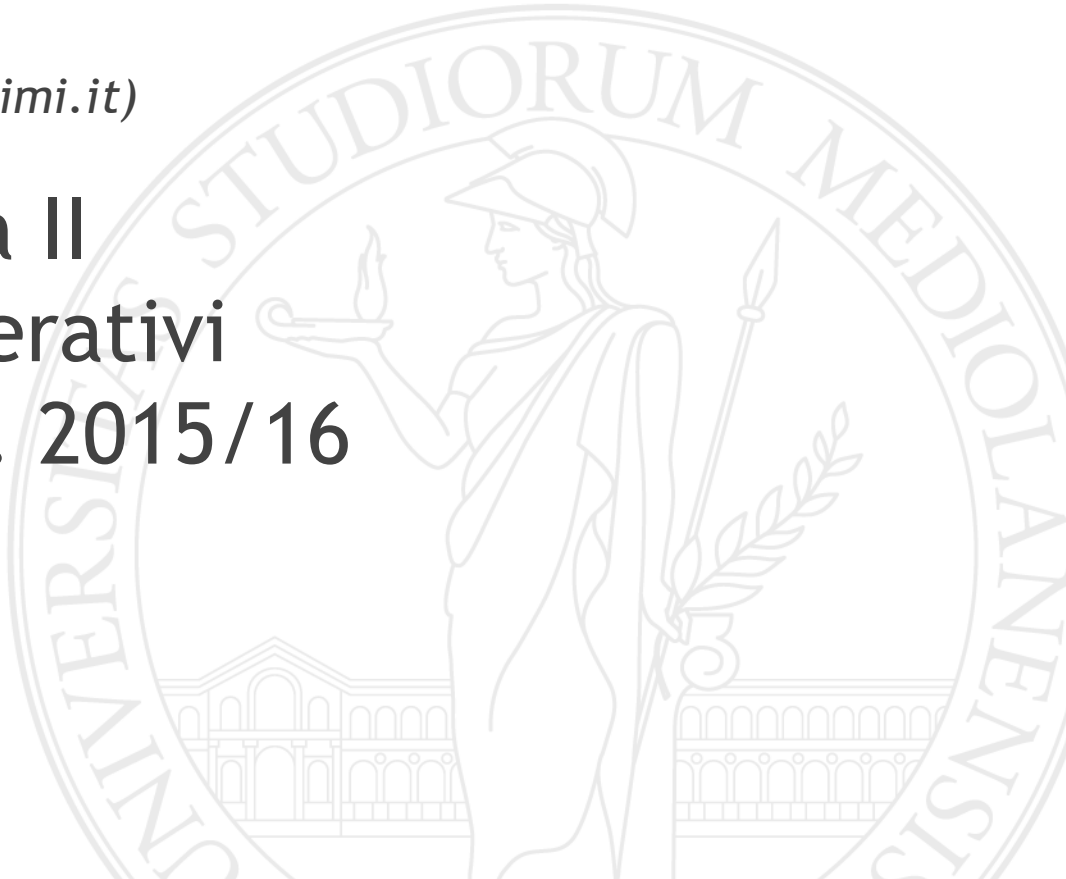
UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI INFORMATICA

Alberto Ceselli
(alberto.ceselli@unimi.it)

Informatica II
Sistemi Operativi
DIGIP - a.a. 2015/16



Sistemi Operativi

(modulo di Informatica II)

Sincronizzazione dei processi

Patrizia Scandurra

Università degli Studi di Bergamo

a.a. 2014-15

Hardware per la sincronizzazione

- Molti sistemi forniscono istruzioni hardware per la risoluzione del problema della sezione critica
- Monoprocessori – possono **disabilitare gli interrupt**
  - La sequenza di istruzioni della "sezione critica" sarà eseguita in ordine senza interruzioni
  - Di solito troppo inefficiente sui sistemi multiprocessore
    - disabilitazione/riabilitazione degli interrupt su ogni processore
  - I SO che lo usano sono carenti in scalabilità
    - L'inefficienza aumenta all'aumentare del numero di CPU
- I calcolatori moderni forniscono **istruzioni hardware speciali che funzionano in modo atomico**
    - Atomico = non-interrompibile
  - Controllare e modificare il contenuto di una parola *get-and-set (o test-and-set)*
  - Scambiare il contenuto di due parole *swap*

Protocollo di accesso alla sezione critica con disabilitazione degli interrupt

- **Acquisizione della risorsa**:
  - disabilito le interruzioni
  - leggo la variabile di lock
  - se la risorsa è libera (lock=0), la marco in uso ponendo lock=1 e riabilito le interruzioni
  - Altrimenti -- la risorsa è in uso (lock=1), riabilito le interruzioni e pongo il processo in attesa che la risorsa si liberi
- **Rilascio della risorsa**:
  - pongo lock=0

Protocollo di accesso alla sezione critica con istruzione hardware get-and-set

```
do {
while(get-and-set(&lock));
 sezione critica
 lock = 0;
 sezione non critica
}while(true);
```

- Istruzione atomica: get-and-set
  - Si dichiara una variabile globale lock, inizializzata a 0 (false)
  - Il processo legge la variabile di lock e la pone in un flag del processore
  - pone poi lock = 1 (true)
  - se il flag (= vecchio valore di lock) vale 0, la risorsa era libera, altrimenti era già occupata e il processo deve attendere
  - All'uscita della sezione critica, rilascia il lock, ponendo lock = 0 (false)

Protocollo di accesso alla sezione critica con istruzione hardware swap

- Istruzione atomica swap

lock: variabile globale
chiave: variabilie locale

```
do {

    chiave = true;
    while (chiave == true)
       Swap(&lock, &chiave);

        sezione critica

    lock = false;

        sezione non critica

} while (true);
```