



UNIVERSITÀ DEGLI STUDI DI PADOVA

Risorse ed eccezioni

Stefano Ghidoni



- Recap sull'acquisizione di risorse
- Gestione delle risorse in presenza di eccezioni
- Resource Acquisition Is Initialization (RAII)
- Garanzie



- In passato abbiamo analizzato la gestione delle risorse
- Es: memoria
 - Acquisizione
 - Utilizzo
 - Rilascio
- La gestione delle risorse può essere corrotta dalle eccezioni



- Un esempio di tale situazione:
- `std::vector` può accedere alla memoria in due modalità
 - Con boundary check – funzione `at()`
 - Senza boundary check – operatore `[]`
- Nel primo caso: se i vincoli non sono soddisfatti, che fare?
 - Eccezioni

Mettere un grande blocco di codice in un unico `try`, non garantisce che si prenda esattamente l'eccezione prevista, soprattutto se si fa un `catch` "universale"



Vector ed eccezioni

- È normale che `std::vector` lanci eccezioni
- In questo contesto, dobbiamo assicurarci che eventuali risorse occupate siano liberate
- Le risorse possono essere:
 - Memoria
 - Lock
 - File aperti
 - Thread
 - Socket
 - Window
 - ...



- Prendiamo un caso di acquisizione delle risorse:

```
void suspicious(int s, int x)
{
    int* p = new int[s];
    // ...
    delete[] p;
}
```

- Come possiamo essere sicuri che la memoria sia rilasciata? Cosa può esserci nei ...?

Se ci fosse un'eccezione in `// ...`, non è detto che venga effettivamente liberata la memoria



- È rilasciata in questo caso?

```
void suspicious(int s, int x)
{
    int* p = new int[s];
    // ...
    if (x) p = q;
    // ...
    delete[] p;
}
```

- Il valore di p è cambiato



- È rilasciata in questo caso?

```
void suspicious(int s, int x)
{
    int* p = new int[s];
    // ...
    if (x) return;
    // ...
    delete[] p;
}
```

- Il flusso di esecuzione non arriva alla delete[]



- È rilasciata in questo caso?

```
void suspicious(int s, int x)
{
    int* p = new int[s];
    vector<int> v;
    // ...
    if (x) p[x] = v.at(x);    // at(): accesso con verifica
    // ...                  // può lanciare eccezioni
    delete[] p;
}
```

- È lanciata un'eccezione