Errori di programmazione

Liceo G.B. Brocchi – Bassano del Grappa (VI)
Liceo Scientifico – opzione scienze applicate
Giovanni Mazzocchin

Dangling pointer

```
int* f() {
  int a;
  return &a;
}

int main() {
  *f() = 7;
}
```

```
bugs.cpp: In function ‘int* f()’:
bugs.cpp:5:10: warning: address of local variable ‘a’
returned [-Wreturn-local-addr]
    5 |   return &a;
      |          ^~
bugs.cpp:4:7: note: declared here
    4 |   int a;
      |       ^
```

```
cyofanni@LAPTOP-
I0S1RKRC:~/Desktop/live_programming$ ./bugs
Segmentation fault
```

f restituisce l’indirizzo di una sua variabile locale, che viene deallocata quando f restituisce il controllo al chiamante

Dangling pointer

```
int* f() {
  int ar[8];
  return ar;
}

int main() {
  f()[0] = 7;
}
```

gli array sono puntatori, di conseguenza non è possibile restituirli e poi utilizzarli in questo modo

Dereferenziazione di puntatore nullo

```
int main() {
  int* p = NULL;
  *p = 7;
}
```

compila, ma a runtime probabilmente darà `Segmentation fault`

Puntatore non inizializzato

```
int main() {
  int* p;
  *p = 7;
}
```

```
cyofanni@LAPTOP-I0S1RKRC:~/Desktop/live_programming$ ./bugs
Segmentation fault
```

Accesso oltre i limiti di un buffer

```
int main() {
  char str[8] = "abcdefg";
  for (int i = 0; i < 8192; i++) {
    cout << str[i];
  }
}
```

output imprevedibile e probabile crash. Non sicuro, stampa il contenuto della memoria adiacente a `str`

Errori di utilizzo della memoria heap

```
int main() {
  int* p = new int;
  delete p;
  delete p;
}
```

```
cyofanni@LAPTOP-I0S1RKRC:~/Desktop/live_programming$ ./bugs
free(): double free detected in tcache 2
Aborted
```

Errori di utilizzo della memoria heap

```
int main() {
   int var = 16;
   int* var_ptr = &var;
   delete var_ptr;
}
```

```
cyofanni@LAPTOP-I0S1RKRC:~/Desktop/live_programming$ g++ -o bugs bugs.cpp
bugs.cpp: In function ‘int main()’:
bugs.cpp:13:10: warning: ‘void operator delete(void*, std::size_t)’ called on unallocated object ‘var’ [-
Wfree-nonheap-object]
   13 |   delete var_ptr;
      |          ^~~~~~~
bugs.cpp:11:7: note: declared here
   11 |   int var = 16;
      |       ^~~

cyofanni@LAPTOP-I0S1RKRC:~/Desktop/live_programming$ ./bugs
free(): invalid pointer
Aborted
```

Errori di utilizzo della memoria heap

```
int main() {
  while (true) {
    int* ar = new int[4096];
  }
}
```

memory leak estremo: questo programma vuole prendersi tutta la memoria. Il sistema operativo si occuperà di killarlo

Considerazioni

- Programmando in C/C++ è molto facile commettere alcuni degli errori visti sopra
- Un programma che contiene anche solo uno di questi errori è pericoloso, per cui dovrebbe crashare il prima possibile
- Se non crasha quando viene testato, probabilmente andrà in crash quando il software sarà in produzione, che è molto peggio
- I sistemi operativi moderni sono in grado di rilevare comportamenti anomali dei programmi, per cui i test che farete porteranno quasi sicuramente al crash del programma