

Lập trình C++ 3

Achieving timely quality

Lê Quốc Tuấn

Optimal seminar group

Tối ưu biên dịch

(Compiler optimization)

Run-time and compile-time là những nhân tố (thứ hai) cần được cải thiện để nâng cao chất lượng chương trình (sản phẩm) về mặt thời gian.

Tối ưu biên dịch được cung cấp dưới dạng tùy chọn đi kèm với trình biên dịch.

- Compiler của Intel sẽ khác với của GNU hay Oracle ...
- Tăng thêm thời gian biên dịch và dung lượng bộ nhớ.
- Có nhiều mức độ tối ưu, cân bằng giữa compile-time và run-time.
- Các tùy chọn riêng (cũng nằm trong các mức tối ưu) có thể gọi riêng lẻ.

Các mức tối ưu:

- Thông dụng: -O, -O1, -O2, -O3.
- Ngoài ra: -O0, -Os, -Og, -Ofast, -Og.

Cú pháp áp dụng:

```
g++ -Ox -o output input.cpp
```

itemize trong đó, x là mức tối ưu.

Listing 1: Đo thời gian giữa 2 thời điểm trong C++

```
1 #include <iostream>
2 #include <ctime>    //for clock time
3 using namespace std;
4
5 int main(){
6     //begin clock time
7     const clock_t begin_time = clock();
8
9     //do something
10
11     cout << float(clock() - begin_time)
12     /CLOCKS_PER_SEC << endl;
13
14     return 0;
15 }
```

Listing 2: C++ compiler optimization sample

```
1 #include <iostream>
2 #include <ctime>    //for clock time
3 using namespace std;
4
5 int main(){
6     //begin clock time
7     const clock_t begin_time = clock();
8     double number=1e+06;
9     double sum=0;
10
11     for(int i=0; i<number; ++i){
12         sum+=i;
13     }
14     cout << float(clock() - begin_time)
15     /CLOCKS_PER_SEC *1e+06 << endl;
16
17     return 0;
18 }
```

Listing 3: Code bash thực hiện chạy lặp chương trình

```
1 #!/bin/bash
2 read loopNumber
3 for ((i=0; i<loopNumber; ++i));
4     do
5         ./optimize_loop
6 done
```

Listing 4: Code python thực hiện tính thời gian trung bình

```
1 openFile = open("data.txt", 'r')
2 lines = openFile.readlines()    #reading lines here
3 sumLines = 0    #sum update each time number is entered
4 linesNumber = sum(1 for line in open("data.txt", 'r'))
5 for line in lines:    #taking each line
6     lineToInt = float(line)    #converting string to float
7     sumLines += lineToInt    #update sumLine
8 print(sumLines/linesNumber)
```

Sử dụng `time` để tính thời `compile-time`:

```
time g++ -Ox -o output input.cpp
```

Kết quả trả về có dạng, chẳng hạn:

```
real 0m0,310s
user 0m0,265s
sys 0m0,044s
```

Trong đó,

real	Elapsed real (wall clock) time used by the process, in seconds.
user	Total number of CPU-seconds that the process used directly (in user mode), in seconds.
sys	Total number of CPU-seconds used by the system on behalf of the process (in kernel mode), in seconds.

real 0m0,314s
user 0m0,280s
sys 0m0,034s

real 0m0,326s
user 0m0,297s
sys 0m0,029s

Df O1

O3 O2

real 0m0,339s
user 0m0,295s
sys 0m0,043s

real 0m0,332s
user 0m0,279s
sys 0m0,053s