

DT019G Objektbaserad programmering i C++

Laboration: Dynamisk minneshantering med `new` och `delete`.

Martin Kjellqvist, Jimmy Åhlander

memory.tex 205 2017-03-06 15:57:39Z jimahl

Innehåll

1	Introduktion	1
2	Syfte	1
3	Uppgift	1
4	Examination	2

1 Introduktion

Dynamisk minneshantering kan orsaka minnesläckage. Minnesläckage kan orsaka problem för körande program.

2 Syfte

Vi kommer i denna lab att undersöka vad som händer då program uppför sig illa.

De nya begrepp som används är följande nyckelord `new`, `delete`.

Du kommer att tvingas hänvisa till ett exakt antal bytes. Det gör du enklast i C++ genom att använda typer vars storlek är specificerade [1]. Typerna `int8_t` och `int16_t` är lämpliga kandidater. Alternativt kan du använda operatören `sizeof` [2] med valfri datatyp, förslagsvis `int`.

Vissa C++-kompilatorer använde tidigare ett returvärde `NULL` från `new`, det används inte längre. Det korrekta skrivsättet visas nedan.

3 Uppgift

Det program du kommer att skapa kan mycket väl göra ditt system tillfälligt icke-responsivt. Var beredd med task-manager eller kill om du vill döda processen.

Formulera en konsollapplikation som hjälper dig att undersöka hur din dator beter sig då en applikation allokerar väldigt mycket minne som den inte ger tillbaka.

Exakt hur många byte är 1 KiB?

1. Allokera minne i stycken om 1 GiB.
2. Allokera minne i stycken om 1 KiB.

Variera målplattform. Använd följande konfigurationer

1. 32-bitars målplattform.
2. 32-bitars målplattform, *Large Address Aware*.
3. 64-bitars målplattform.
4. 64-bitars målplattform, *Large Address Aware*.

```
int main(){
    try{
// Skriv new-satserna och annan relaterad programkod här.

    }catch( std::bad_alloc e ){
        cerr<< e.what()<<endl; // Körs denna rad?
    }
    return 0;
}
```

4 Examination

Din lösning ska redovisas muntligen för en lärare vid något av kursens redovisningstillfällen. När du redovisat och fått godkänt laddar du upp din källkod (och byggsript) till inlämningslådan i lärplattformen.

Skriv ett dokument som beskriver hur ditt system uppför sig i de olika fallen.

Referenser

- [1] *cppreference 'integer types'*. cppreference, . URL <http://en.cppreference.com/w/cpp/types/integer>.
- [2] *cppreference 'sizeof operator'*. cppreference, . URL <http://en.cppreference.com/w/cpp/language/sizeof>.