

Memory Management

<memory>



Wycieki Pamięci

Wyciek pamięci (ang. **memory leak**) – szczególny rodzaj niezamierzonego użycia pamięci przez program komputerowy, gdy nie zwalnia on zaalokowanej wcześniej pamięci, która nie jest już mu potrzebna, a może nawet rezerwować nową. Wycieki pamięci są efektem bardzo niepożądanym.

Valgrind

Uruchomienie aplikacji w valgrind powinno wyglądać tak:

```
1 valgrind [opcje valgrinda] /sciezka/do/programu parametry startowe programu
```

Opcje programu valgrind możemy sprawdzić korzystając z polecenia:

```
1 valgrind --help
```

Do uzyskania pełnej informacji o miejscach wycieków należy wykorzystać opcję `--leak-check=full`.

Unique_ptr

- `unique_ptr<int> uptr = make_unique<int>(value);`
- `auto uptr2 = move(uptr);`
- Wskaźnika `unique_ptr` nie można kopiować, jedyną możliwością aby go przekazać jest `std::move`.
- `Unique_ptr` jest jedynym odniesieniem do zaalokowanej pamięci. Pamięć zostanie zwolniona w momencie zniszczenia `unique_ptr`.
- `reset()`, `get()`, `release()`

Shared_ptr

- `shared_ptr<int> sptr = make_shared<int>(value);`
- `auto sptr2 = sptr;`
- Wskaźnik można kopiować, a kopie wskaźników utrzymują między sobą licznik referencji (++ make, -- remove). Pamięć zostanie zwolniona z ostatnim usuniętym wskaźnikiem.
- „Ostatni gasi światło”
- `reset()`, `get()`, `use_count()`

Weak_ptr

- `weak_ptr<int> wptr = sptr;`
- Używamy do śledzenia obiektu na który wskazuje `shared_ptr`.
- Tworzony na podstawie `shared_ptr`.
- Gdy nie wskazuje na nic zwraca 0.
- Pamięć zostanie zwolniona w momencie usunięcia ostatniego `shared_ptr`.
- `lock()`, `reset()`, `use_count()`

Źródła

- <https://en.cppreference.com/w/cpp/memory>
- <https://kacperkolodziej.pl/programowanie/zwalczanie-wyciekow-bledow-pamieci-cpp11.html>