

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector

Inicjalizacja i niszczenie obiektów w wybranym języku programowania.

Patryk Lisik

Uniwersytet Łódzki

2023/24

Agenda

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odświeżanie
Garbage
Collector

- Pojęcia
- Inicjalizacja w językach systemowych.
 - Stos (stack)
 - Sarta (heap)
 - Problem fragmentacji
 - Alokacja pamięci
- Inicjalizacja w językach obiektowych
- Bezpieczeństwo pamięci w językach bez GC
 - Zliczanie referencji(shared_ptr)
 - Przekazywanie własności(unique_ptr)
- Problem cyklicznych referencji
- Garbage collectory
 - Taksonomia
 - Działanie Mark/Sweep/Compact

Co to typ?

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector

Typ

Dane i zbiór operacji, które można na nich wykonać [1].

```
class LineFunction {  
    double a, b; //dane  
  
    // operacja  
    double operator()(double x) const{  
        return a * x + b;  
    }  
};
```

Rysunek 1: Przykład własnego typu w C++

Co to obiekt?

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odświeżanie
Garbage
Collector

Obiekt

Ukonkretnienie typu w postaci zaalokowanej pamięci.

```
LineFunction func{4.5, 2.1};  
double d = func(7.2);
```

Rola systemu operacyjnego w przydziale pamięci

Cykl życia obiektów

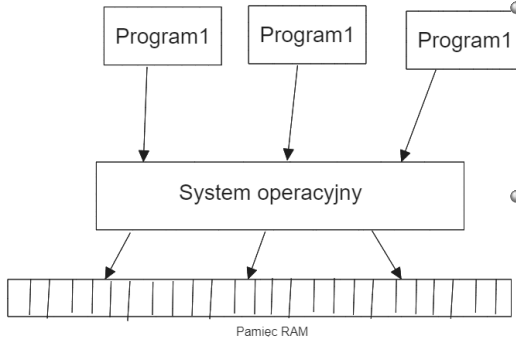
Patryk Lisik

Pojęcia

Inicjalizacja w językach systemowych

Automatyczne zarządzanie pamięcią w C++

Odświeżanie Garbage Collector



Rysunek 2: Układ pamięci programu

- Program/Process ma dostęp tylko do pamięci przydzielonej przez OS.
- Odczyt z nie dozwolonej pamięci kończy się wyjątkiem segmentation fault core dumped

Pamięć programu

Cykl życia
obiektów

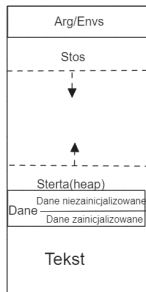
Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector



Rysunek 3: Układ pamięci programu

Argumenty	*argc, zmienne środowiskowe
Stos	Zmienne lokalne programu. ok 1mb
Sarta	Dane alokowane dynamicznie
Dane	Zmienne globalne, vtable,
Tekst	Kod programu

Alokacja na stosie

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector

```
long foo(){  
    short d = 7;  
    int c = 3  
    return d*c;  
}  
int main(){  
    int a = 11;  
    long b = foo();  
    return 0;  
}
```

Szybkość

Rozmiar

Dane

Kolejna zmienna jest
zawsze alokowana na
wskaźniku stosu +1

Rozmiar stosu jest stały.
Przy jego przekroczeniu
występuje wyjątek stack
overflow.

Rozmiar danych
przechowywanych na stosie
musi być znany w czasie
kompilacji.

Dynamiczna alokacja na stosie?

Cykl życia
obiektów

Patryk Lisik

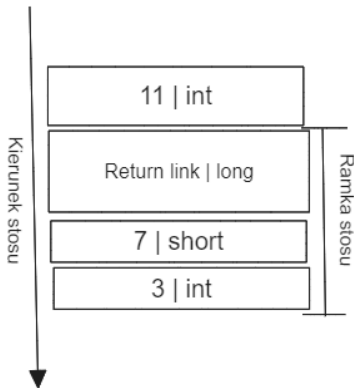
Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odświeżanie
Garbage
Collector

```
long foo(){  
    short d = 7;  
    int c = 3  
    return d*c;  
}  
  
int main(){  
    int a = 11;  
    long b = foo();  
    return 0;  
}
```



Dynamiczna alokacja na stosie jest niemożliwa, ponieważ elementy nadpisywały by swoje wartości.

Alokacja na sterpie

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odświeżanie
Garbage
Collector

```
long foo(){  
    short d = 7;  
    int c = 3  
    return d*c;  
}  
int main(){  
    int a = 11;  
    long b = foo();  
    return 0;  
}
```

Szybkość

Kolejna zmienna jest
zawsze alokowana na
wskaźniku stosu +1

Ograniczony rozmiar Rozmiar stosu jest
stały. Przy jego
przekroczeniu występuje
wyjątek stack overflow.

Działanie malloc/new

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector

Alokacja pamięci

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector

Alokacja pamięci

Alokacja pamięci to rezerwacja przez system operacyjny pamięci dla procesu [2, 3].

(Nie rozpatrujemy sytuacji gdy nie ma systemu operacyjnego)

POSIX

W systemach zgodnych z POSIX alokacja pamięci odbywa się poprzez 'mmap()' [2].

Windows

W systemach Windows alokacja pamięci odbywa się poprzez VirtualAlloc[memoryapi.h] [3].

Bezpieczeństwo pamięci w językach bez GC

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector

Najpierw wywoływany jest konstruktor rodzica.

Zliczanie referencji

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector

`std::shared_ptr` składa się z licznika referencji i właściwego wskaźnika. Podczas kopi licznik referencji jest zwiększany, a podczas wywołania destruktora zmniejszany. Przy wywołaniu ostatniego destruktora pamięć jest zwalniana [4].

Uniqe_ptr

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector

Własność w programowaniu

Własność określa kto jest odpowiedzialny za zwolnienie pamięci [5, 6].

std::move informuje kompilator o możliwości przeniesienie zasobów już istniejącego obiektu.

Problem cyklicznych referencji

Cykl życia
obiektów

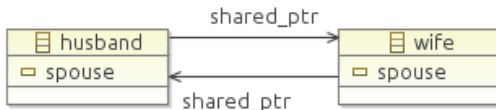
Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector



Rysunek 4: Problem cyklicznych referencji gdzie pamięć nigdy nie jest zwalniana. Źródło: [7]

`std::weak_ptr`

Rozwiązaniem problemu cyklicznych referencji jest referencja słaba(`std::weak_ptr`) która nie powoduje inkrementacji licznika referencji. Nie gwarantuje one że obiekt nie został już usunięty [7, 8].

Oznacz i zmieć (ang. *mark and sweep*)

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector

Faza mark

Przeszukujemy "w głąb" graf obiektów i oznaczamy każdy obiekt do którego dotarliśmy [9].

Faza sweep

Usuujemy obiekty nieoznaczone.

Faza compact(Opcjonalna)

Obiekty przesuwane są blisko siebie w pamięci.

Krytyka

- Jest to bardzo wolny algorytm, który wymaga zatrzymania przetwarzania do działania (ang. *stop the world GC*).
- Bez fazy compact powoduje dużą fragmentację pamięci.

Inne rodzaje odśmiecaczy/garbage collectorów

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmiecanie
Garbage
Collector

Generacyjny Trzykolorowy Arenowy

Literatura I

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector



“lbn.com – type definitions.”

<https://www.ibm.com/docs/en/epfz/5.3?topic=reference-type-definitions>, (dostęp 20-10-2023).



“mmap(2) — linux manual page.” <https://man7.org/linux/man-pages/man2/mmap.2.html>,

(dostęp 31-10-2023).



“Virtualalloc function (memoryapi.h).”

<https://learn.microsoft.com/en-gb/windows/win32/api/memoryapi/nf-memoryapi-virtualalloc?redirectedfrom=MSDN>, (dostęp 31-10-2023).



“Cpp reference - std::move.”

<https://en.cppreference.com/w/cpp/utility/move>, (dostęp 31-10-2023).

Literatura II

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector



“The rust programming language sekcja [what is ownership?].” <https://doc.rust-lang.org/book/ch04-01-what-is-ownership.html#what-is-ownership>, (dostęp 31-10-2023).



“Cpp reference - dynamic memory managemnt - std::unique_ptr.” https://en.cppreference.com/w/cpp/memory/unique_ptr, (dostęp 31-10-2023).



“Weak pointers and circular references in c++ 11.” <https://visualstudiomagazine.com/articles/2012/10/19/circular-references.aspx>, (dostęp 31-10-2023).

Literatura III

Cykl życia
obiektów

Patryk Lisik

Pojęcia

Inicjalizacja w
językach
systemowych

Automatyczne
zarządzanie
pamięcią w
C++

Odśmianie
Garbage
Collector



“Cpp reference - dynamic memory management -
std::weak_ptr.” [https:](https://en.cppreference.com/w/cpp/memory/weak_ptr)

[//en.cppreference.com/w/cpp/memory/weak_ptr,](https://en.cppreference.com/w/cpp/memory/weak_ptr)
(dostęp 31-10-2023).



“Cs330 - mark and sweep.” [https://www.cs.odu.edu/](https://www.cs.odu.edu/~zeil/cs330/f13/Public/garbageCollection/garbageCollection-htmlsu5.html)
[~zeil/cs330/f13/Public/garbageCollection/](https://www.cs.odu.edu/~zeil/cs330/f13/Public/garbageCollection/garbageCollection-htmlsu5.html)
[garbageCollection-htmlsu5.html,](https://www.cs.odu.edu/~zeil/cs330/f13/Public/garbageCollection/garbageCollection-htmlsu5.html) (dostęp
31-10-2023).