Async-profiler, Perf, FlameGraph
trio do profilowania produkcji

Krzysztof Ślusarski

Krótko o mnie

- Programuję od 1992

Krótko o mnie

- Programuję od 1992

- Zawodowo:
  - Od 10.2006
    Java Programmer /
    Team Leader /
    System Architect /
    Solution Architect

Krótko o mnie

- Poza 8h 5/7:

Krótko o mnie

- Poza 8h 5/7:
  - Szkolenia
    - Bebechy JVM
    - Tuning JVM
    - Wycieki pamięci
    - Profiling

Krótko o mnie

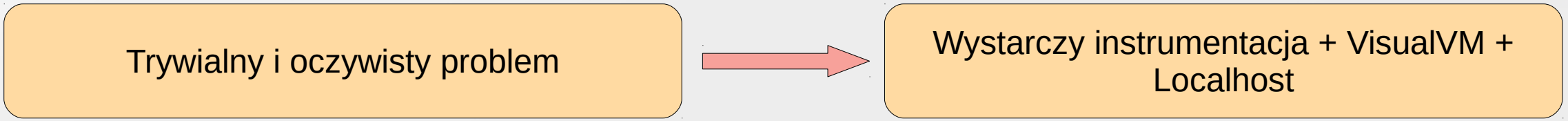
- Poza 8h 5/7:
  - Szkolenia
    - Bebechy JVM
    - Tuning JVM
    - Wycieki pamięci
    - Profiling
  - Diagnoza awarii produkcyjnych + Profiling

Czemu na produkcji?

Czemu na produkcji?

Trywialny i oczywisty problem

Czemu na produkcji?



Czemu na produkcji?

Czy:

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?
- Ta sama dystrybucja OS?

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?
- Ta sama dystrybucja OS?
- Ta sama wersja jądra?

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?
- Ta sama dystrybucja OS?
- Ta sama wersja jądra?
- Ten sam sprzęt (+ wirtualizacja)?

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?
- Ta sama dystrybucja OS?
- Ta sama wersja jądra?
- Ten sam sprzęt (+ wirtualizacja)?
- Tak samo wyskalowany?

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?
- Ta sama dystrybucja OS?
- Ta sama wersja jądra?
- Ten sam sprzęt (+ wirtualizacja)?
- Tak samo wyskalowany?
- Z takim samym ruchem?

Czemu na produkcji?

Czy:

	DEV
Ta sama dystrybucja JDK?	Tak
Ta sama wersja co do update-u?	Nie
Ten sam OS?	Nie
Ta sama dystrybucja OS?	Nie
Ta sama wersja jądra?	Nie
Ten sam sprzęt (+ wirtualizacja)?	Nie
Tak samo wyskalowany?	Nie
Z takim samym ruchem?	Nie

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?
- Ta sama dystrybucja OS?
- Ta sama wersja jądra?
- Ten sam sprzęt (+ wirtualizacja)?
- Tak samo wyskalowany?
- Z takim samym ruchem?

DEV	DEV 2
Tak	Tak
Nie	Tak
Nie	Tak
Nie	Nie
Nie	Nie
Nie	Nie
Nie	Nie
Nie	Nie

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?
- Ta sama dystrybucja OS?
- Ta sama wersja jądra?
- Ten sam sprzęt (+ wirtualizacja)?
- Tak samo wyskalowany?
- Z takim samym ruchem?

DEV	DEV 2	UAT
Tak	Tak	Tak
Nie	Tak	Tak
Nie	Tak	Tak
Nie	Nie	Tak
Nie	Nie	Tak
Nie	Nie	Tak
Nie	Nie	Nie
Nie	Nie	Nie

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?
- Ta sama dystrybucja OS?
- Ta sama wersja jądra?
- Ten sam sprzęt (+ wirtualizacja)?
- Tak samo wyskalowany?
- Z takim samym ruchem?

DEV	DEV 2	UAT	UAT 2
Tak	Tak	Tak	Tak
Nie	Tak	Tak	Tak
Nie	Tak	Tak	Tak
Nie	Nie	Tak	Tak
Nie	Nie	Tak	Tak
Nie	Nie	Tak	Tak
Nie	Nie	Nie	Tak
Nie	Nie	Nie	Nie

Czemu na produkcji?

Czy:

- Ta sama dystrybucja JDK?
- Ta sama wersja co do update-u?
- Ten sam OS?
- Ta sama dystrybucja OS?
- Ta sama wersja jądra?
- Ten sam sprzęt (+ wirtualizacja)?
- Tak samo wyskalowany?
- Z takim samym ruchem?

DEV	DEV 2	UAT	UAT 2	PROD
Tak	Tak	Tak	Tak	Tak
Nie	Tak	Tak	Tak	Tak
Nie	Tak	Tak	Tak	Tak
Nie	Nie	Tak	Tak	Tak
Nie	Nie	Tak	Tak	Tak
Nie	Nie	Tak	Tak	Tak
Nie	Nie	Nie	Tak	Tak
Nie	Nie	Nie	Nie	Tak

Collapsed stack

- Format wyjściowy profilera

Collapsed stack

- Format wyjściowy profilera
- Format tekstowy

Collapsed stack

- Format wyjściowy profilera
- Format tekstowy
- Bardzo łatwe parsowanie

Collapsed stack

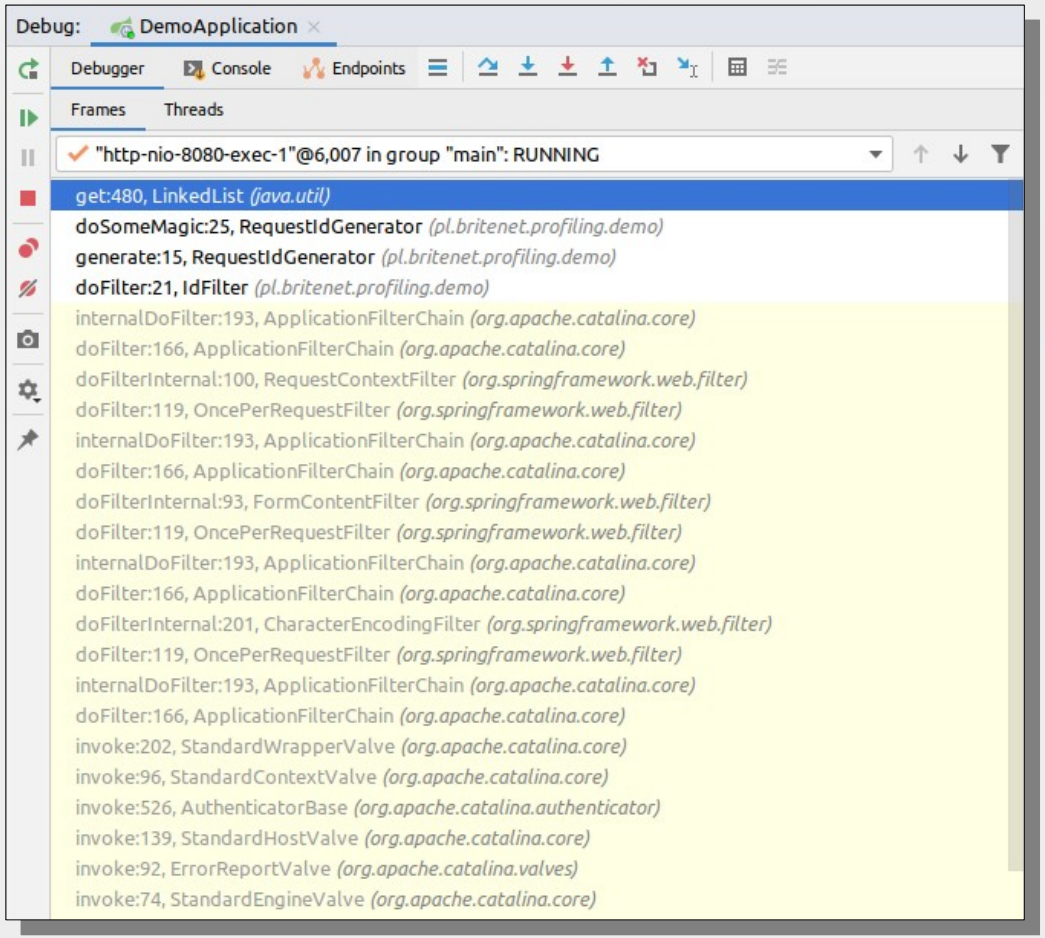
- Format wyjściowy profilera
- Format tekstowy
- Bardzo łatwe parsowanie
- Bardzo łatwe transformacje pliku

Collapsed stack

```
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'jitte
    at org.springframework.beans.factory.annotation.InitDestroyAnnotationBeanPostProcessor.pos
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyBean
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initialize
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateB
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean
    at org.springframework.beans.factory.support.AbstractBeanFactory.lambda$doGetBean$0(Abstr
    at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(De
    at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFa
    at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFact
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSin
    at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitia
    at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplica
    at org.springframework.boot.web.servlet.context.ServletWebServerApplicationContext.refres
    at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:747) ~[sprin
    at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:397)
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:315) ~[spring-bo
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:1226) ~[spring-b
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:1215) ~[spring-b
    at pl.britenet.profiling.demo.DemoApplication.main(DemoApplication.java:13) ~[classes/:na
Caused by: java.lang.NullPointerException: null
    at pl.britenet.profiling.demo.RequestIdGenerator.doSomeMagic(RequestIdGenerator.java:27)
    at pl.britenet.profiling.demo.RequestIdGenerator.generate(RequestIdGenerator.java:17) ~[c
    at pl.britenet.profiling.demo.Jitter.jitIt(Jitter.java:33) ~[classes/:na] <4 internal cal
    at org.springframework.beans.factory.annotation.InitDestroyAnnotationBeanPostProcessor$Li
    at org.springframework.beans.factory.annotation.InitDestroyAnnotationBeanPostProcessor$Li
    at org.springframework.beans.factory.annotation.InitDestroyAnnotationBeanPostProcessor.pos
    ... 18 common frames omitted
```

Collapsed stack

```
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'jitte
    at org.springframework.beans.factory.annotation.InitDestroyAnnotationBeanPostProcessor.pos
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyBeanI
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initialize
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBe
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean
    at org.springframework.beans.factory.support.AbstractBeanFactory.lambda$doGetBean$0(Abstra
    at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(De
    at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFac
    at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFact
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSin
    at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitia
    at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplica
    at org.springframework.boot.web.servlet.context.ServletWebServerApplicationContext.refres
    at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:747) ~[sprin
    at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:397)
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:315) ~[spring-bo
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:1226) ~[spring-b
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:1215) ~[spring-b
    at pl.britenet.profiling.demo.DemoApplication.main(DemoApplication.java:13) ~[classes/:na
Caused by: java.lang.NullPointerException: null
    at pl.britenet.profiling.demo.RequestIdGenerator.doSomeMagic(RequestIdGenerator.java:27)
    at pl.britenet.profiling.demo.RequestIdGenerator.generate(RequestIdGenerator.java:17) ~[c
    at pl.britenet.profiling.demo.Jitter.jitIt(Jitter.java:33) ~[classes/:na] <4 internal cal
    at org.springframework.beans.factory.annotation.InitDestroyAnnotationBeanPostProcessor$Li
    at org.springframework.beans.factory.annotation.InitDestroyAnnotationBeanPostProcessor$Li
    at org.springframework.beans.factory.annotation.InitDestroyAnnotationBeanPostProcessor.pos
    ... 18 common frames omitted
```



Collapsed stack

```
java/lang/Thread.run;org/apache/tomcat/util/threads/TaskThread$WrappingRunnable.run;java/util/concurrent/
ThreadPoolExecutor$Worker.run;java/util/concurrent/ThreadPoolExecutor.runWorker;org/apache/tomcat/util/
net/SocketProcessorBase.run;org/apache/tomcat/util/net/NioEndpoint$SocketProcessor.doRun;org/apache/
coyote/AbstractProtocol$ConnectionHandler.process;org/apache/coyote/AbstractProcessorLight.process;org/
apache/coyote/http11/Http11Processor.service;org/apache/catalina/connector/CoyoteAdapter.service;org/
apache/catalina/core/StandardEngineValve.invoke;org/apache/catalina/valves/ErrorReportValve.invoke;org/
apache/catalina/core/StandardHostValve.invoke;org/apache/catalina/authenticator/
AuthenticatorBase.invoke;org/apache/catalina/core/StandardContextValve.invoke;org/apache/catalina/core/
StandardWrapperValve.invoke;org/apache/catalina/core/ApplicationFilterChain.doFilter;org/apache/catalina/
core/ApplicationFilterChain.internalDoFilter;org/springframework/web/filter/
OncePerRequestFilter.doFilter;org/springframework/web/filter/
CharacterEncodingFilter.doFilterInternal;org/apache/catalina/core/ApplicationFilterChain.doFilter;org/
apache/catalina/core/ApplicationFilterChain.internalDoFilter;org/springframework/web/filter/
OncePerRequestFilter.doFilter;org/springframework/web/filter/FormContentFilter.doFilterInternal;org/
apache/catalina/core/ApplicationFilterChain.doFilter;org/apache/catalina/core/
ApplicationFilterChain.internalDoFilter;org/springframework/web/filter/OncePerRequestFilter.doFilter;org/
springframework/web/filter/RequestContextFilter.doFilterInternal;org/apache/catalina/core/
ApplicationFilterChain.doFilter;org/apache/catalina/core/ApplicationFilterChain.internalDoFilter;pl/
britenet/profiling/demo/IdFilter.doFilter;pl/britenet/profiling/demo/RequestIdGenerator.generate;pl/
britenet/profiling/demo/RequestIdGenerator.doSomeMagic;java/util/LinkedList.get  2
```

Collapsed stack – parser

```
String line = reader.readLine();
int delimiterChar = line.lastIndexOf(" ");
String stack = line.substring(0, delimiterChar);
long count = Long.parseLong(line.substring(delimiterChar + 1));
String[] splittedStack = stack.split(";");
```

Collapsed stack

```
a;h 3
b;d;e;f 5
b;d;e;g 2
b;d 2
c 2
```

Collapsed stack – self time

```
a;h 3
b;d;e;f 5
b;d;e;g 2
b;d 2
c 2
```

Self time (d) = 2/14

Collapsed stack – total time

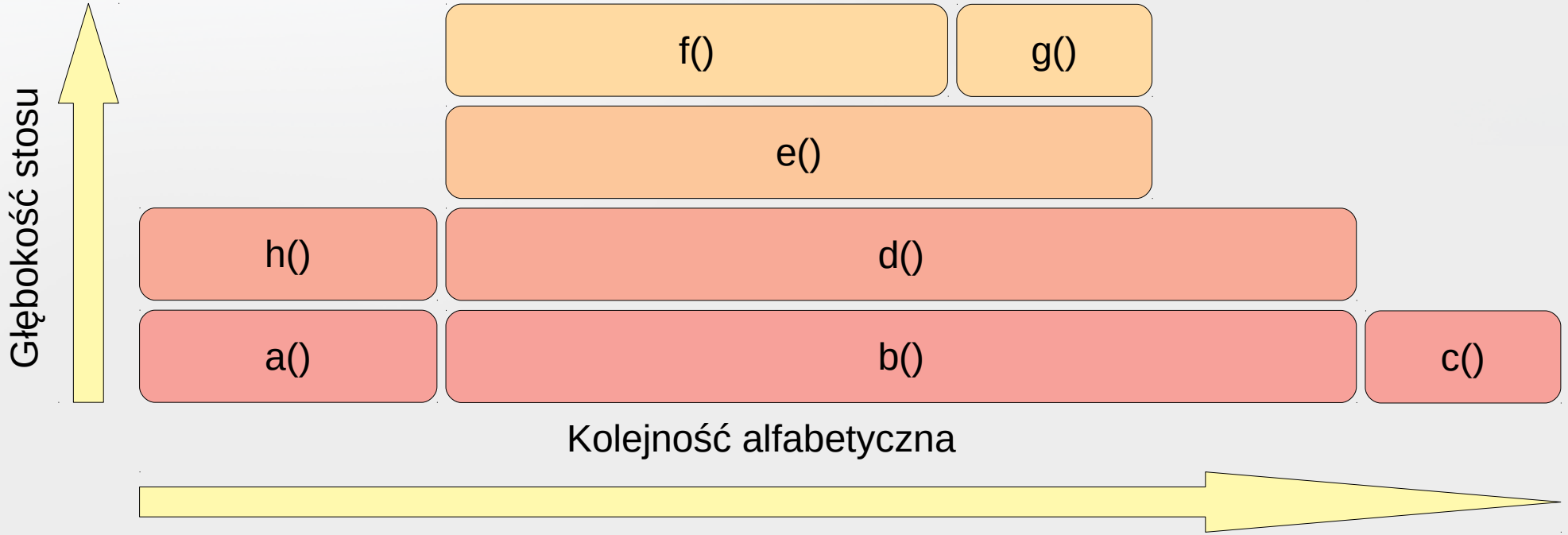
```
a;h 3
b;d;e;f 5
b;d;e;g 2
b;d 2
c 2


Total time (d) == 9/14
```

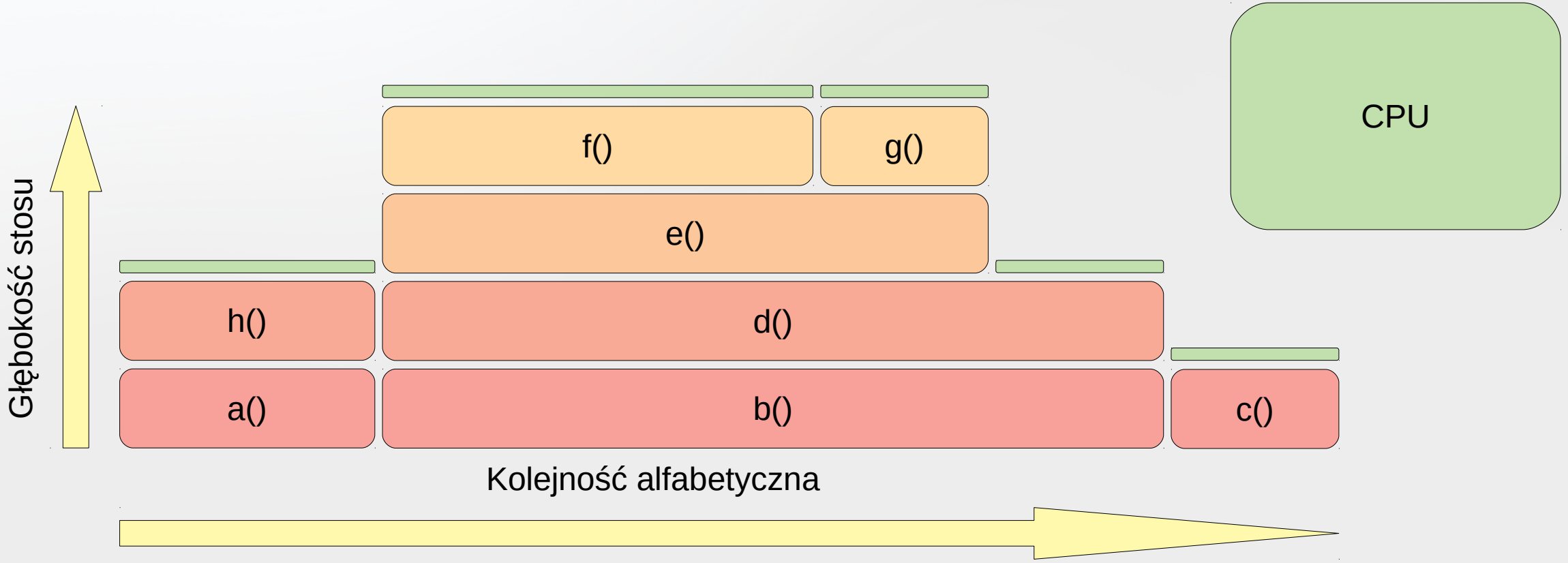
FlameGraph


f() f() f() f() f() g() g()
e() e() e() e() e() e() e()
h() h() h() d() d() d() d() d() d() d() d() d()
a() a() a() b() b() b() b() b() b() b() b() b() c() c()
Głębokość stosu
Kolejność alfabetyczna

FlameGraph



FlameGraph



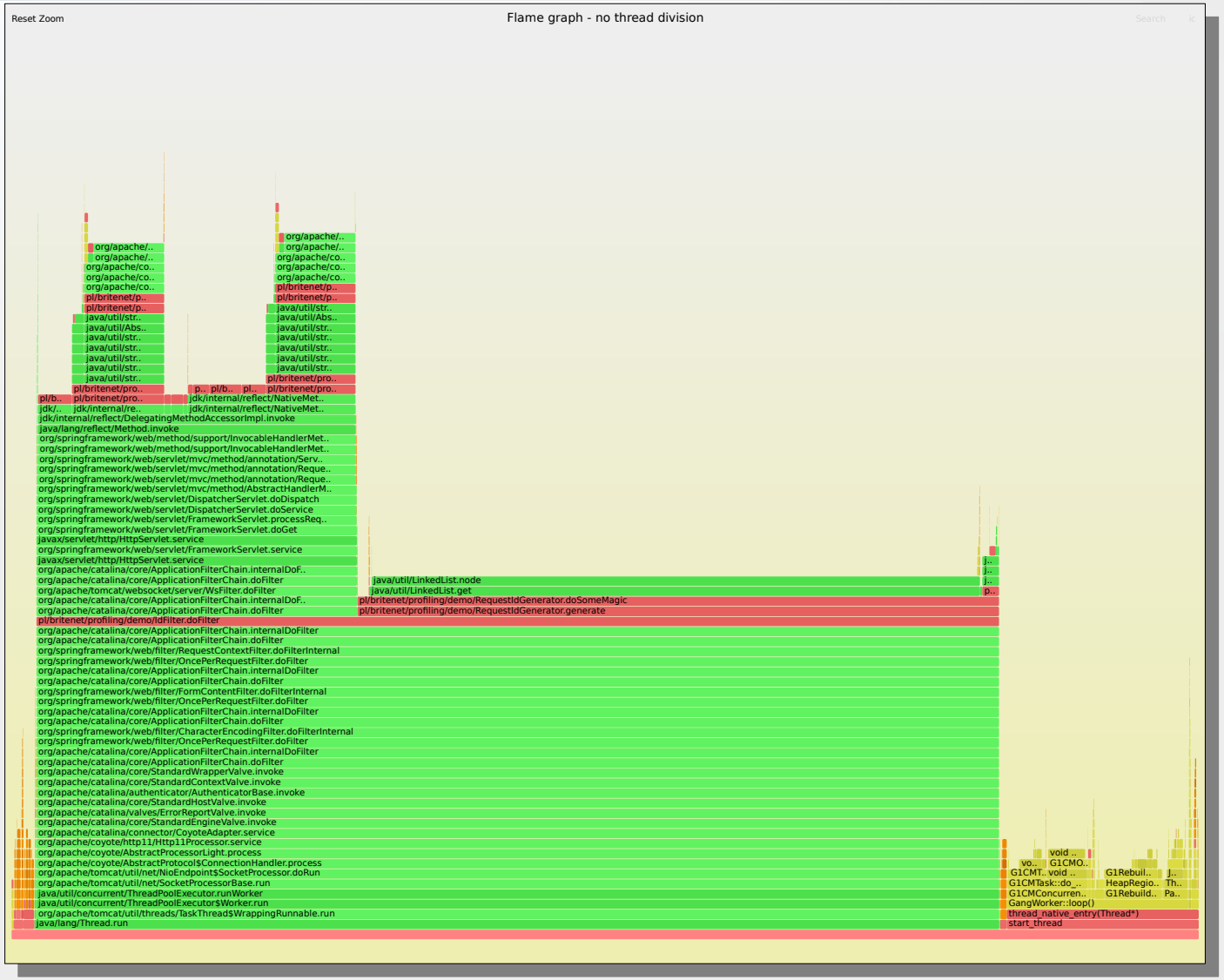
FlameGraph

Java

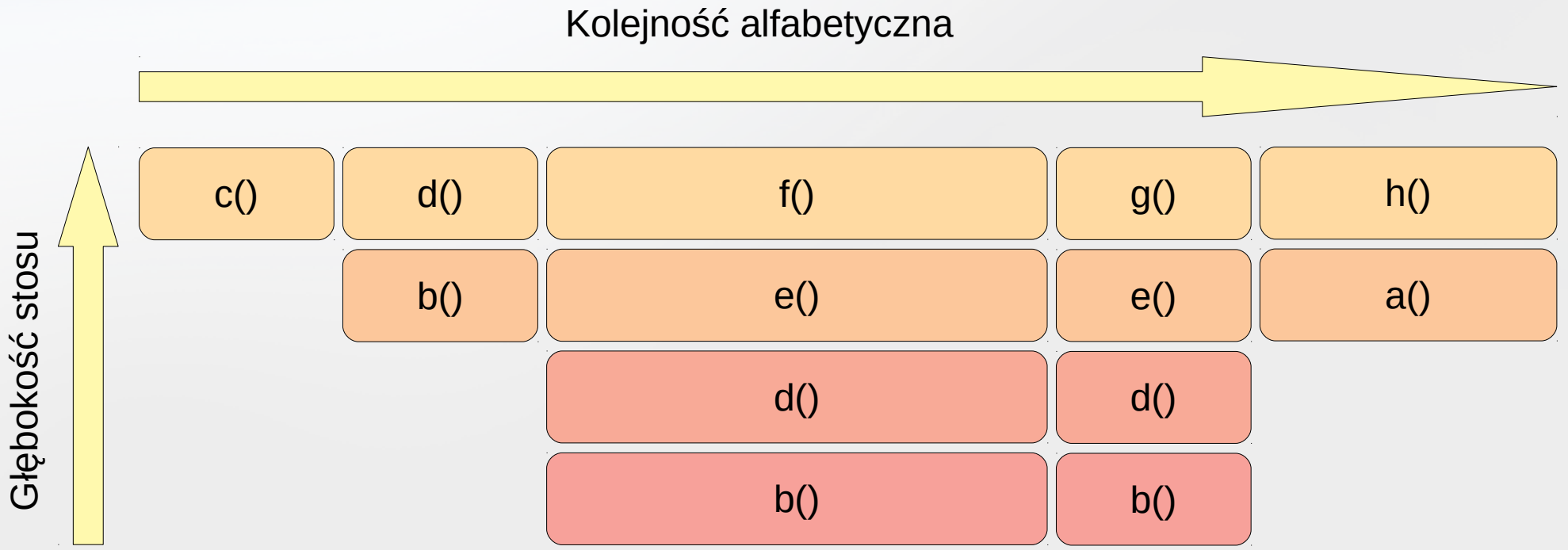
JVM

Kernel

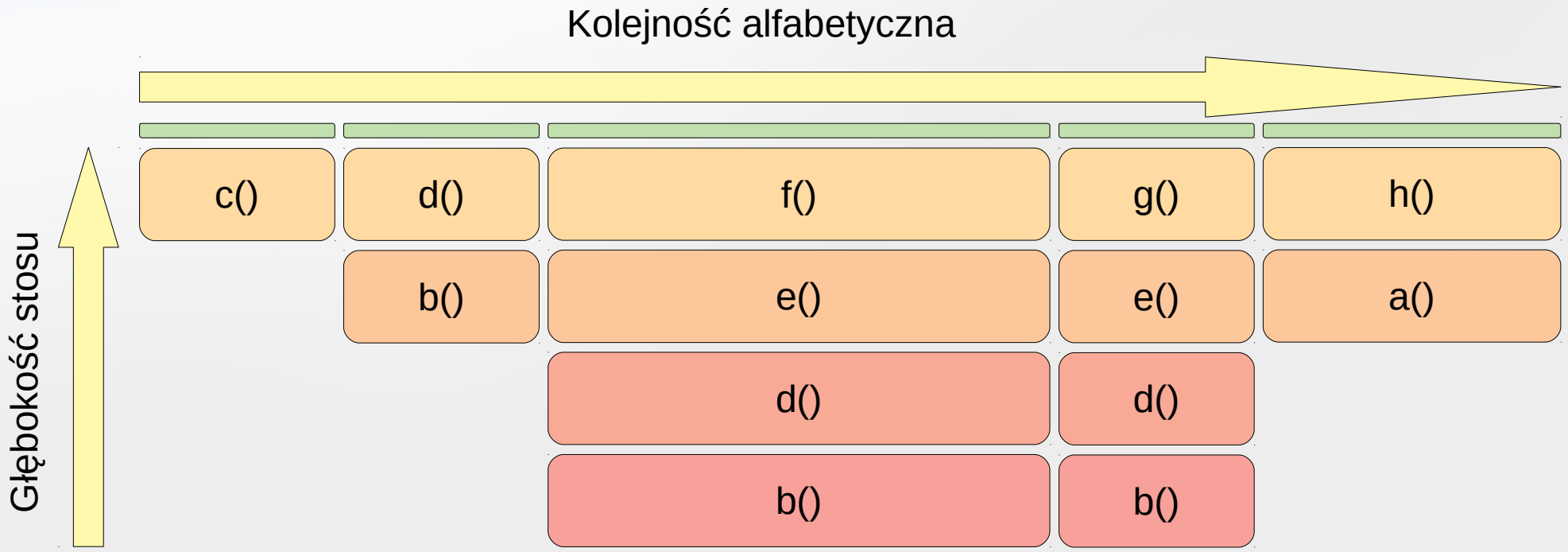
Other – user space



FlameGraph – reversed & inverted



FlameGraph – reversed & inverted

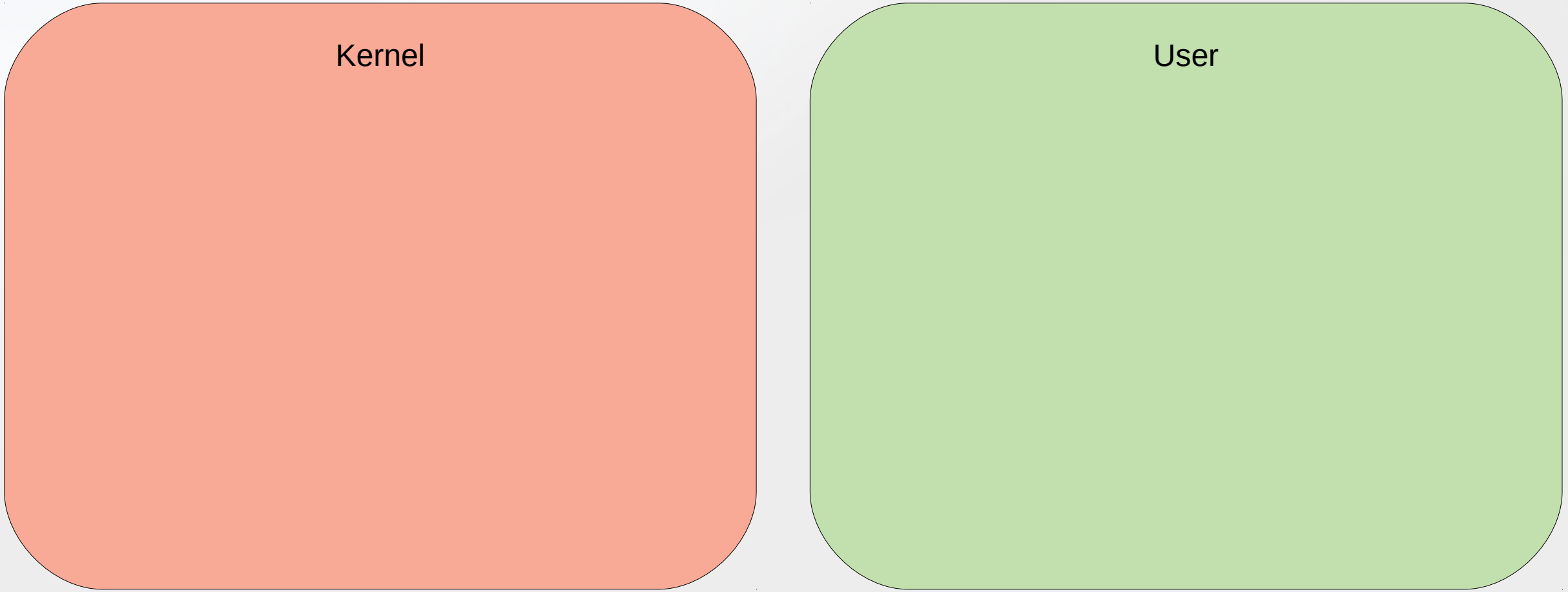


Collapsed stack + FlameGraph - przykład

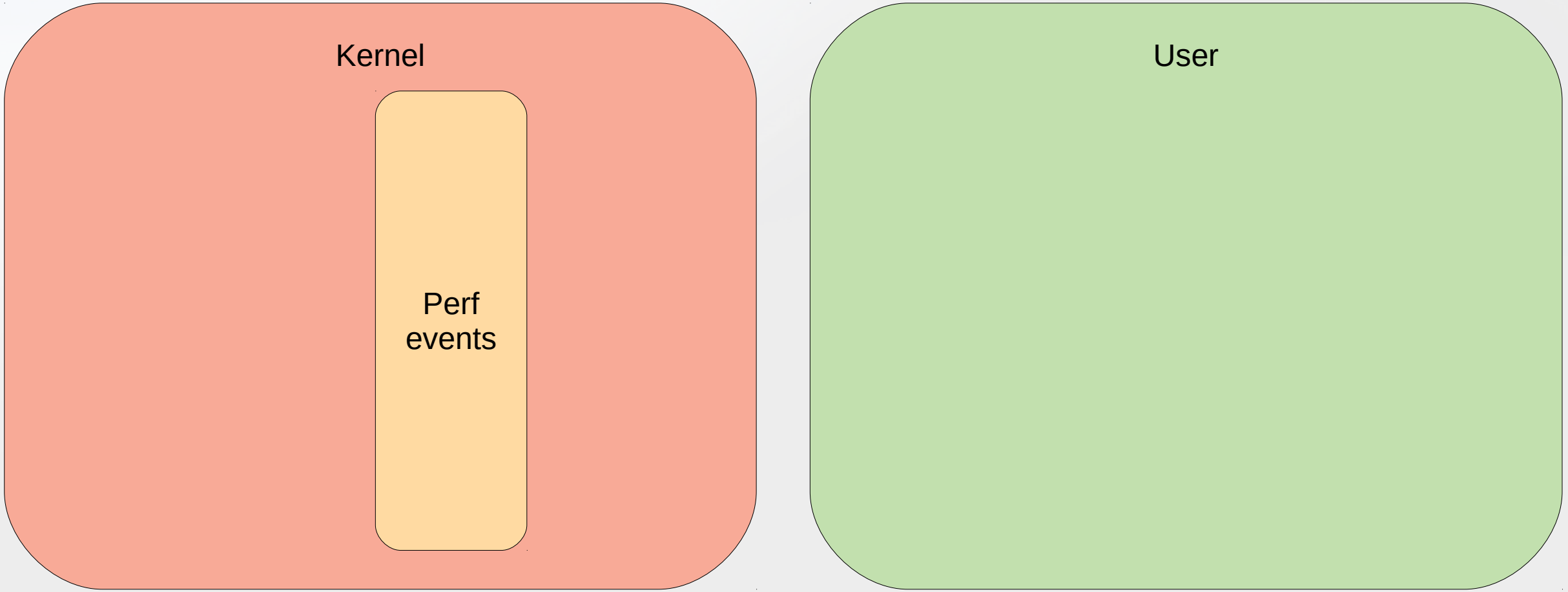
Teoria a życie

VS.

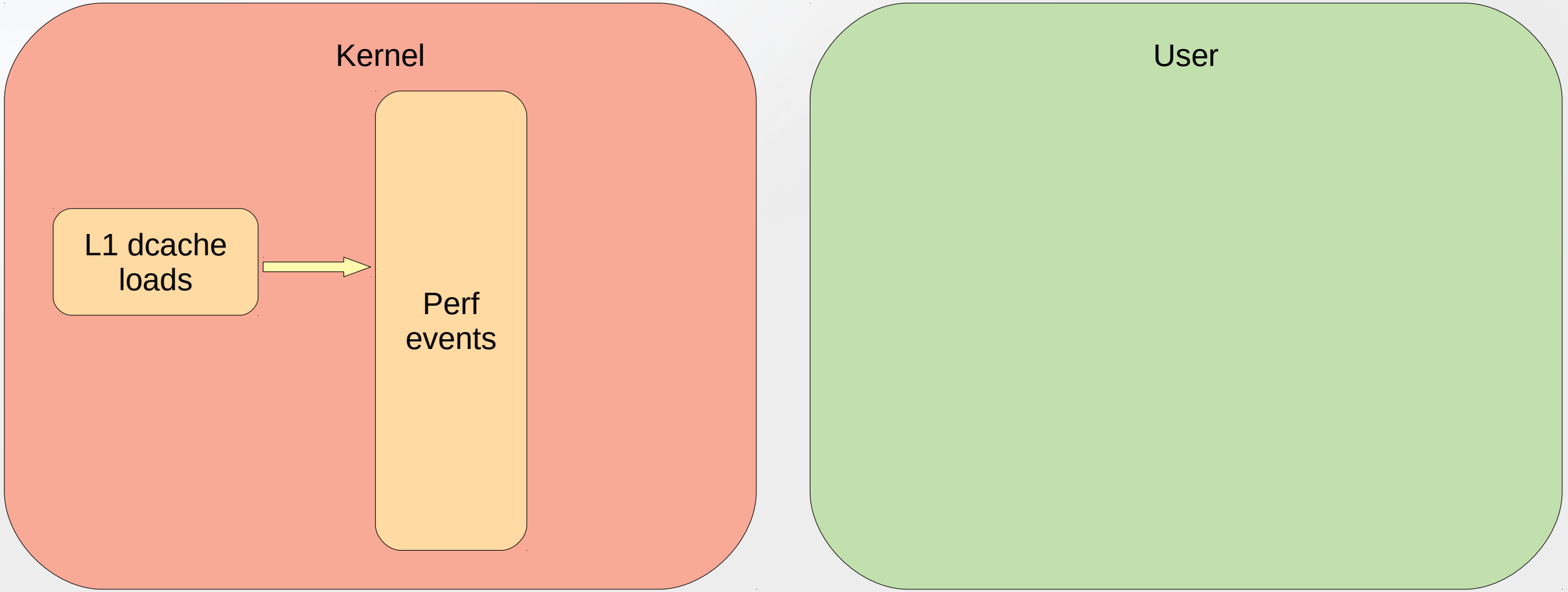
Perf – architektura



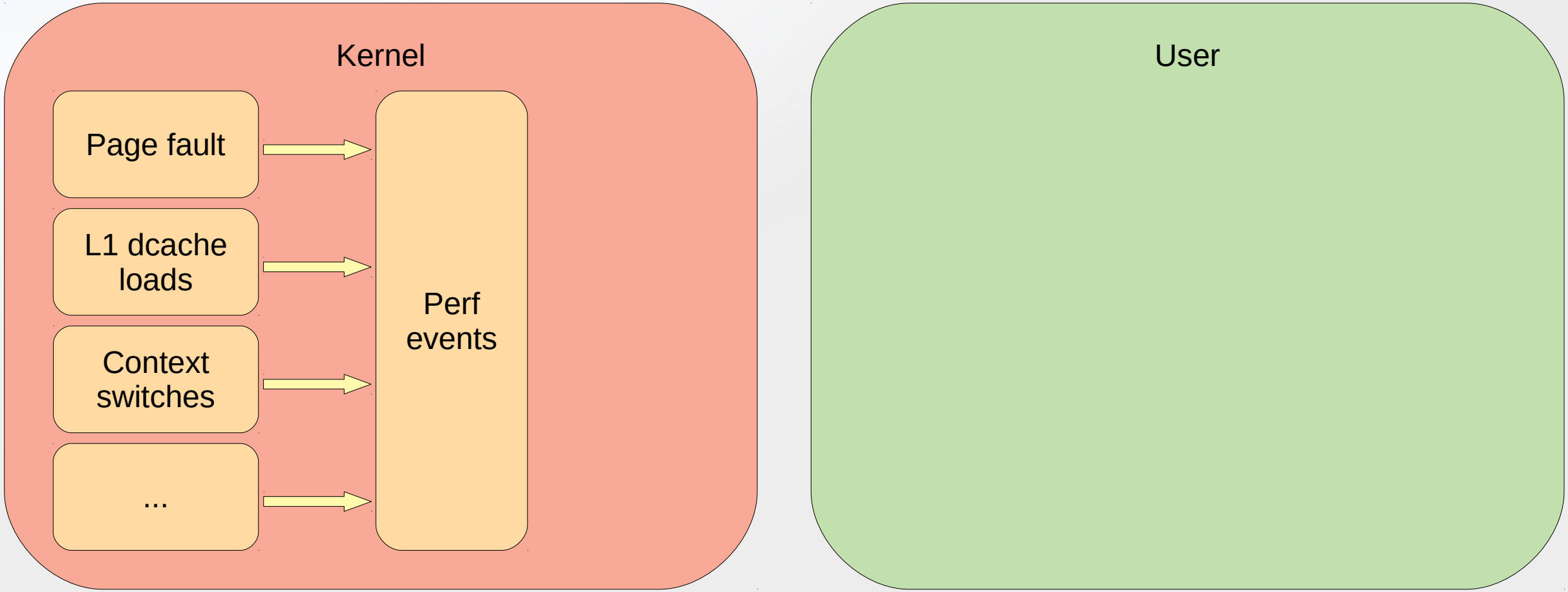
Perf – architektura



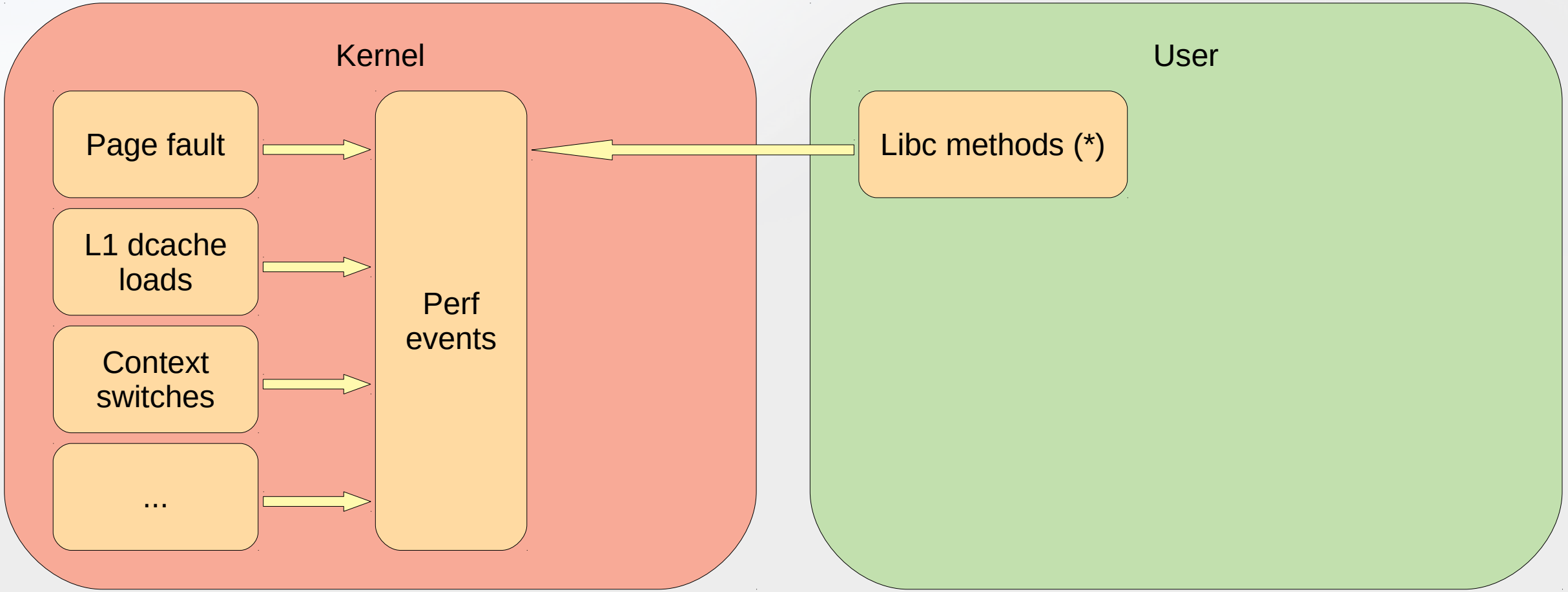
Perf – architektura



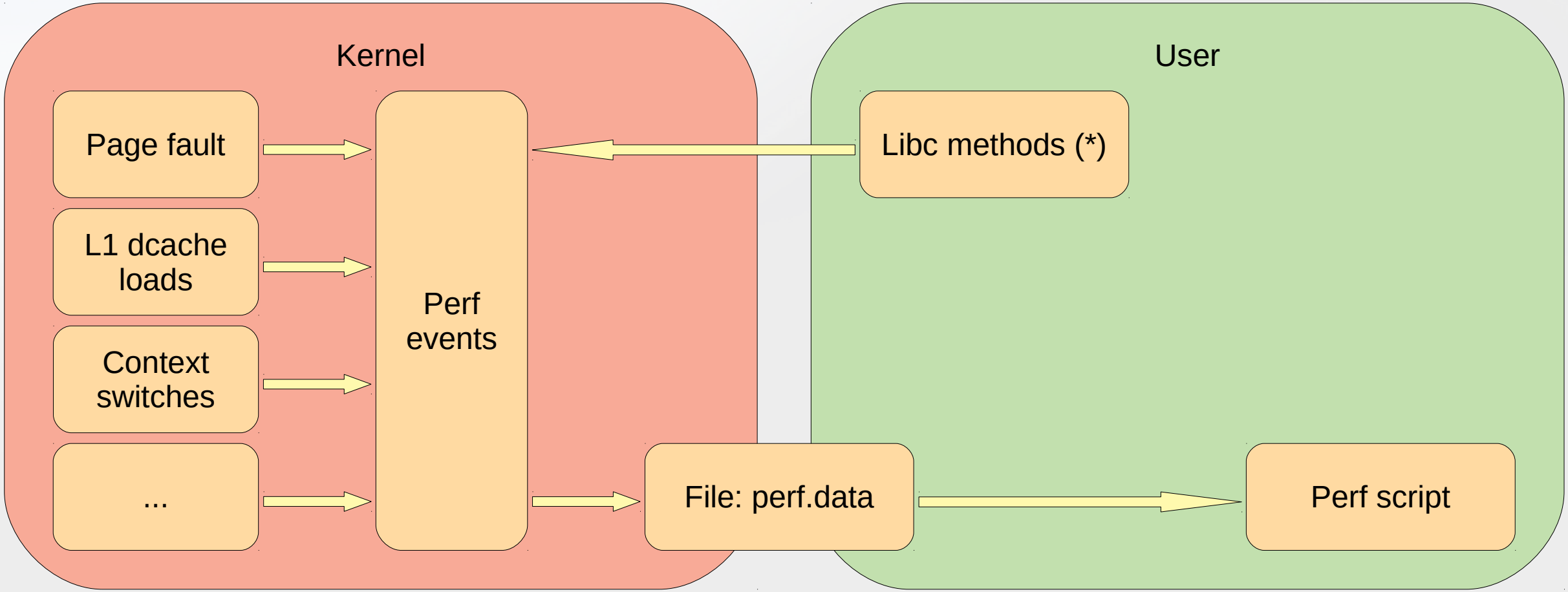
Perf – architektura



Perf – architektura



Perf – architektura



Perf – zalety

- Darmowy

Perf – zalety

- Darmowy
- Wbudowany w Linuksa

Perf – zalety

- Darmowy
- Wbudowany w Linuksa
- Wyjście można przekonwertować do formatu „Collapsed stack”

Perf – zalety

- Darmowy
- Wbudowany w Linuksa
- Wyjście można przekonwertować do formatu „Collapsed stack”
- Bardzo nisko overhead --> śmiało można używać na produkcji

Perf – zalety

- Darmowy
- Wbudowany w Linuksa
- Wyjście można przekonwertować do formatu „Collapsed stack”
- Bardzo nisko overhead --> śmiało można używać na produkcji
- Stabilny, chociaż w przeszłości było różnie

Perf – zalety

- Darmowy
- Wbudowany w Linuksa
- Wyjście można przekonwertować do formatu „Collapsed stack”
- Bardzo nisko overhead --> śmiało można używać na produkcji
- Stabilny, chociaż w przeszłości było różnie
- Umie śledzić cały OS

Perf – zalety

- Darmowy
- Wbudowany w Linuksa
- Wyjście można przekonwertować do formatu „Collapsed stack”
- Bardzo nisko overhead --> śmiało można używać na produkcji
- Stabilny, chociaż w przeszłości było różnie
- Umie śledzić cały OS
- Dużo narzędzi pomocniczych (dziękujemy - Brendan Gregg)

Perf – wady

- Nie wie co to JVM – potrzebuje Perf-map-agenta

Perf – wady

- Nie wie co to JVM – potrzebuje Perf-map-agenta
- Spory próg wejścia

Perf – wady

- Nie wie co to JVM – potrzebuje Perf-map-agenta
- Spory próg wejścia
- Należy włączyć `-XX:+PreserveFramePointer` – dodatkowy overhead ~1-3%, edge cases 0/10%

Perf – wady

- Nie wie co to JVM – potrzebuje Perf-map-agenta
- Spory próg wejścia
- Należy włączyć `-XX:+PreserveFramePointer` – dodatkowy overhead ~1-3%, edge cases 0/10%
- Przesyła bardzo dużo danych z przestrzeni jądra do przestrzeni użytkownika (konkurencja BPF)

Perf – wady

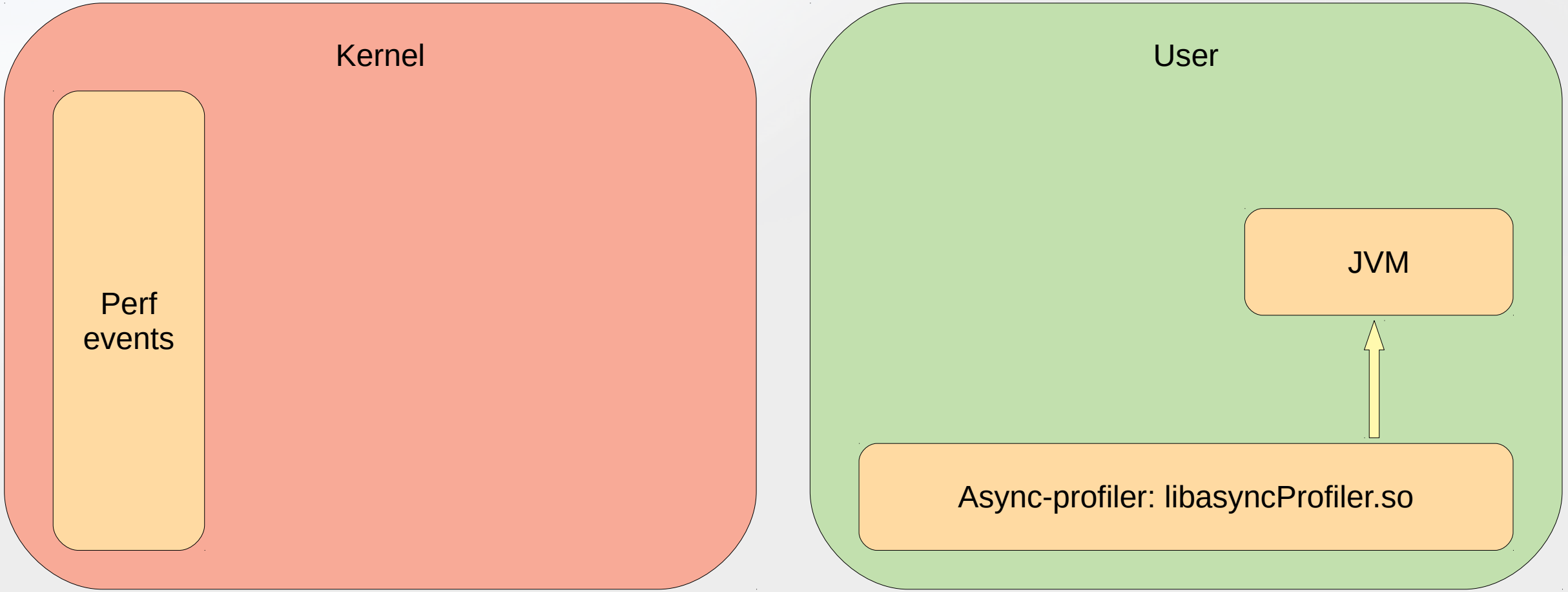
- Nie wie co to JVM – potrzebuje Perf-map-agenta
- Spory próg wejścia
- Należy włączyć `-XX:+PreserveFramePointer` – dodatkowy overhead ~1-3%, edge cases 0/10%
- Przesyła bardzo dużo danych z przestrzeni jądra do przestrzeni użytkownika (konkurencja BPF)
- Brak GUI

Perf – wady

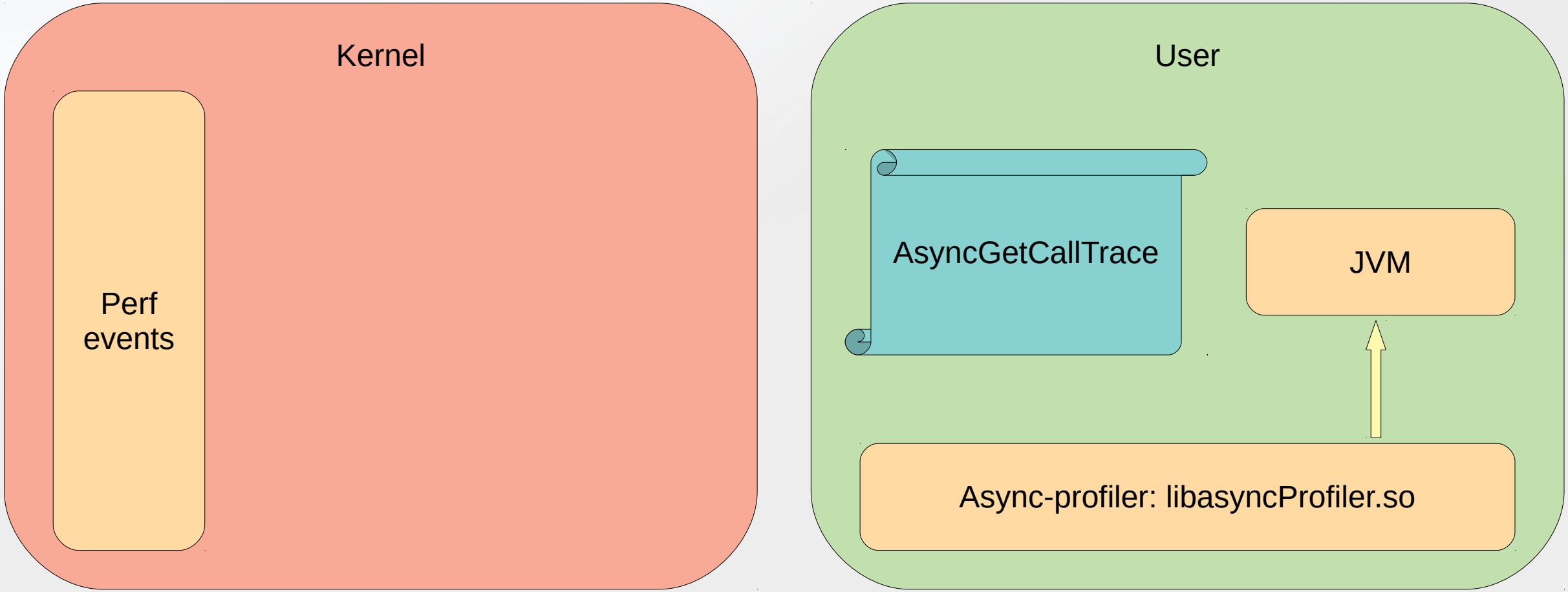
- Nie wie co to JVM – potrzebuje Perf-map-agenta
- Spory próg wejścia
- Należy włączyć `-XX:+PreserveFramePointer` – dodatkowy overhead ~1-3%, edge cases 0/10%
- Przesyła bardzo dużo danych z przestrzeni jądra do przestrzeni użytkownika (konkurencja BPF)
- Brak GUI
- Brak widoku pokazującego coś „na żywo”

Perf – uruchomienie – przykład

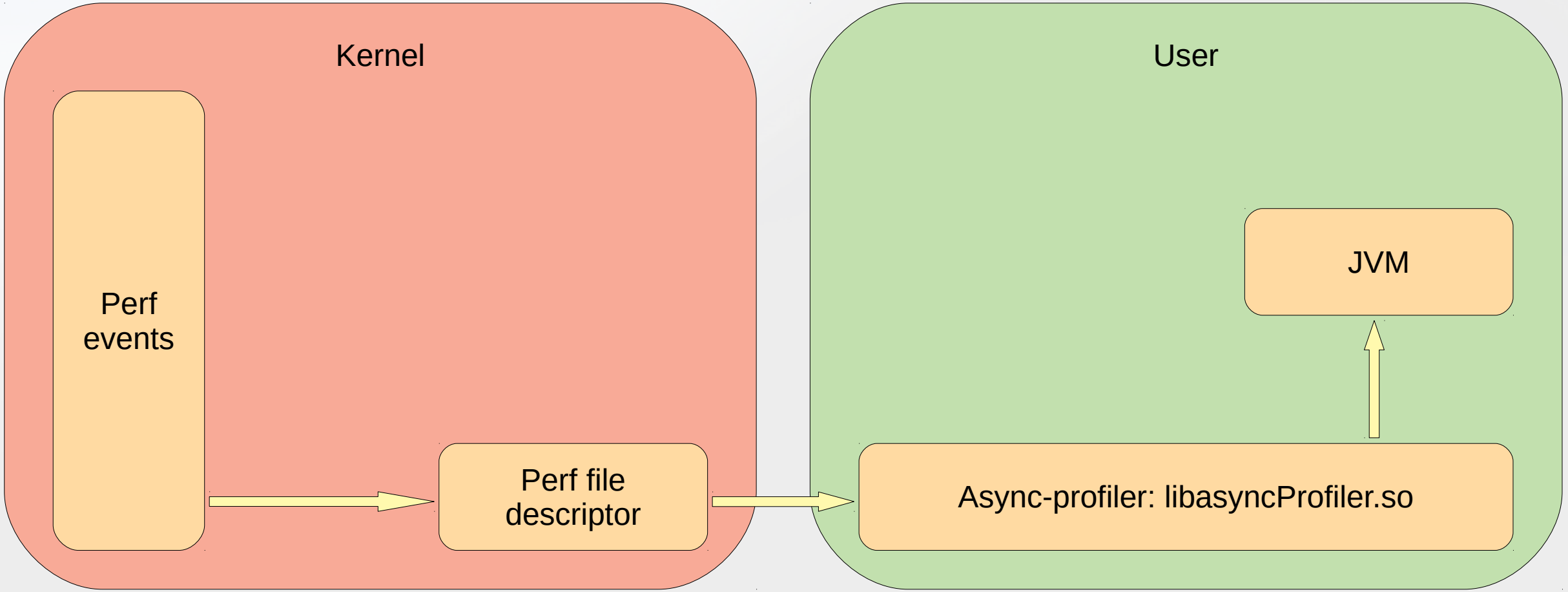
Async-profiler – architektura



Async-profiler – architektura



Async-profiler – architektura



Async-profiler – zalety

- Darmowy (dziękujemy Andrei Pangin)

Async-profiler – zalety

- Darmowy (dziękujemy Andrei Pangin)
- Profiluje kod Javy, JVM i Kernel

Async-profiler – zalety

- Darmowy (dziękujemy Andrei Pangin)
- Profiluje kod Javy, JVM i Kernel
- Nie potrzebuje szerokich uprawnień

Async-profiler – zalety

- Darmowy (dziękujemy Andrei Pangin)
- Profiluje kod Javy, JVM i Kernel
- Nie potrzebuje szerokich uprawnień
- Wbudowany w IntelliJ Idea

Async-profiler – zalety

- Darmowy (dziękujemy Andrei Pangin)
- Profiluje kod Javy, JVM i Kernel
- Nie potrzebuje szerokich uprawnień
- Wbudowany w IntelliJ Idea
- Format „Collapsed stack” i JFR

Async-profiler – zalety

- Darmowy (dziękujemy Andrei Pangin)
- Profiluje kod Javy, JVM i Kernel
- Nie potrzebuje szerokich uprawnień
- Wbudowany w IntelliJ Idea
- Format „Collapsed stack” i JFR
- Bardzo nisko overhead ~1% --> śmiało można używać na produkcji (*)

Async-profiler – zalety

- Darmowy (dziękujemy Andrei Pangin)
- Profiluje kod Javy, JVM i Kernel
- Nie potrzebuje szerokich uprawnień
- Wbudowany w IntelliJ Idea
- Format „Collapsed stack” i JFR
- Bardzo nisko overhead ~1% --> śmiało można używać na produkcji (*)
- Możliwość podłączenia

Async-profiler – zalety

- Darmowy (dziękujemy Andrei Pangin)
- Profiluje kod Javy, JVM i Kernel
- Nie potrzebuje szerokich uprawnień
- Wbudowany w IntelliJ Idea
- Format „Collapsed stack” i JFR
- Bardzo nisko overhead ~1% --> śmiało można używać na produkcji (*)
- Możliwość podłączenia
  - Jako agent

Async-profiler – zalety

- Darmowy (dziękujemy Andrei Pangin)
- Profiluje kod Javy, JVM i Kernel
- Nie potrzebuje szerokich uprawnień
- Wbudowany w IntelliJ Idea
- Format „Collapsed stack” i JFR
- Bardzo nisko overhead ~1% --> śmiało można używać na produkcji (*)
- Możliwość podłączenia
  - Jako agent
  - Do każdego JVMa bez restartu (*)

Async-profiler – wady

- Tylko

Async-profiler – wady

- Tylko
  - Linux (x64 / x86 / ARM / AArch64)
  - MacOS (x64)

Async-profiler – wady

- Tylko
  - Linux (x64 / x86 / ARM / AArch64)
  - MacOS (x64)
- Brak GUI

Async-profiler – wady

- Tylko
  - Linux (x64 / x86 / ARM / AArch64)
  - MacOS (x64)
- Brak GUI
- Brak widoku pokazującego coś „na żywo”

Async-profiler – wady

- Tylko
  - Linux (x64 / x86 / ARM / AArch64)
  - MacOS (x64)
- Brak GUI
- Brak widoku pokazującego coś „na żywo”
- (*) Umie wywalić JVMa – bug w JVM
  Workaround – wystartować Async-profiler jako agent
  `https://github.com/jvm-profiling-tools/async-profiler/issues/154`
  `https://bugs.openjdk.java.net/browse/JDK-8212155`

Async-profiler – uruchomienie – przykład

Perf vs Async-profiler

Perf	Async-profiler

Perf vs Async-profiler

Perf	Async-profiler
Bardziej wygrzany	Relatywnie nowy

Perf vs Async-profiler

Perf	Async-profiler
Bardziej wygrzany	Relatywnie nowy
Umie profilować cały OS	Profiluje w kontekście procesu

Perf vs Async-profiler

Perf	Async-profiler
Bardziej wygrzany	Relatywnie nowy
Umie profilować cały OS	Profiluje w kontekście procesu
Umie profilować kontenery z „hosta”	Profiluje w kontekście procesu

Perf vs Async-profiler

Perf	Async-profiler
Bardziej wygrzany	Relatywnie nowy
Umie profilować cały OS	Profiluje w kontekście procesu
Umie profilować kontenery z „hosta”	Profiluje w kontekście procesu
Nie zna metod interpretowanych	Zna metody interpretowane

Perf vs Async-profiler

Perf	Async-profiler
Bardziej wygrzany	Relatywnie nowy
Umie profilować cały OS	Profiluje w kontekście procesu
Umie profilować kontenery z „hosta”	Profiluje w kontekście procesu
Nie zna metod interpretowanych	Zna metody interpretowane
Wymaga -XX:+PreserveFramePointer	Nie wymaga dodatkowych flag JVM

Perf vs Async-profiler

Perf	Async-profiler
Bardziej wygrzany	Relatywnie nowy
Umie profilować cały OS	Profiluje w kontekście procesu
Umie profilować kontenery z „hosta”	Profiluje w kontekście procesu
Nie zna metod interpretowanych	Zna metody interpretowane
Wymaga -XX:+PreserveFramePointer	Nie wymaga dodatkowych flag JVM
JDK >= 8u60	JDK >= 7u40 dla niektórych funkcjonalności

Perf vs Async-profiler

Perf	Async-profiler
Bardziej wygrzany	Relatywnie nowy
Umie profilować cały OS	Profiluje w kontekście procesu
Umie profilować kontenery z „hosta”	Profiluje w kontekście procesu
Nie zna metod interpretowanych	Zna metody interpretowane
Wymaga -XX:+PreserveFramePointer	Nie wymaga dodatkowych flag JVM
JDK >= 8u60	JDK >= 7u40 dla niektórych funkcjonalności
Potrzebuje Perf-map-agenta	Nie potrzebuje zewnętrznych bibliotek

Perf vs Async-profiler

Perf	Async-profiler
Bardziej wygrzany	Relatywnie nowy
Umie profilować cały OS	Profiluje w kontekście procesu
Umie profilować kontenery z „hosta”	Profiluje w kontekście procesu
Nie zna metod interpretowanych	Zna metody interpretowane
Wymaga -XX:+PreserveFramePointer	Nie wymaga dodatkowych flag JVM
JDK >= 8u60	JDK >= 7u40 dla niektórych funkcjonalności
Potrzebuje Perf-map-agenta	Nie potrzebuje zewnętrznych bibliotek
Nie wie co to JVM	Dedykowane funkcjonalności dla JVM

Perf vs Async-profiler

Perf	Async-profiler
Bardziej wygrzany	Relatywnie nowy
Umie profilować cały OS	Profiluje w kontekście procesu
Umie profilować kontenery z „hosta”	Profiluje w kontekście procesu
Nie zna metod interpretowanych	Zna metody interpretowane
Wymaga -XX:+PreserveFramePointer	Nie wymaga dodatkowych flag JVM
JDK >= 8u60	JDK >= 7u40 dla niektórych funkcjonalności
Potrzebuje Perf-map-agenta	Nie potrzebuje zewnętrznych bibliotek
Nie wie co to JVM	Dedykowane funkcjonalności dla JVM
Format można skonwertować do „Collapsed stack”	Format „Collapsed stack”

Co umie?

Profilowana aplikacja

Używane biblioteki, frameworki i serwery

JVM

OS
HW

Co umie?

Profilowana aplikacja

JVM

Używane biblioteki, frameworki i serwery

OS
HW

Co umie?

Profilowana aplikacja

Używane biblioteki, frameworki i serwery

JVM

OS
HW

Co umie?

Profilowana aplikacja

Używane biblioteki, frameworki i serwery

JVM

OS
HW

Co umie Async-profiler?

- CPU

Co umie Async-profiler?

- CPU
- Wall

Wall vs CPU

Wall vs CPU

- Zatankować

Wall vs CPU

- Zatankować
- Stanąć do kasy

Wall vs CPU

- Zatankować
- Stanąć do kasy
- Zapłacić

Wall vs CPU

- Zatankować
- Stanąć do kasy
- Zapłacić
- Inne

Wall vs CPU

- Zatankować
- Stanąć do kasy
- Zapłacić
- Inne

CPU

Wall vs CPU

- Zatankować
- Stanąć do kasy
- Zapłacić
- Inne

CPU

Lock

Wall vs CPU

- Zatankować
- Stanąć do kasy
- Zapłacić
- Inne

CPU

Lock

IO

Wall vs CPU

- Zatankować
- Stanąć do kasy
- Zapłacić
- Inne

CPU

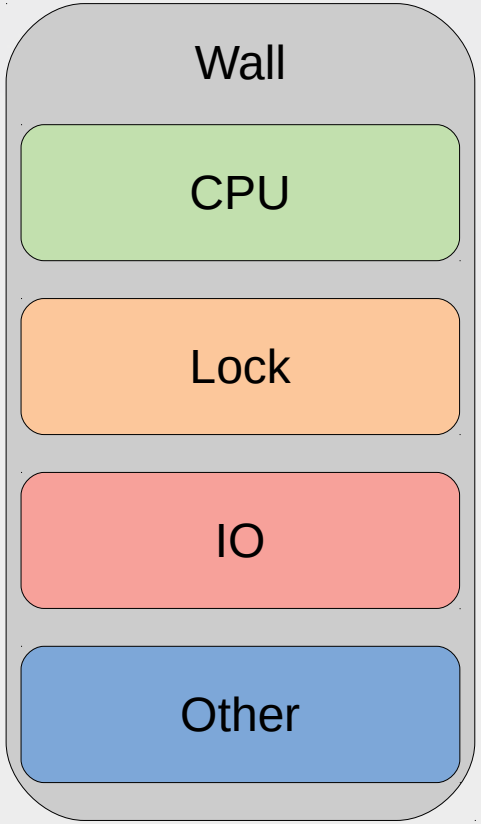
Lock

IO

Other

Wall vs CPU

- Zatankować
- Stanąć do kasy
- Zapłacić
- Inne



Co umie Async-profiler?

- CPU
- Wall
- Method profiling

Co umie Async-profiler?

- CPU
- Wall
- Method profiling
- Memory

Co umie Async-profiler?

- CPU
- Wall
- Method profiling
- Memory
- Lock

Co umie Async-profiler?

- CPU
- Wall
- Method profiling
- Memory
- Lock
- Perf events

Po co „wygrzewam apliakcję”?

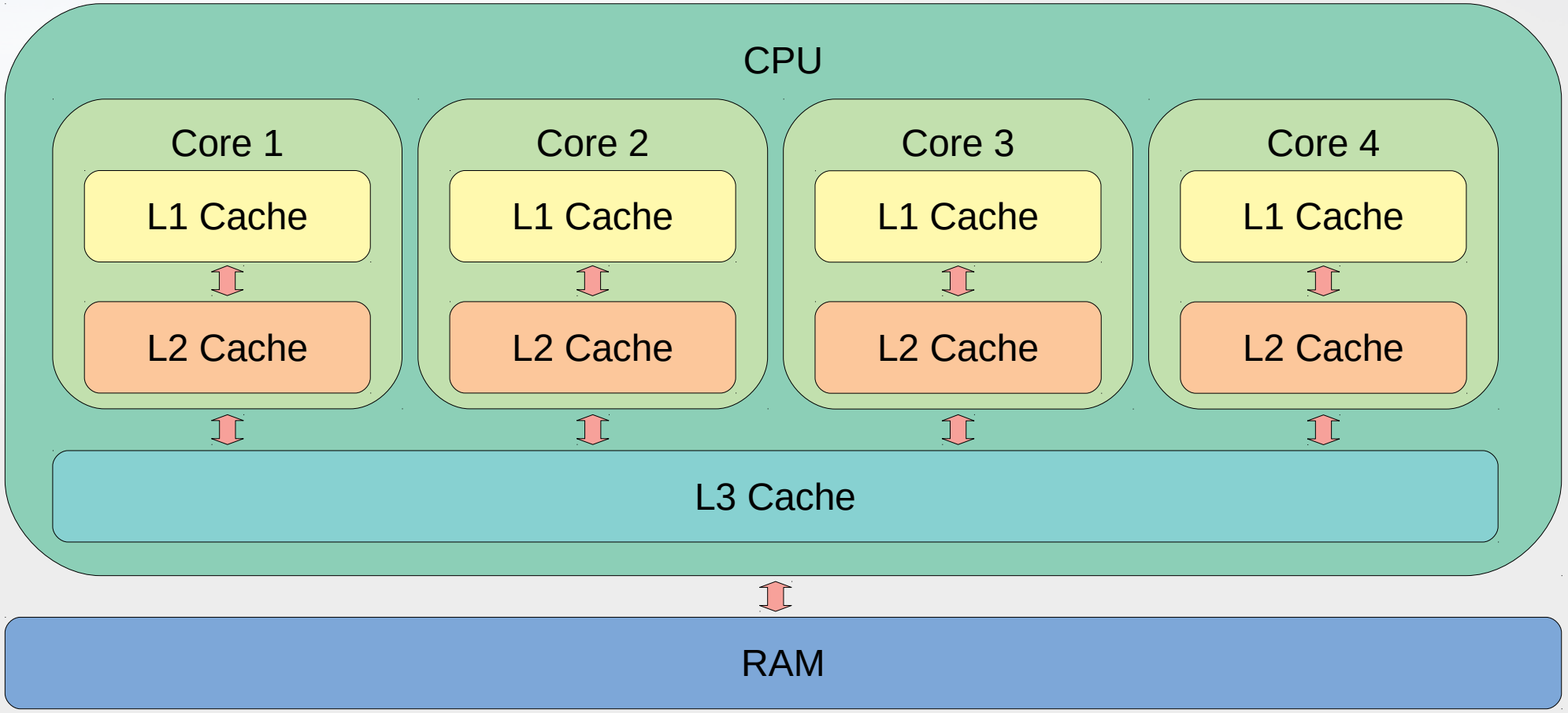
Cała aplikacja wolno działa – przykład

Konkretne coś działa wolno – przykład

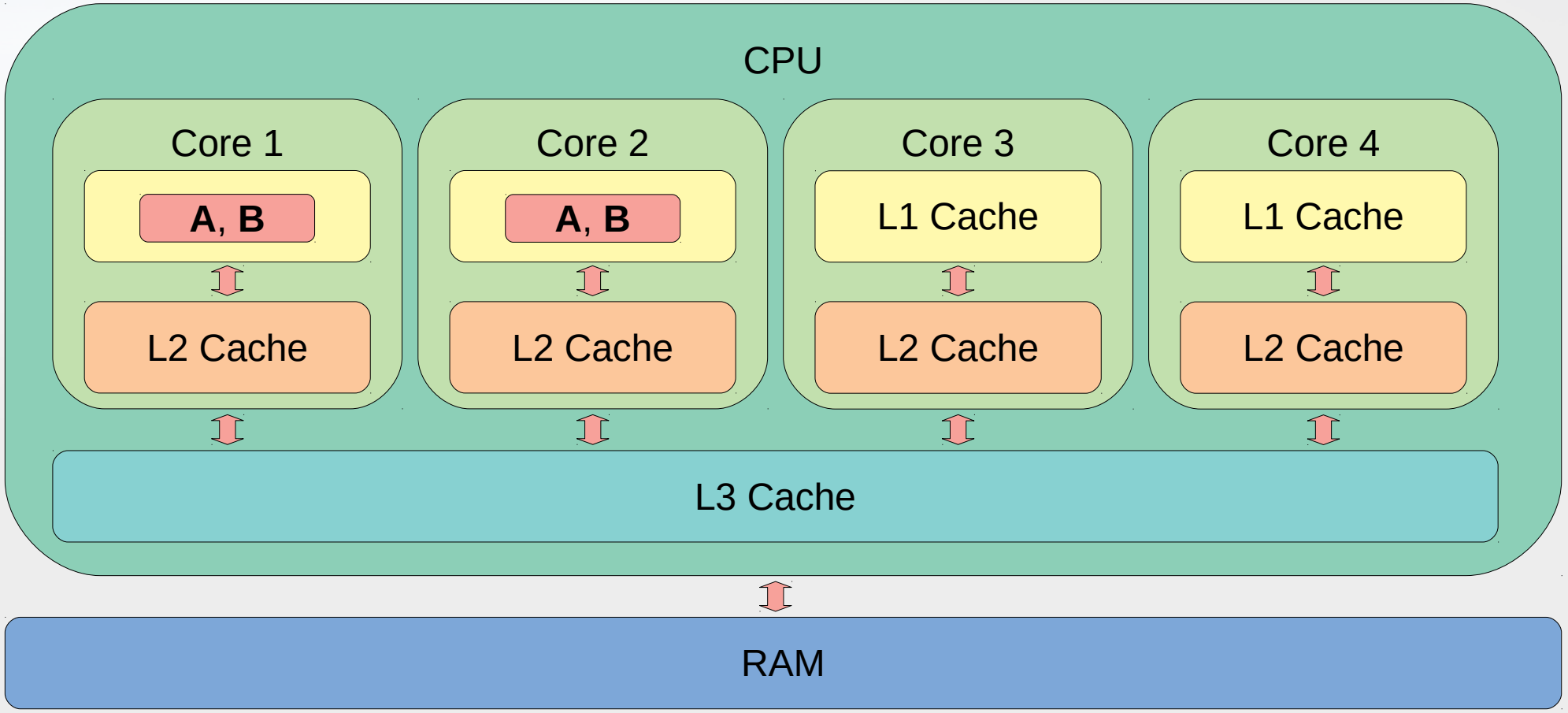
A przy okazji JVM?

https://github.com/jvm-profiling-tools/async-profiler/issues/74

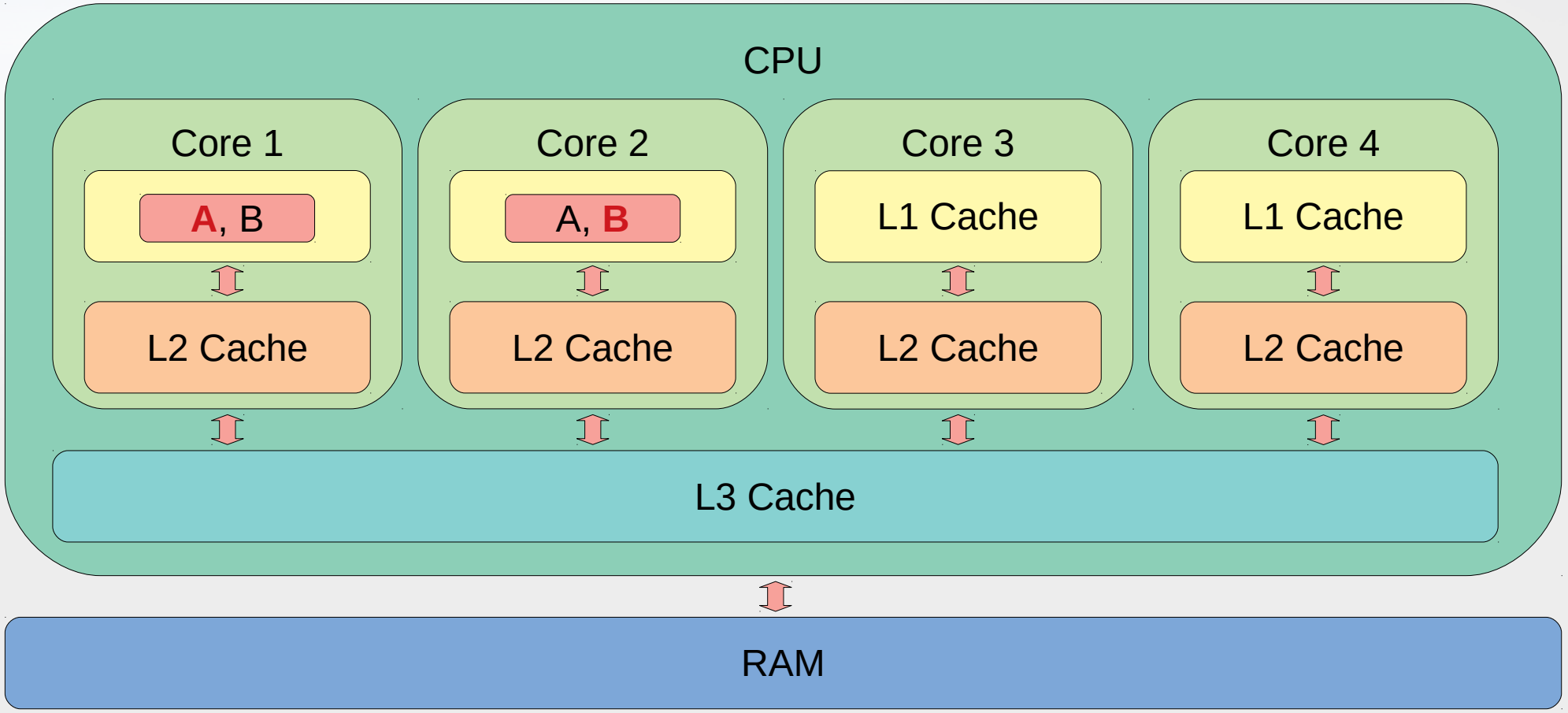
Hardware – false sharing



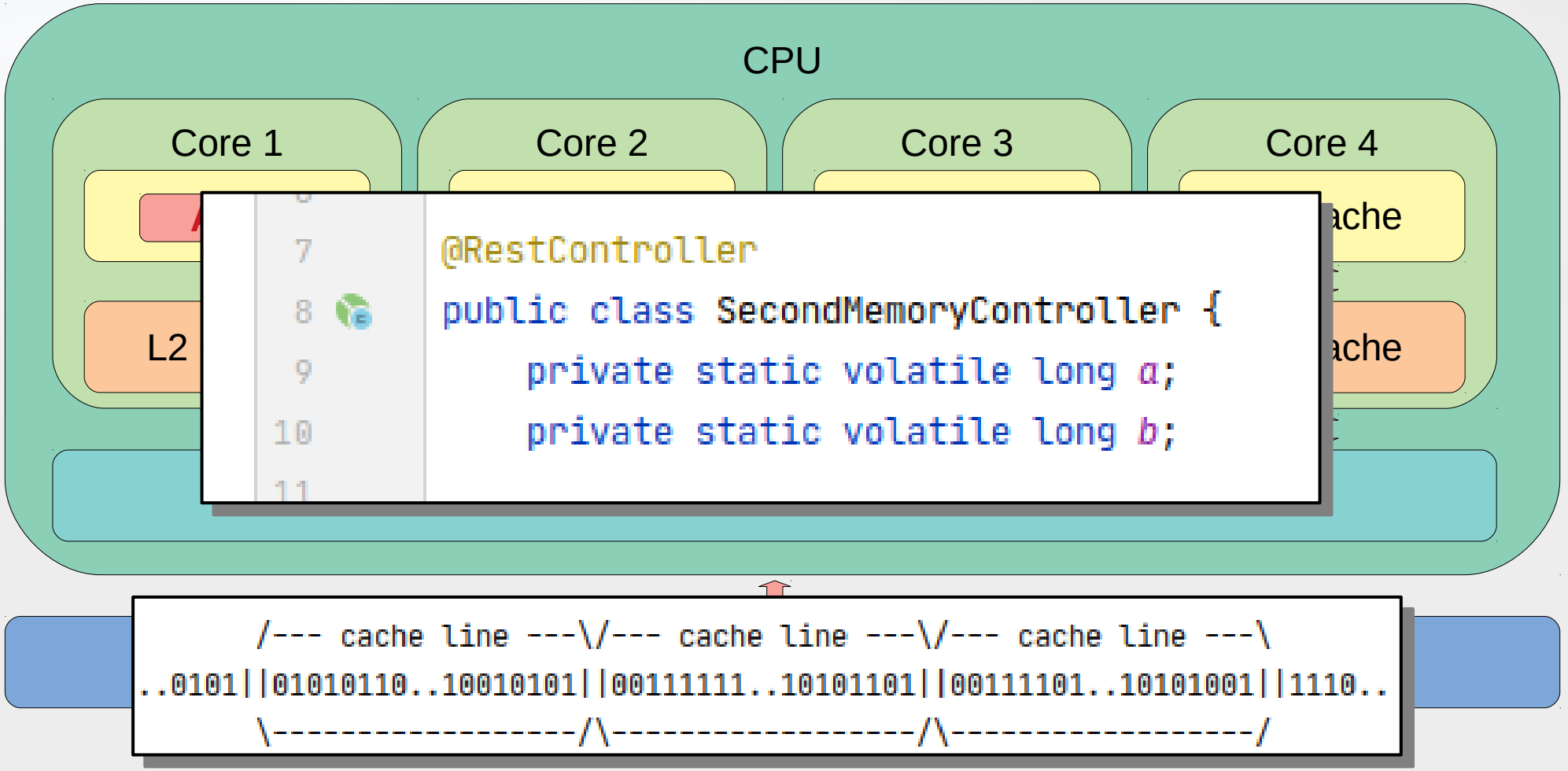
Hardware – false sharing



Hardware – false sharing

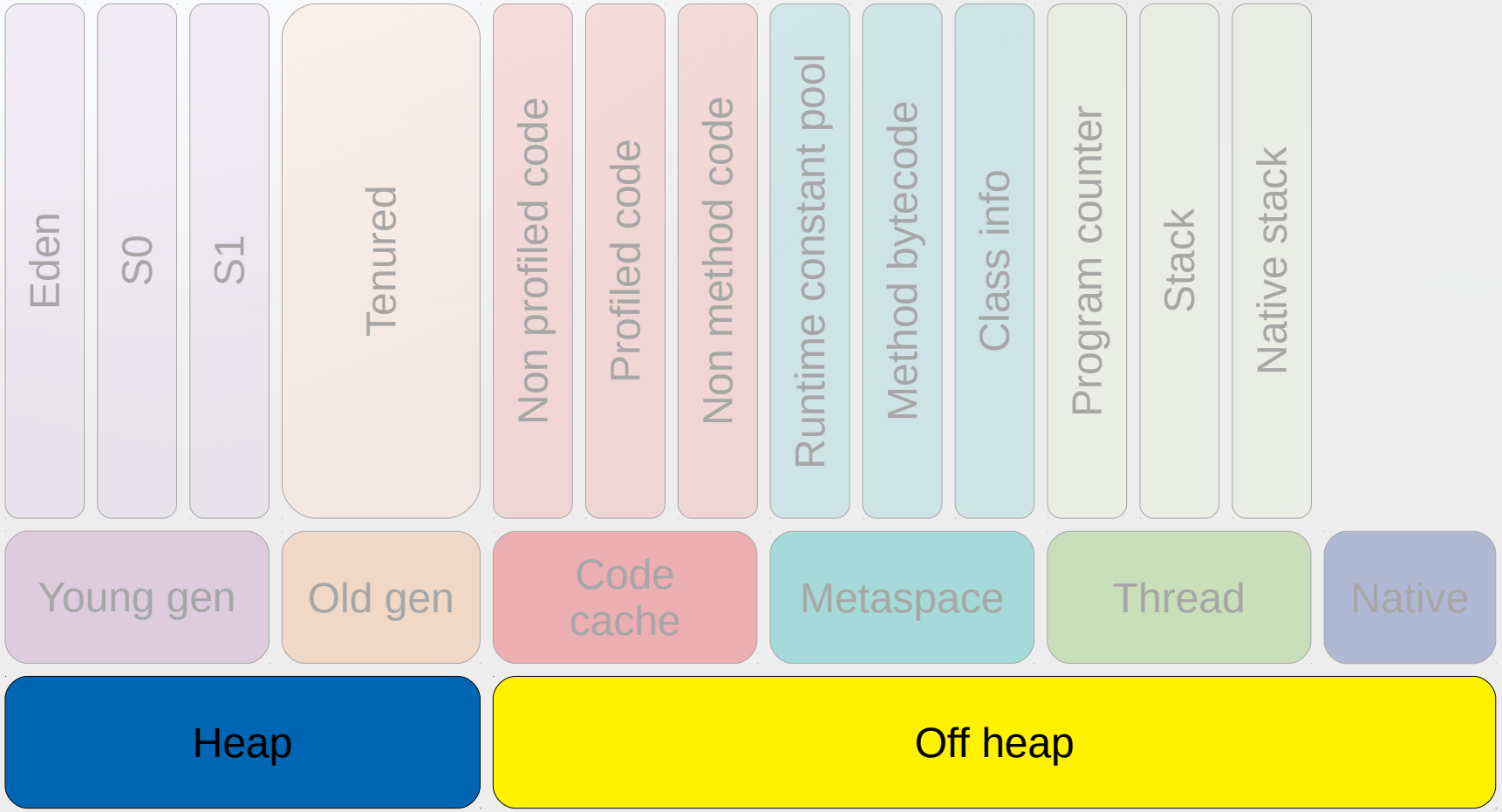


Hardware – false sharing

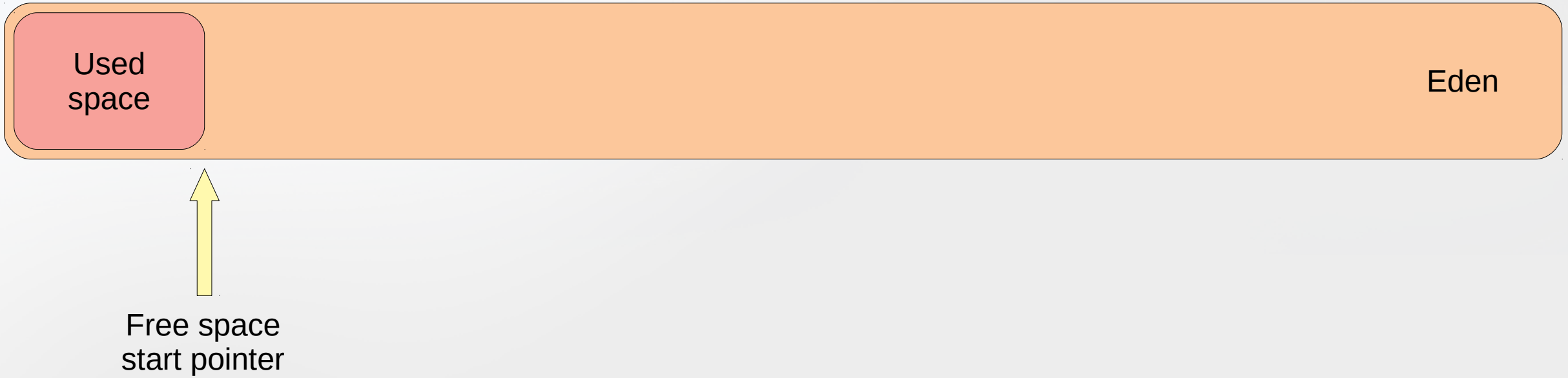


False sharing - przykład

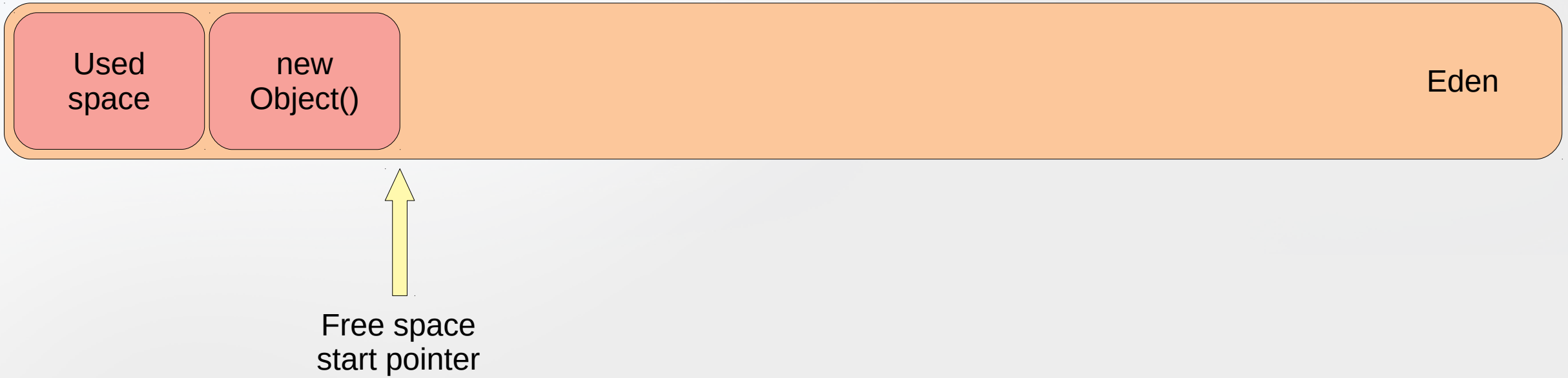
Podział pamięci w JVM – JDK9 - prallel



new Object()



new Object()

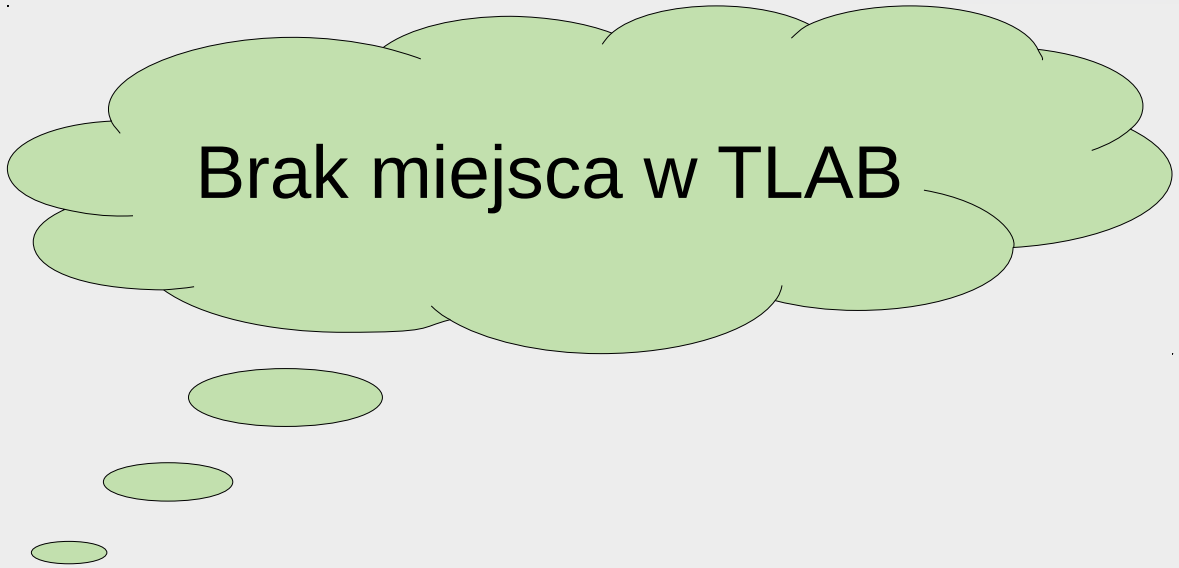


new Object()


Used space
new Object()
Eden
Free space start pointer
>1 Thread ???

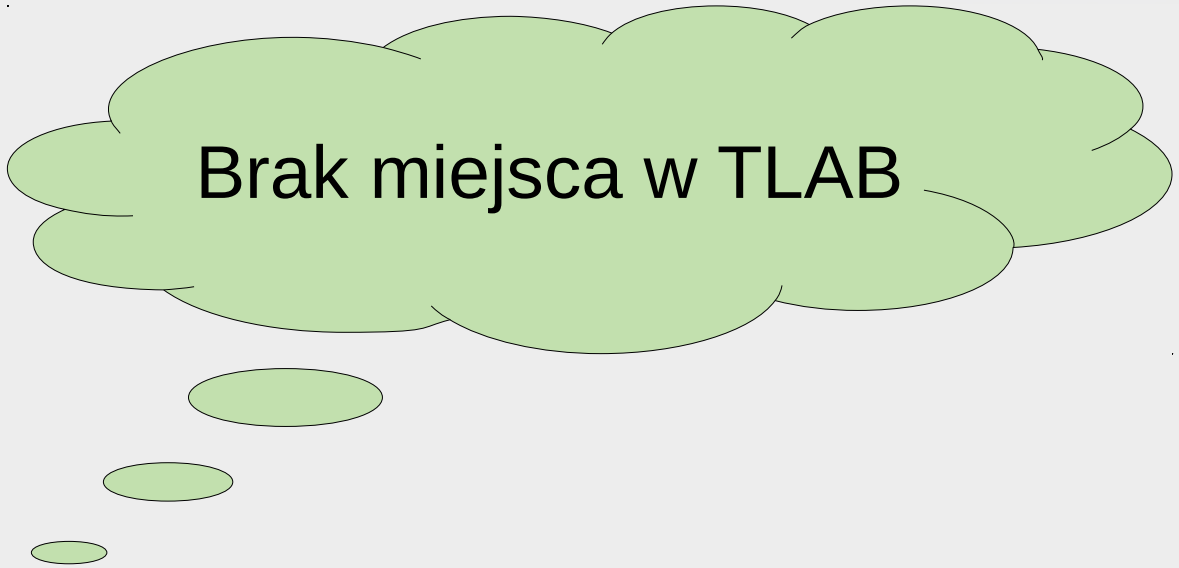
TLAB

TLAB



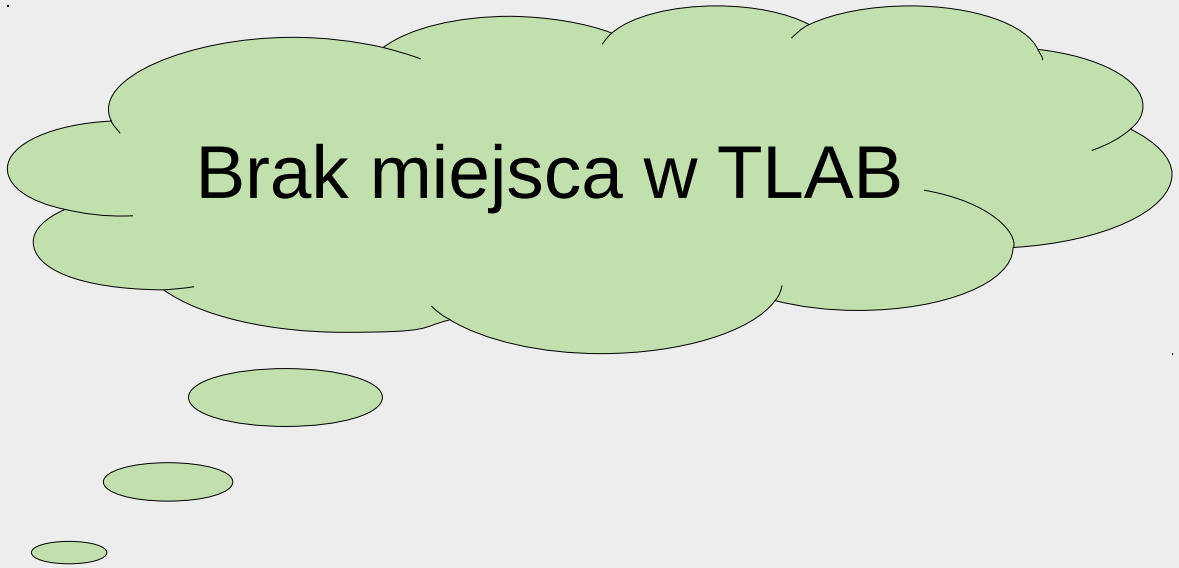
TLAB

- Tworzymy nowy TLAB dedykowany dla wątku



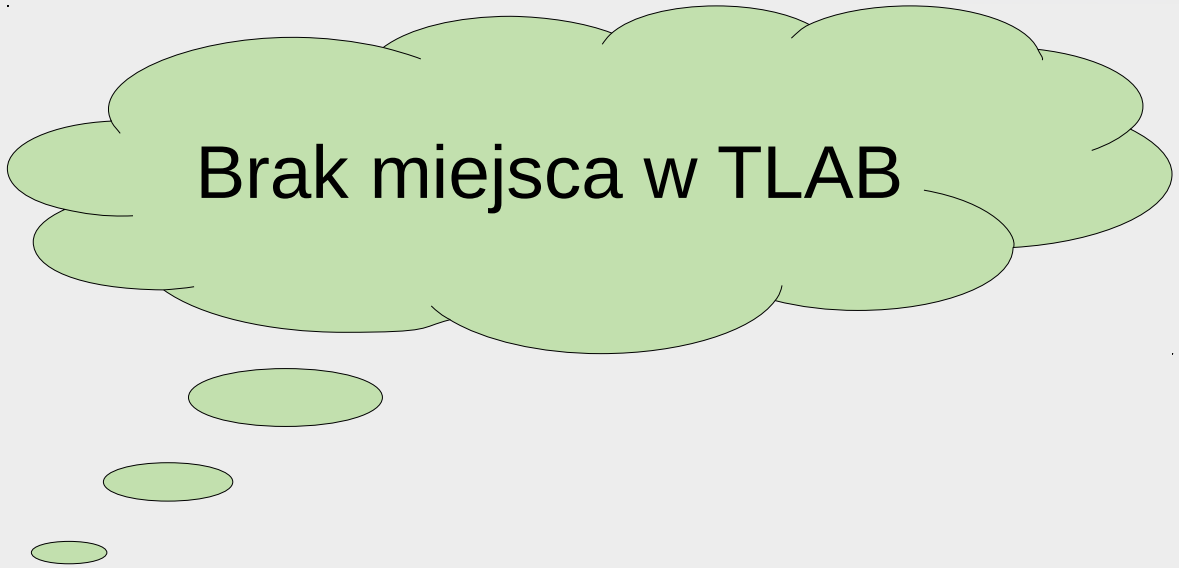
TLAB

- Tworzymy nowy TLAB dedykowany dla wątku
- Stary TLAB jest „retired”



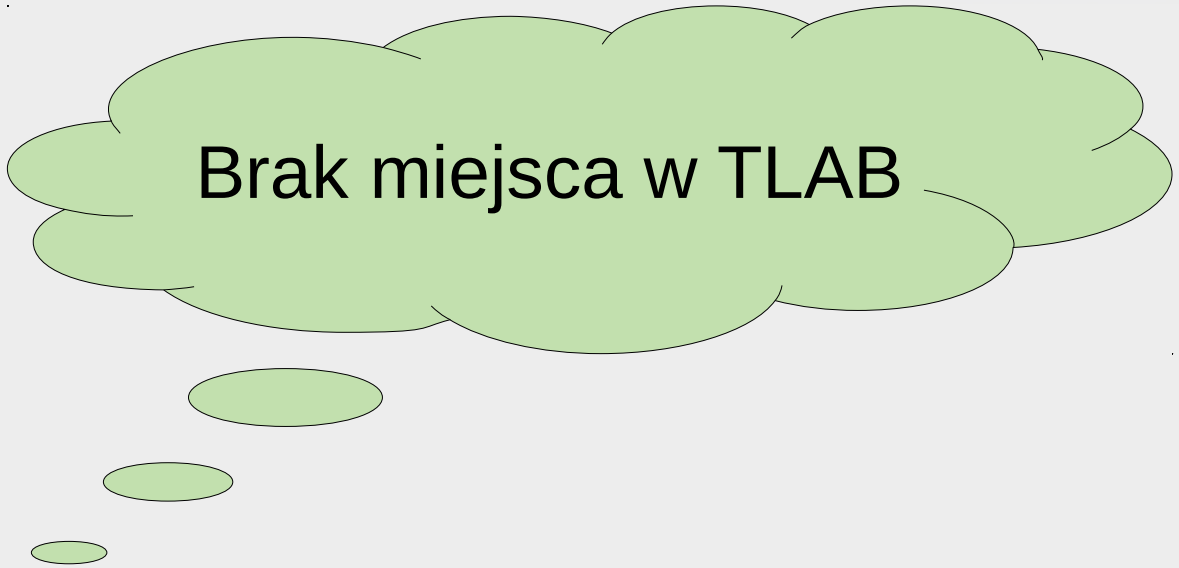
TLAB

- Tworzymy nowy TLAB dedykowany dla wątku
- Stary TLAB jest „retired”
- Jeżeli była jakaś wolna przestrzeń --> jest zmarnowana



TLAB

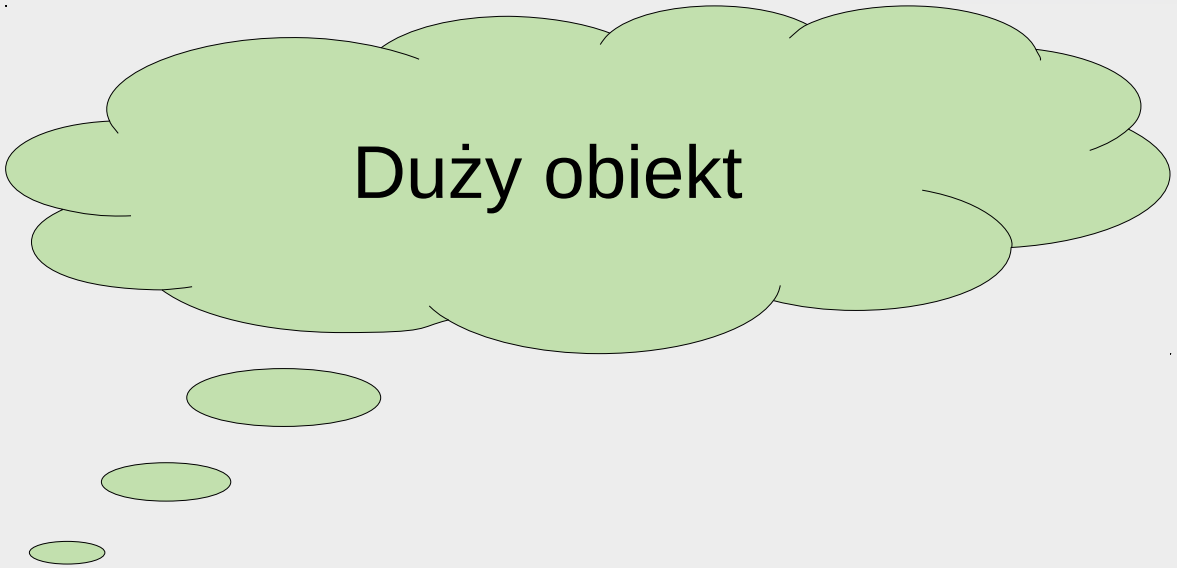
- Tworzymy nowy TLAB dedykowany dla wątku
- Stary TLAB jest „retired”
- Jeżeli była jakaś wolna przestrzeń --> jest zmarnowana
- `-Xlog:gc+tlab=trace`



TLAB

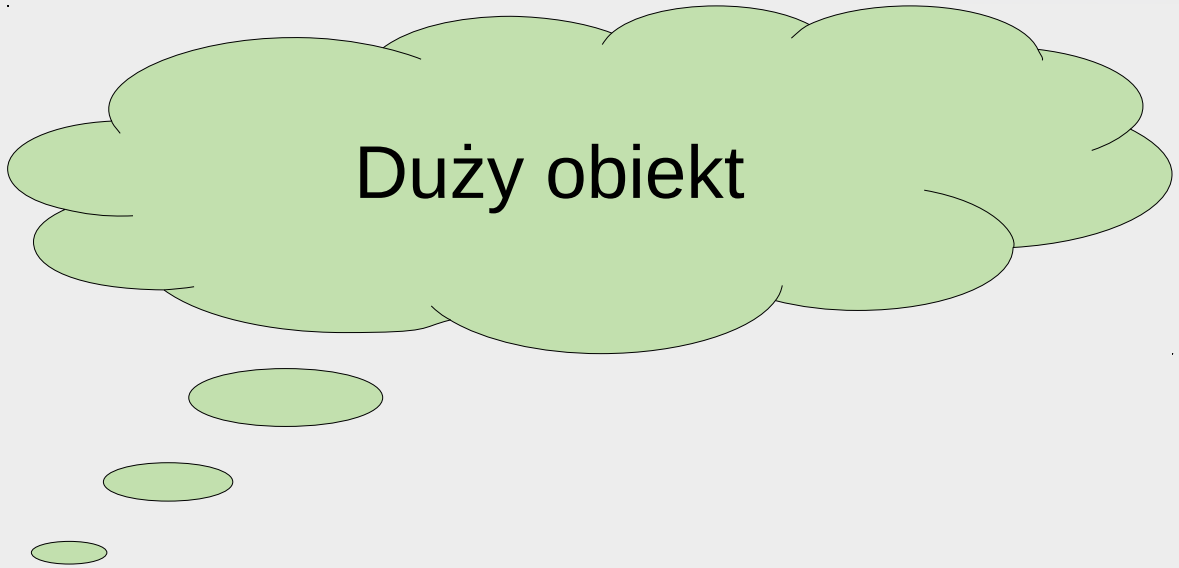
TLAB


Thread 1
TLAB 1
Thread 2
TLAB 1
Thread 1
TLAB 2
Eden


Duży obiekt

TLAB

- „Slow path” - alokacja bezpośrednio w Eden



Async-profiler – memory

- Wie o obiektach alokowanych poza TLAB

Async-profiler – memory

- Wie o obiektach alokowanych poza TLAB
- Wie o obiektach, które wymagały nowego TLABa

Async-profiler – memory

- Wie o obiektach alokowanych poza TLAB
- Wie o obiektach, które wymagały nowego TLABa
- Zalety:

Async-profiler – memory

- Wie o obiektach alokowanych poza TLAB
- Wie o obiektach, które wymagały nowego TLABa
- Zalety:
  - Nie potrzebna instrumentacja kodu

Async-profiler – memory

- Wie o obiektach alokowanych poza TLAB
- Wie o obiektach, które wymagały nowego TLABa
- Zalety:
  - Nie potrzebna instrumentacja kodu
  - Niski overhead

Async-profiler – memory

- Wie o obiektach alokowanych poza TLAB
- Wie o obiektach, które wymagały nowego TLABa
- Zalety:
  - Nie potrzebna instrumentacja kodu
  - Niski overhead
  - To samo umie JFR+JMC --> w starszych JDK płatne

Async-profiler – memory

- Wie o obiektach alokowanych poza TLAB
- Wie o obiektach, które wymagały nowego TLABa
- Zalety:
  - Nie potrzebna instrumentacja kodu
  - Niski overhead
  - To samo umie JFR+JMC --> w starszych JDK płatne
- Wady:

Async-profiler – memory

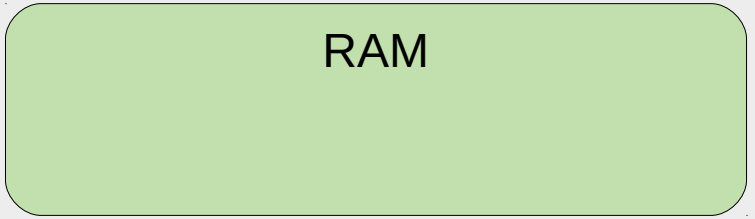
- Wie o obiektach alokowanych poza TLAB
- Wie o obiektach, które wymagały nowego TLABa
- Zalety:
  - Nie potrzebna instrumentacja kodu
  - Niski overhead
  - To samo umie JFR+JMC --> w starszych JDK płatne
- Wady:
  - Widzimy tylko część alokacji (co N kB)

Async-profiler – memory

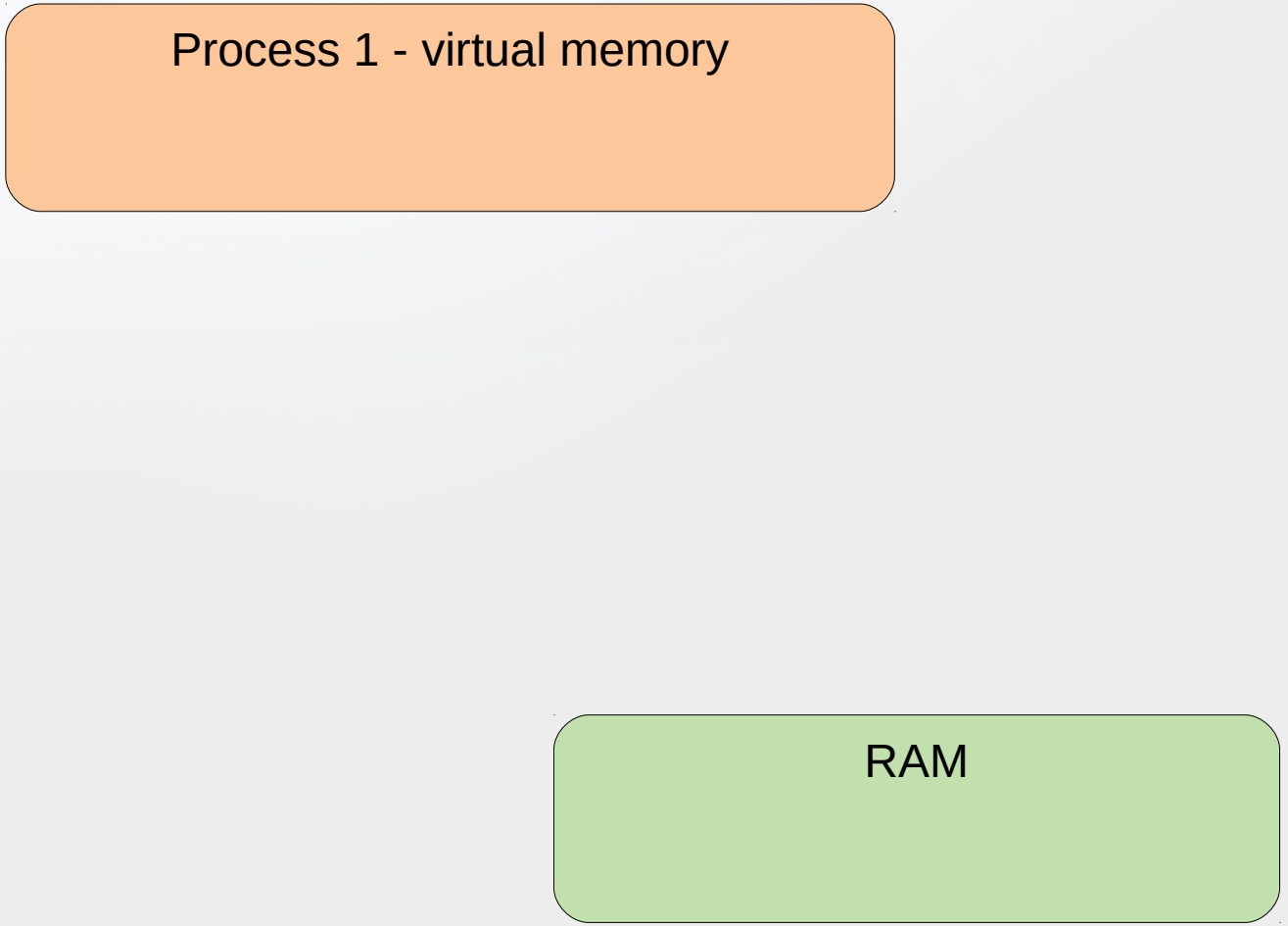
- Wie o obiektach alokowanych poza TLAB
- Wie o obiektach, które wymagały nowego TLABa
- Zalety:
  - Nie potrzebna instrumentacja kodu
  - Niski overhead
  - To samo umie JFR+JMC --> w starszych JDK płatne
- Wady:
  - Widzimy tylko część alokacji (co N kB)
  - Nie śledzi czy obiekt jest żywy czy martwy

Heap allocation – przykład

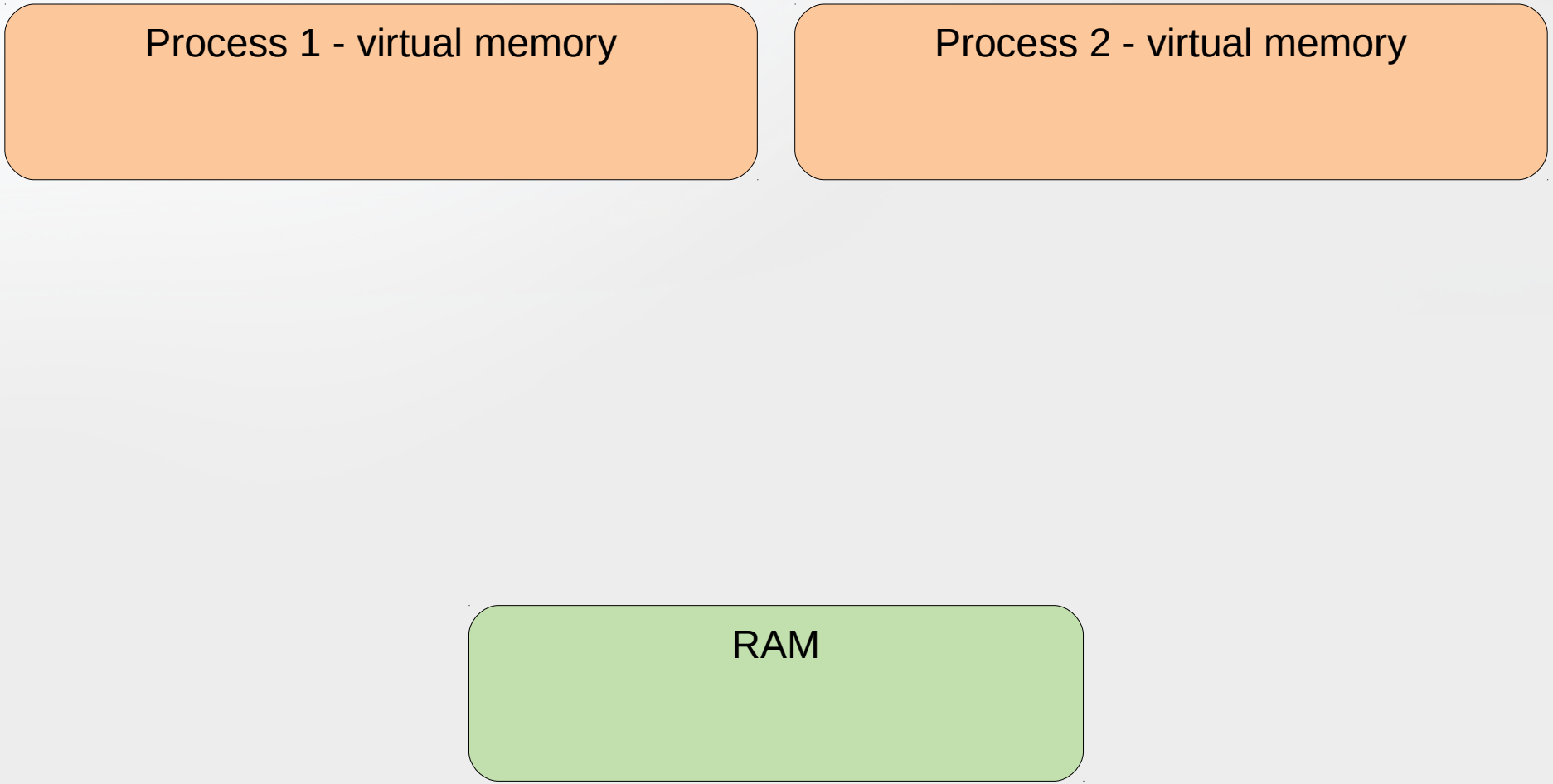
Jak się zjada RAM?



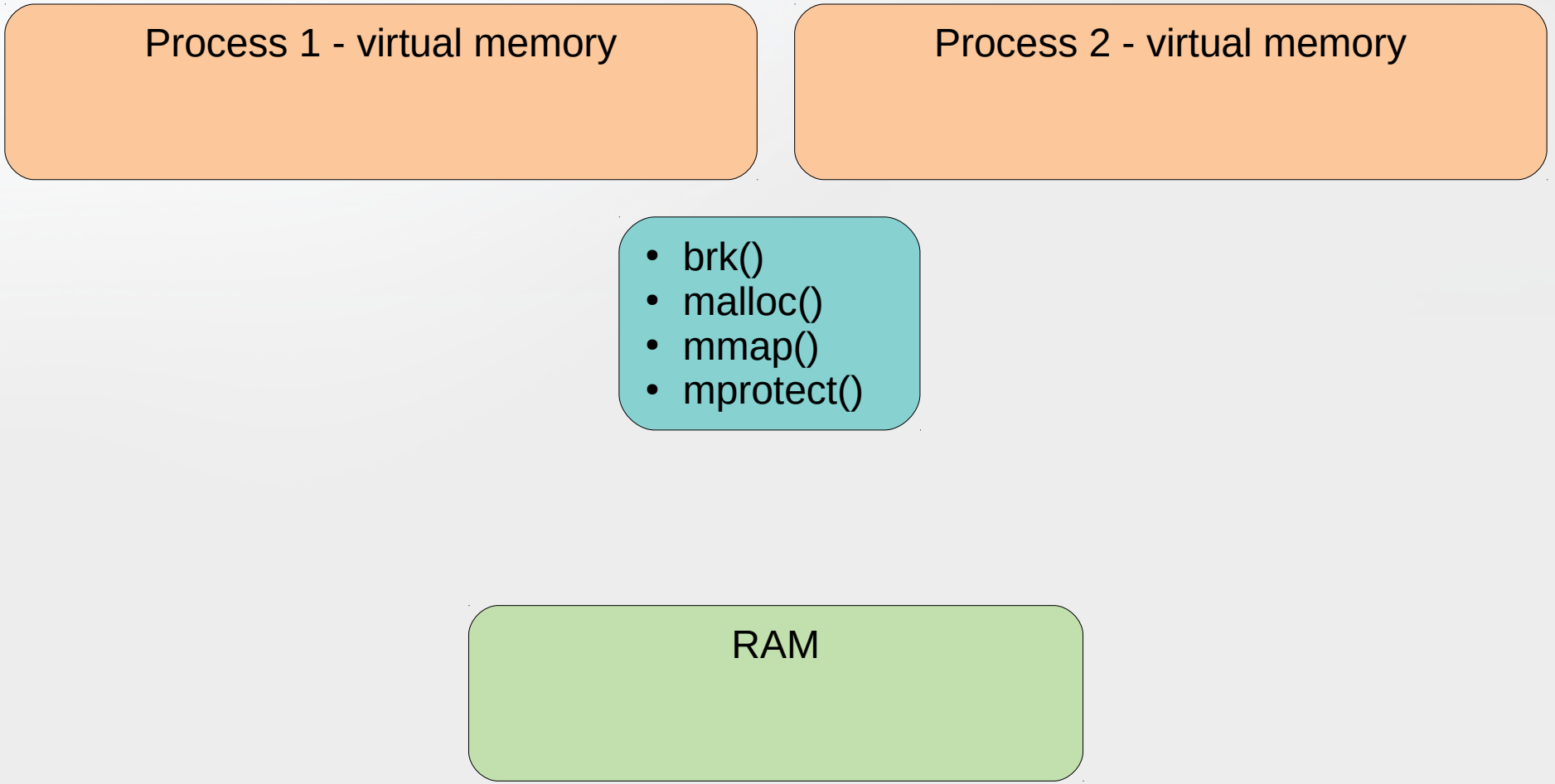
Jak się zjada RAM?



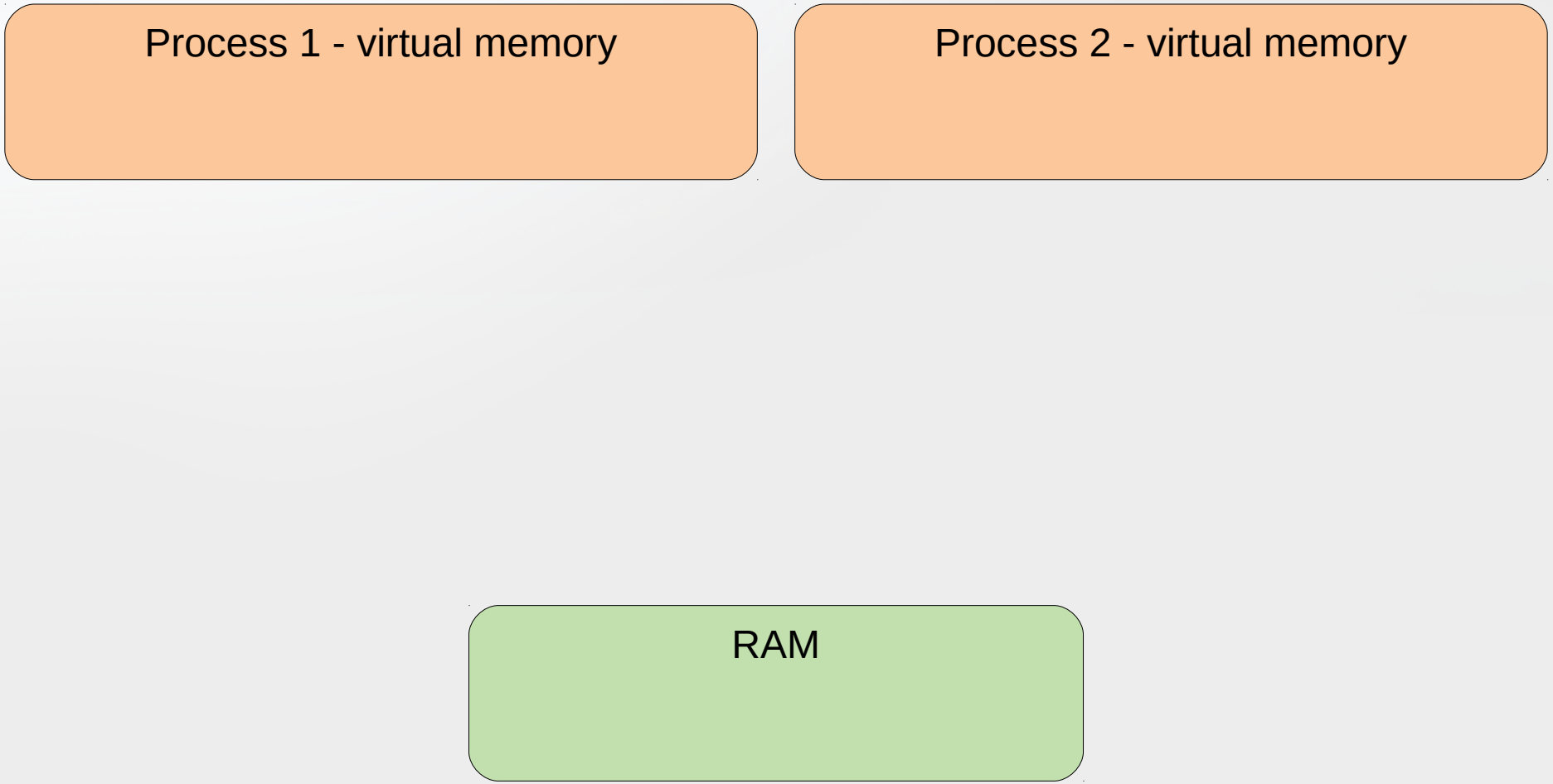
Jak się zjada RAM?



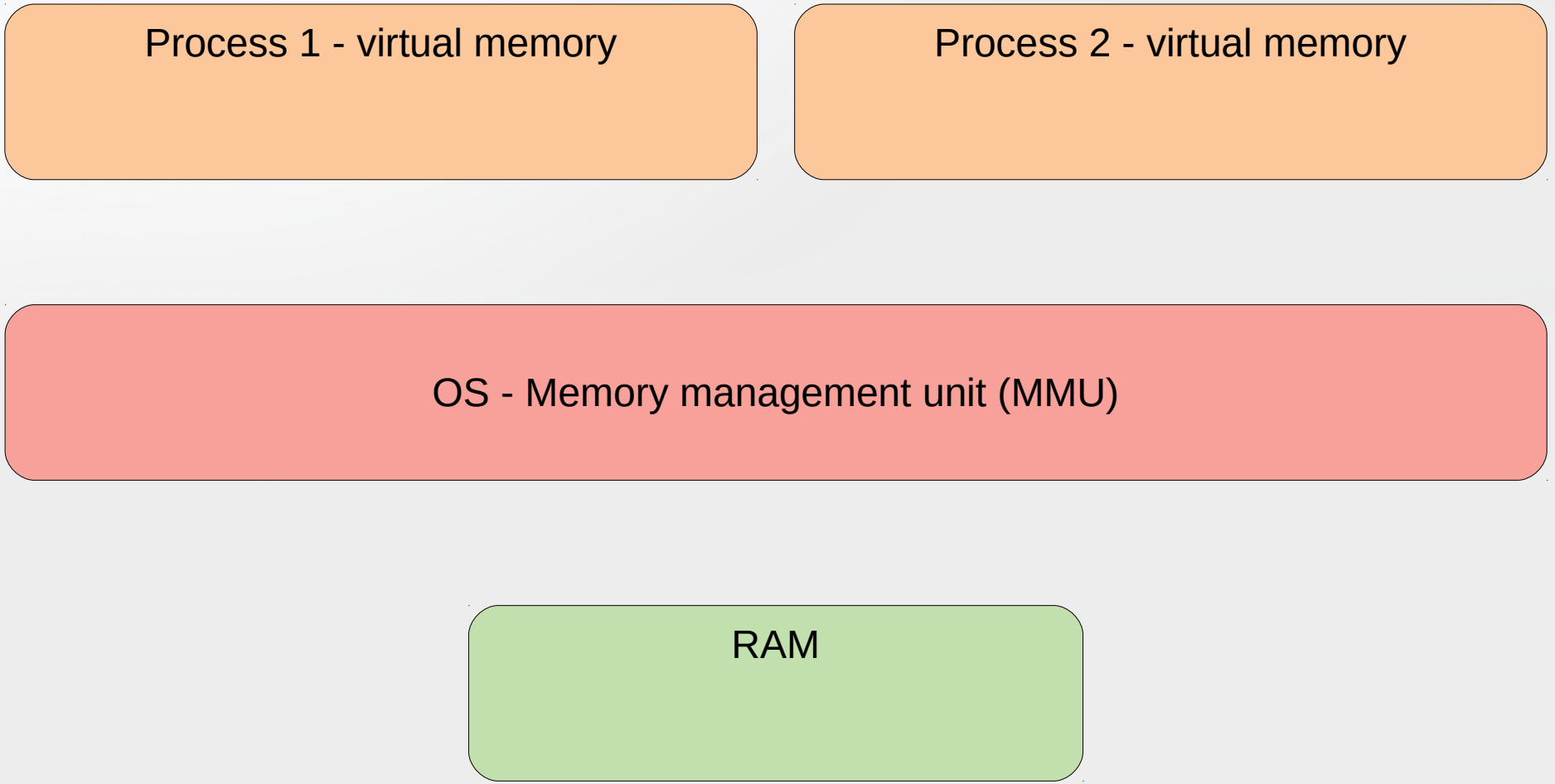
Jak się zjada RAM?



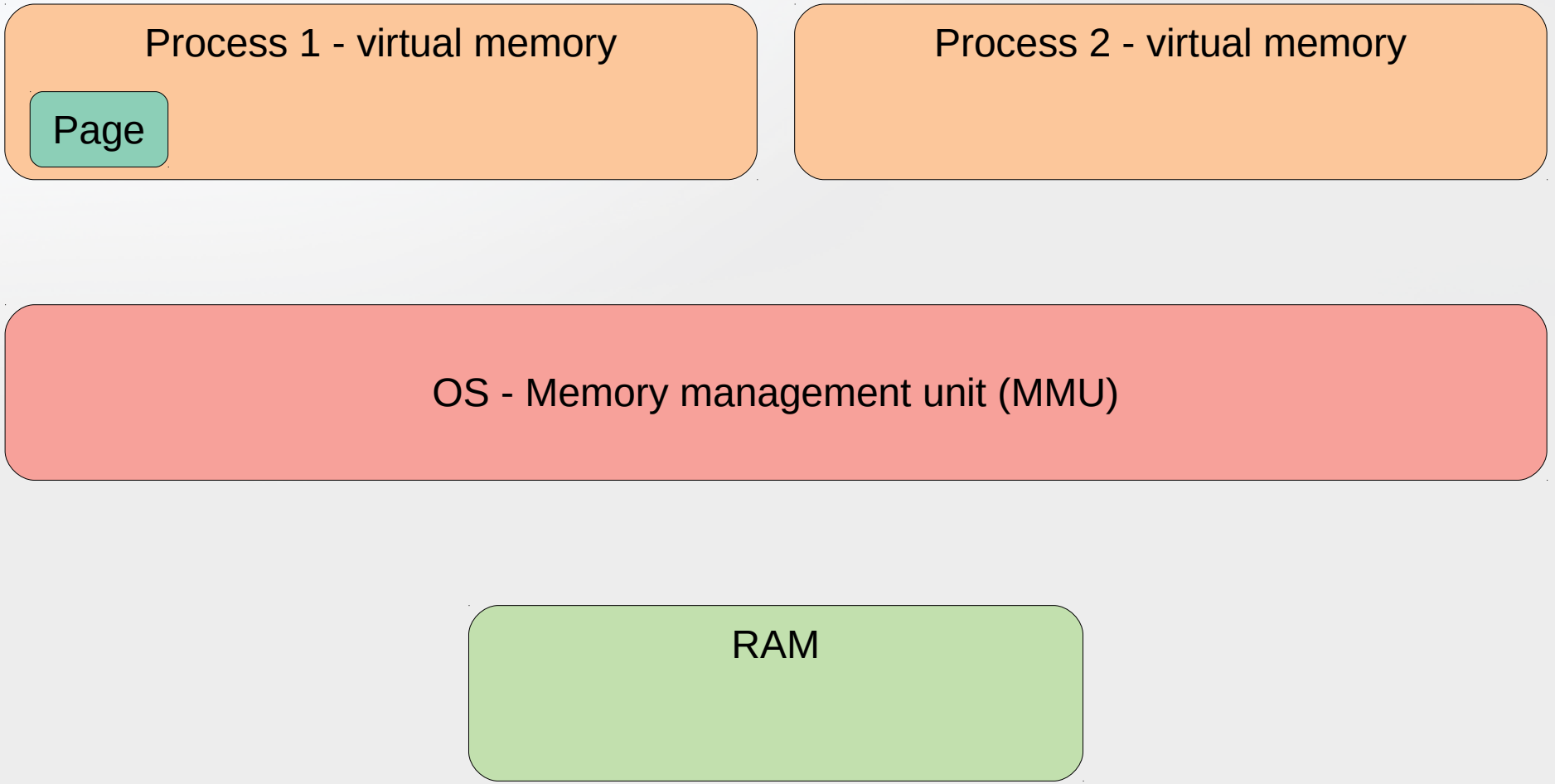
Jak się zjada RAM?



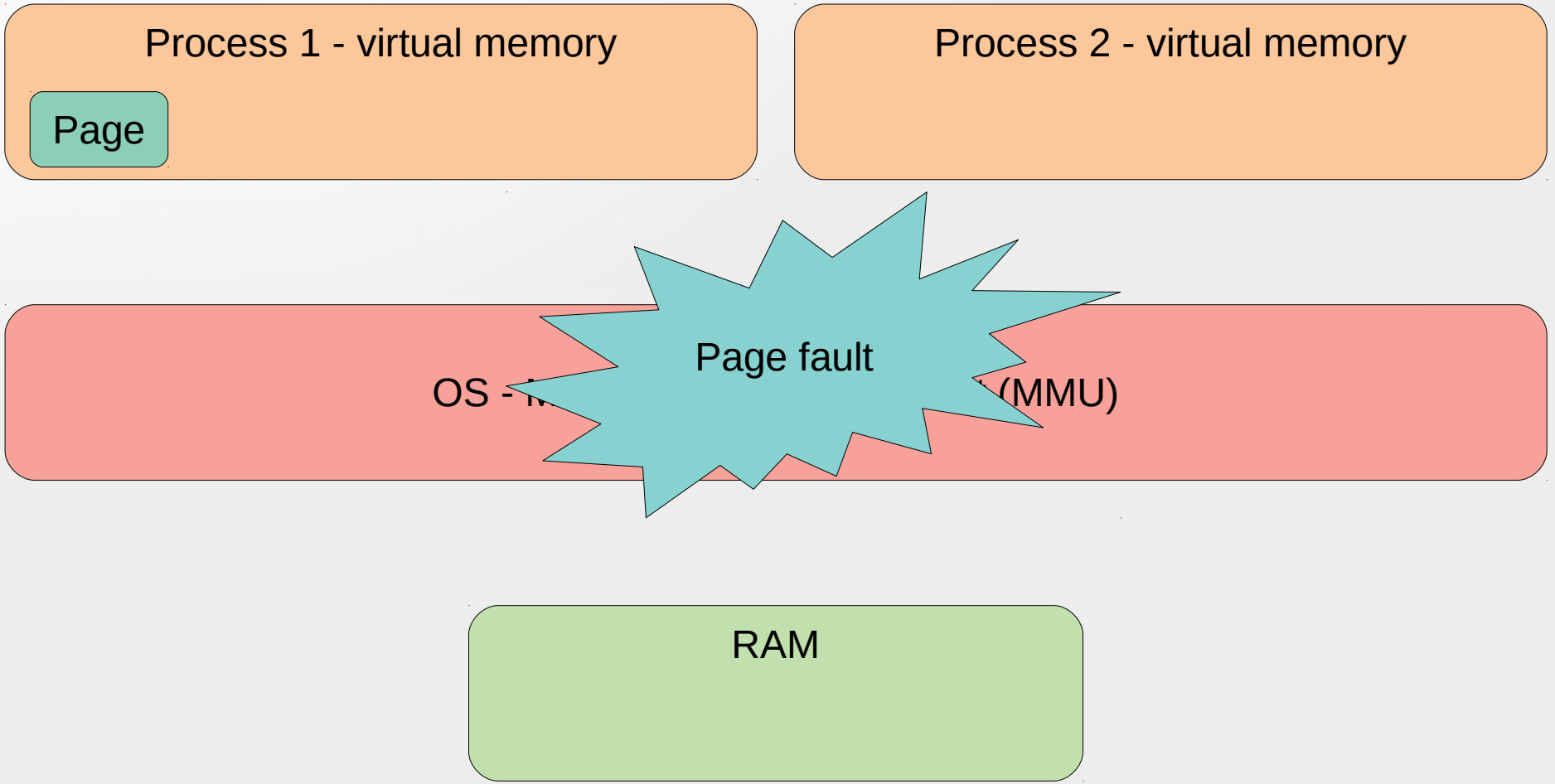
Jak się zjada RAM?



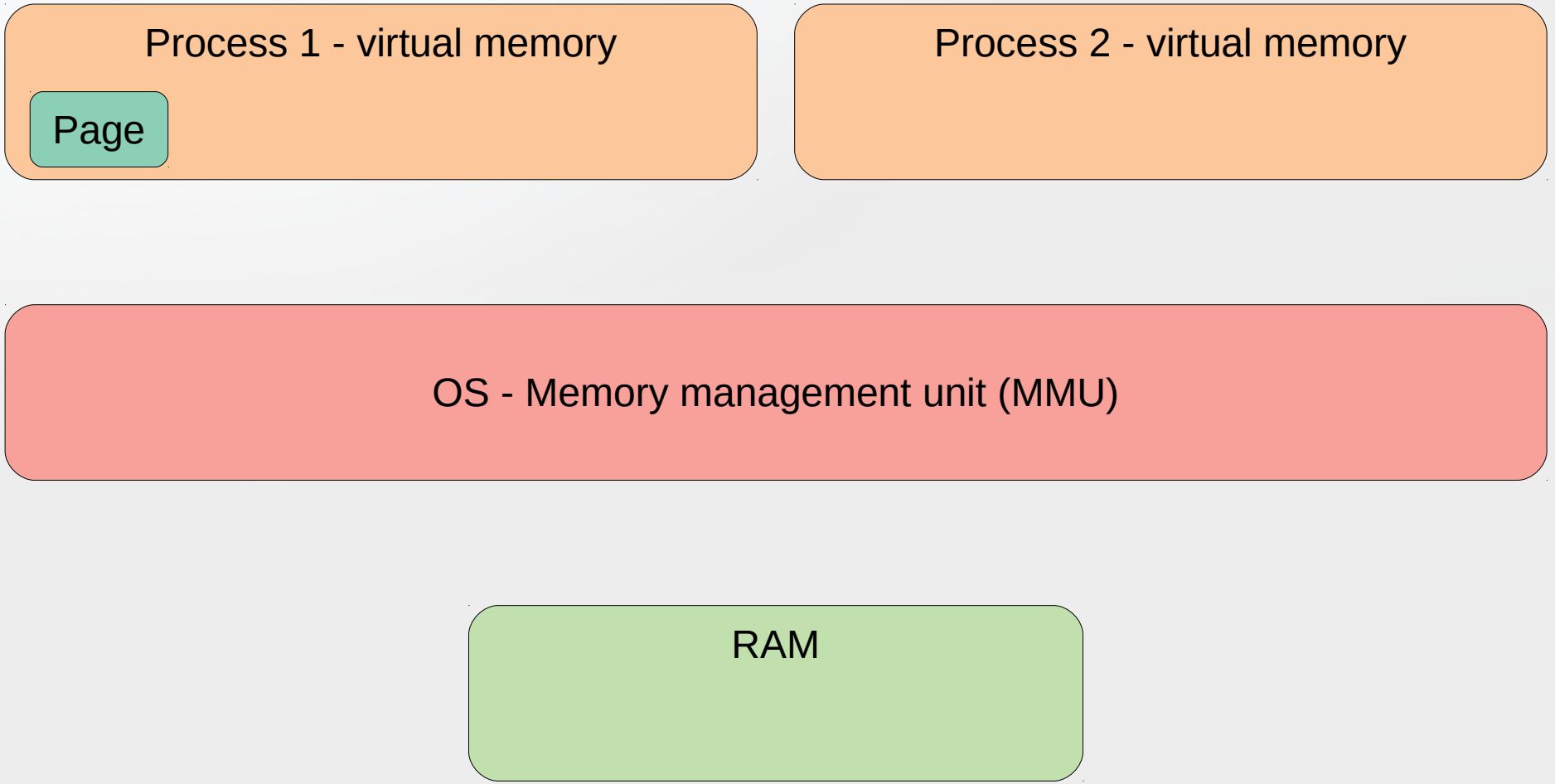
Jak się zjada RAM?



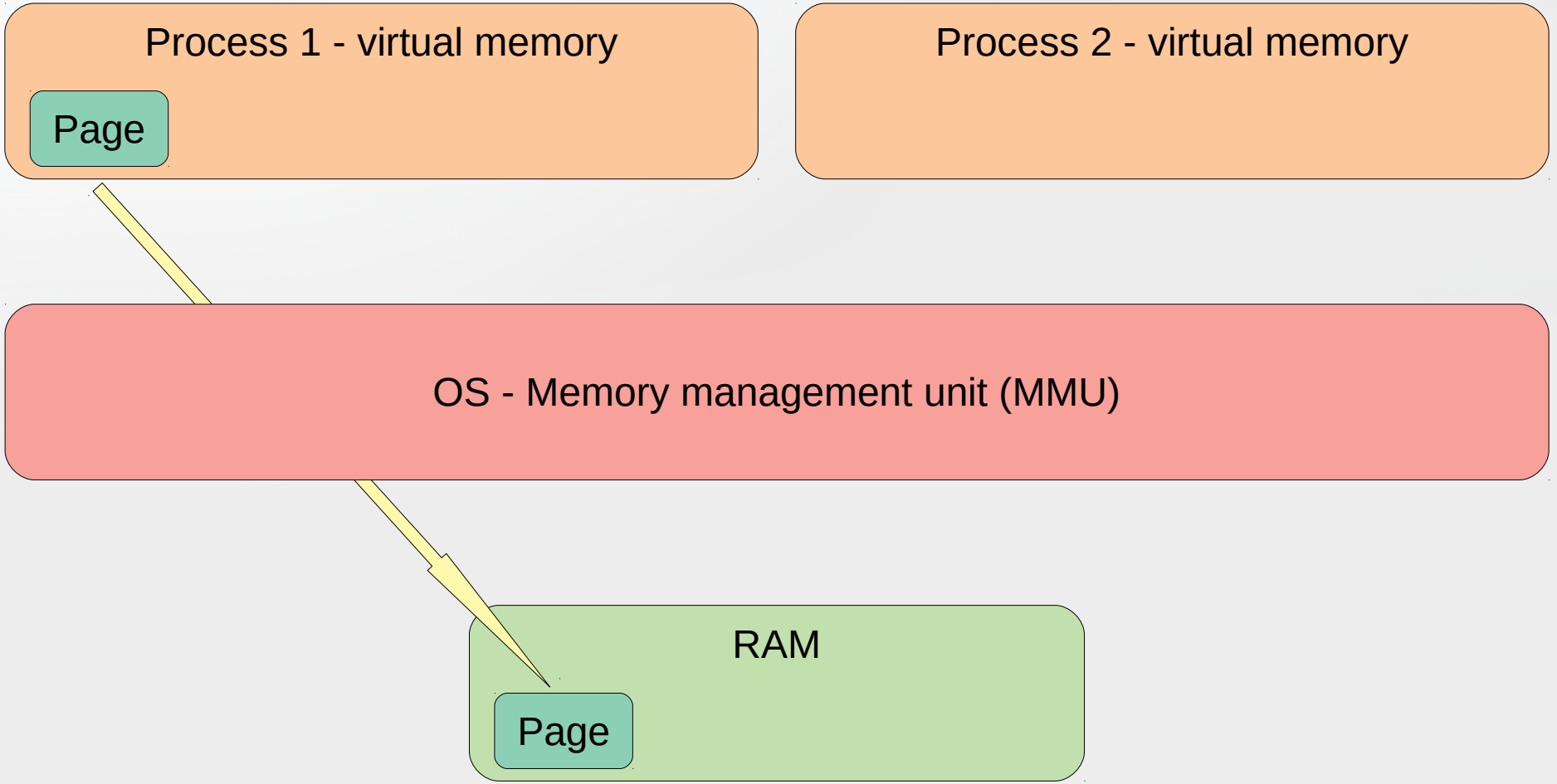
Jak się zjada RAM?



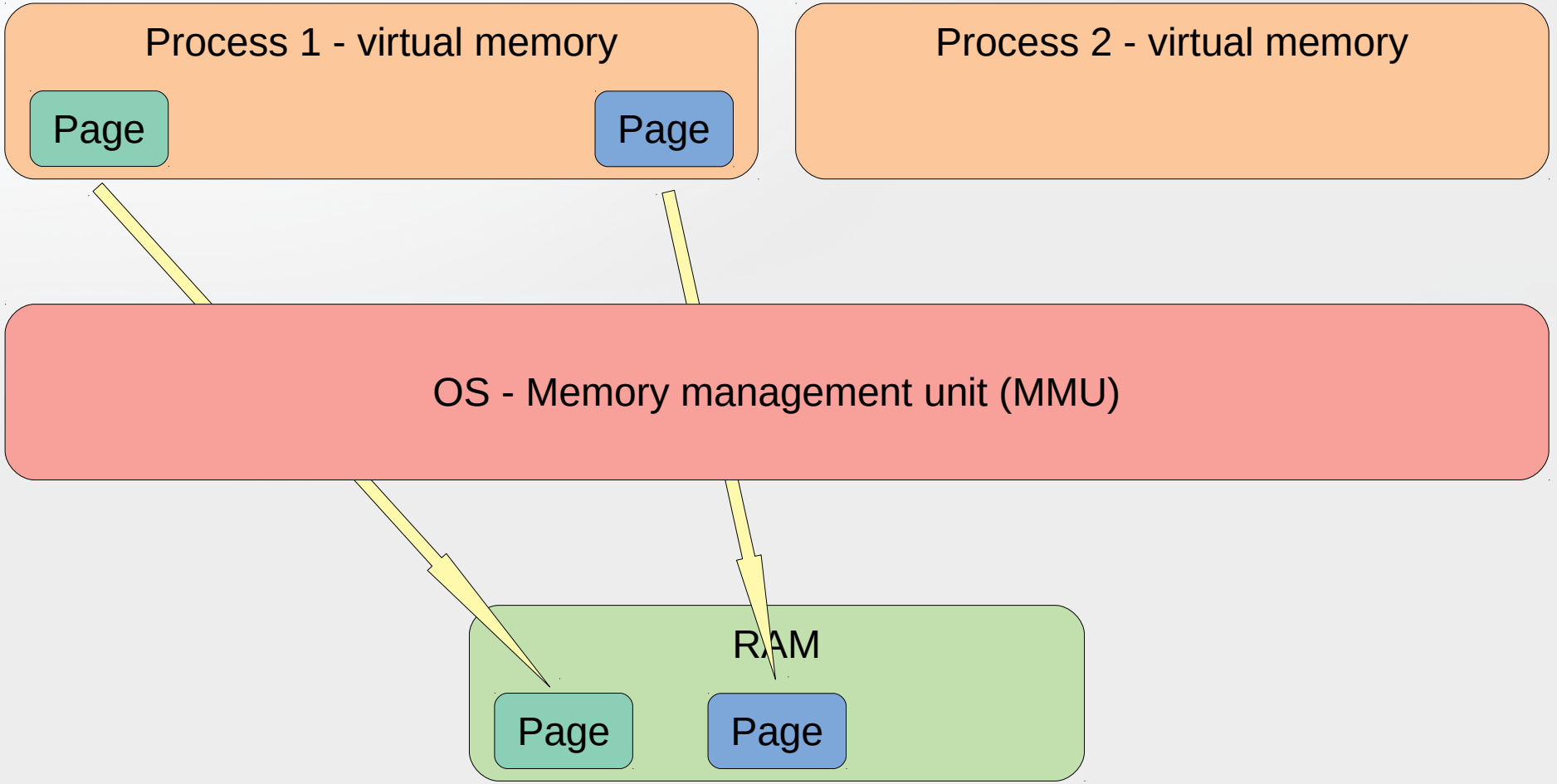
Jak się zjada RAM?



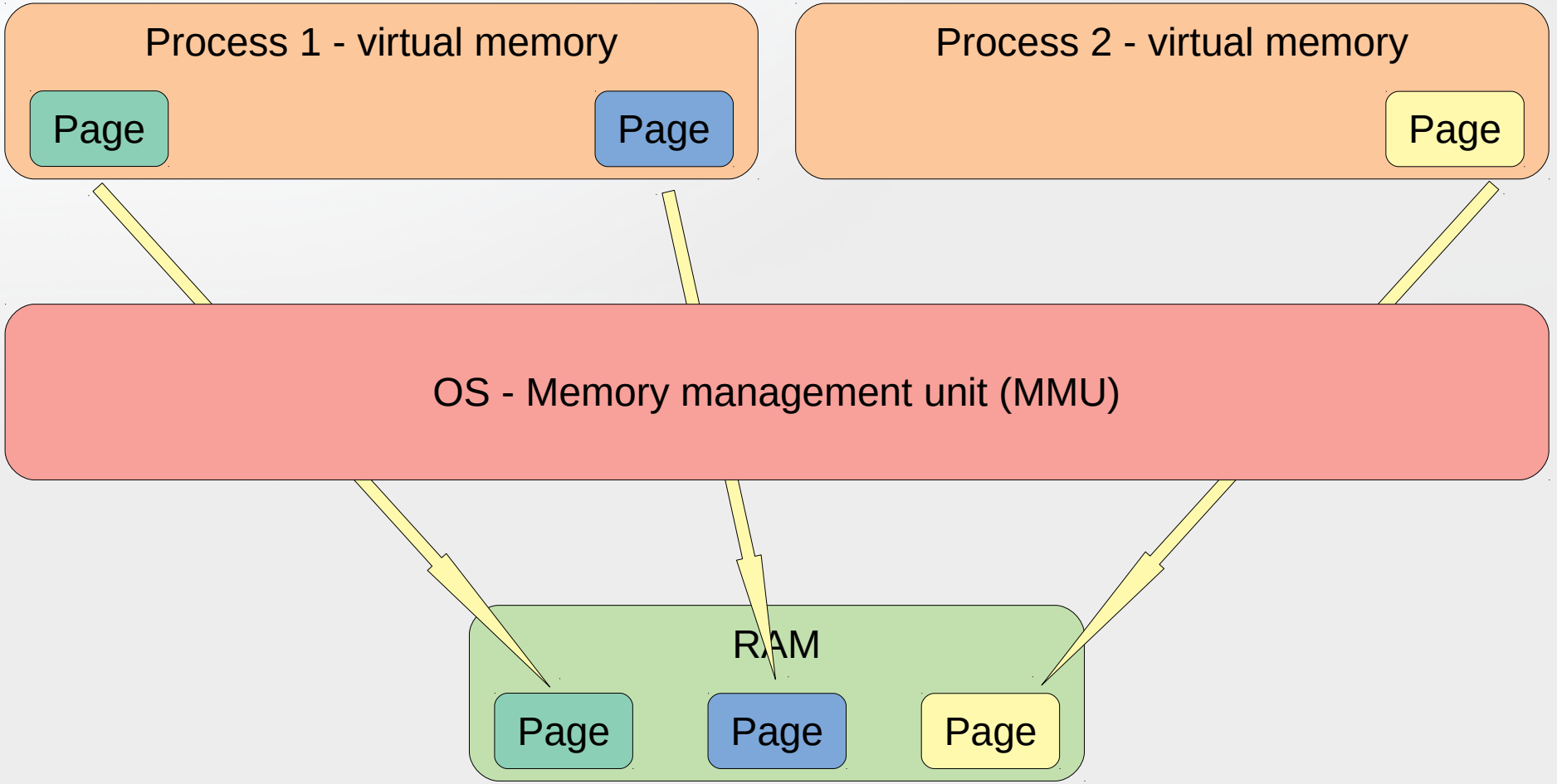
Jak się zjada RAM?



Jak się zjada RAM?



Jak się zjada RAM?



Off-heap allocation – przykład

Złote rady wujka Krzyśka

Złote rady wujka Krzyśka

- Daj szansę FlameGraphom

Złote rady wujka Krzyśka

- Daj szansę FlameGraphom
- Daj szansę Async-profilerowi

Złote rady wujka Krzyśka

- Daj szansę FlameGraphom
- Daj szansę Async-profilerowi
  - Znajdziesz błąd --> zgłoś

Złote rady wujka Krzyśka

- Daj szansę FlameGraphom
- Daj szansę Async-profilerowi
  - Znajdziesz błąd --> zgłoś
  - Masz pomysł na poprawkę --> zgłoś

Złote rady wujka Krzyśka

- Daj szansę FlameGraphom
- Daj szansę Async-profilerowi
  - Znajdziesz błąd --> zgłoś
  - Masz pomysł na poprawkę --> zgłoś
- Dużo cierpliwości

Feedback please

http://jug2020-2.kś.pl/

http://kś.pl
http://gclogs.com

conf@kś.pl
ks@gclogs.com

Dziękuję

http://jug2020-2.kś.pl/