



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání



Programování a algoritmizace

Mocnina a faktoriál
Iterační implementace

Strategický projekt UTB ve Zlíně, reg. č. CZ.02.2.69/0.0/0.0/16_015/0002204



Ing. et Ing. Erik Král, Ph.D.
ÚPKS
Fakulta aplikované informatiky

Obsah

Algoritmus pro výpočtu mocniny a faktoriálu

Popis iterační implementace

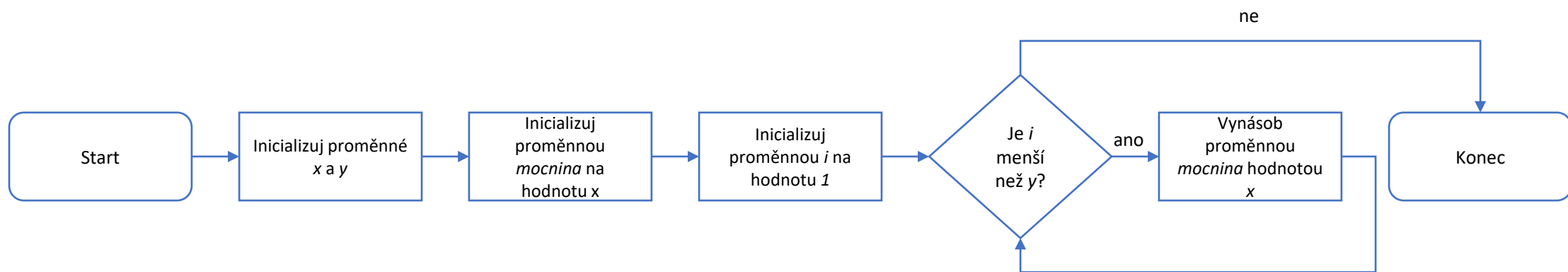
Úvod

- V následujících snímcích probereme algoritmy výpočtu mocniny s přirozeným mocnitelem (exponentem).
- Na těchto příkladech si demonstrujeme práci s cyklem *for* [1].

Mocnina

- Následující algoritmus spočítá hodnotu výrazu x^y
- Nejprve si nadefinujeme proměnnou x představující základ mocniny a proměnnou y představující exponent.
- Poté nadefinujeme proměnnou *mocnina* kterou inicializujeme na hodnotu základu mocniny, tedy proměnné x .
- Následně v cyklu vynásobíme proměnnou *mocnina* $y - 1$ n-krát.

Mocnina



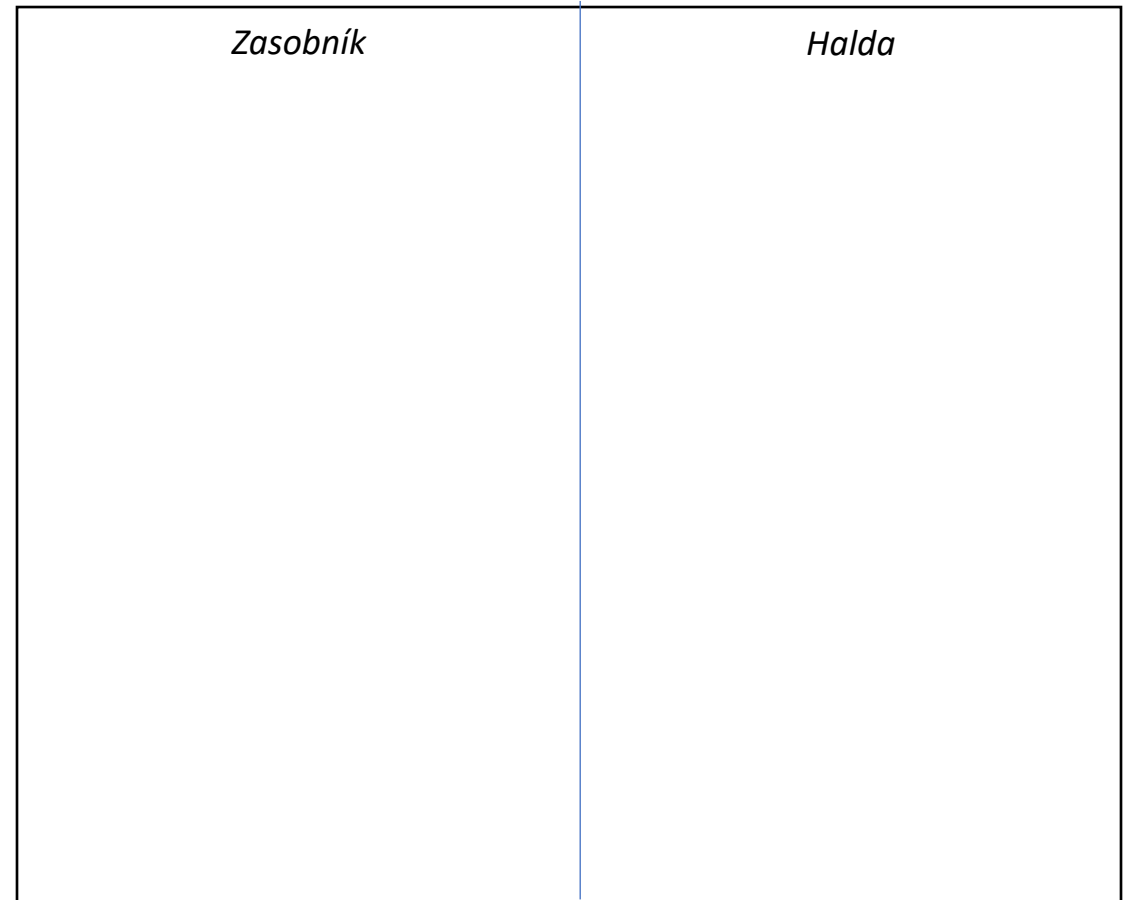
Algoritmus a paměť

- Algoritmus si alokuje paměť pro parametry, lokální proměnné a další hodnoty na zásobníku (Stack) a pro dynamicky alokované objekty alokuje paměť na haldě (Heap).
- V příkladech je **zjednodušeně** demonstrováno využití paměti z hlediska zásobníku a haldy.
- Práce se zásobníkem je ve skutečnosti složitější a v příkladech jsou zobrazeny **pouze proměnné přímo související s algoritmem** a jsou vynechány uložené hodnoty registrů nebo návratové hodnoty. Také pořadí předávaných argumentů a parametrů metody může být jiné.

Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}  
  
Console.WriteLine(x);
```

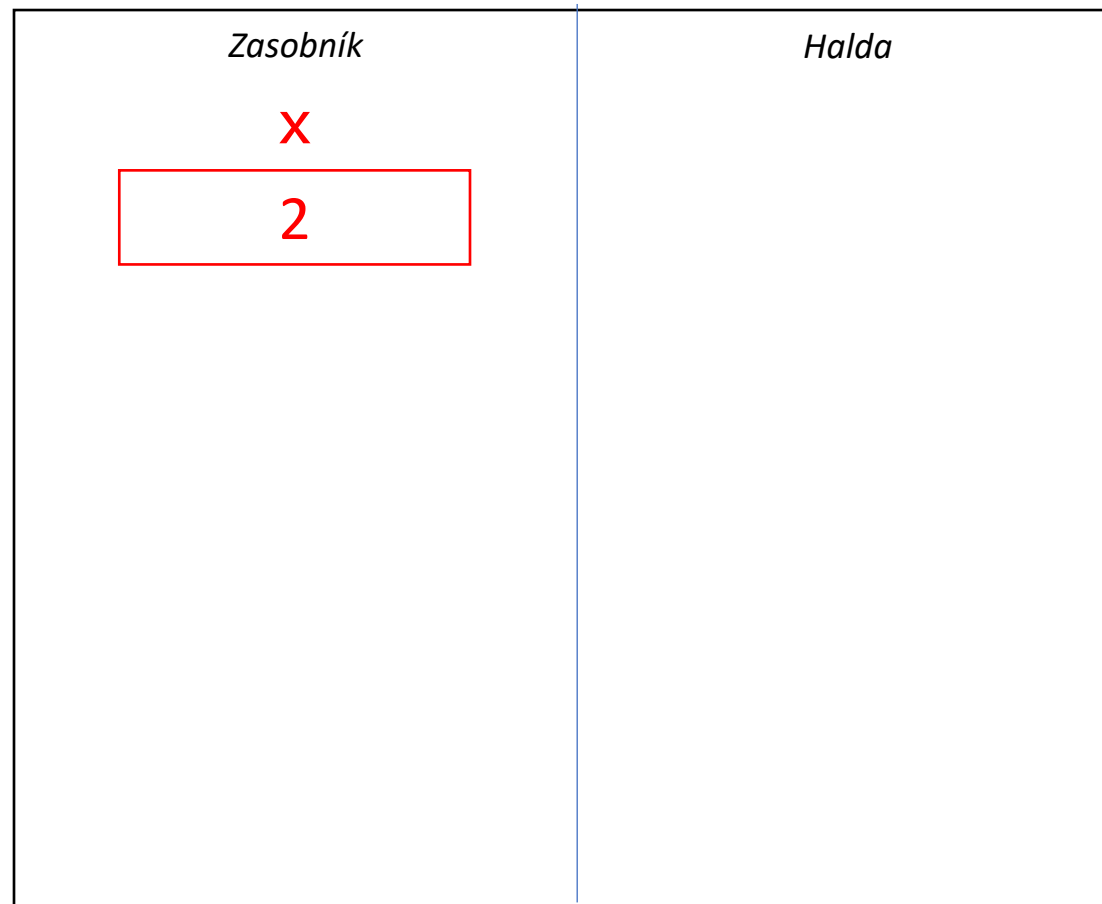
Paměť RAM



Mocnina

```
int x = 2;
```

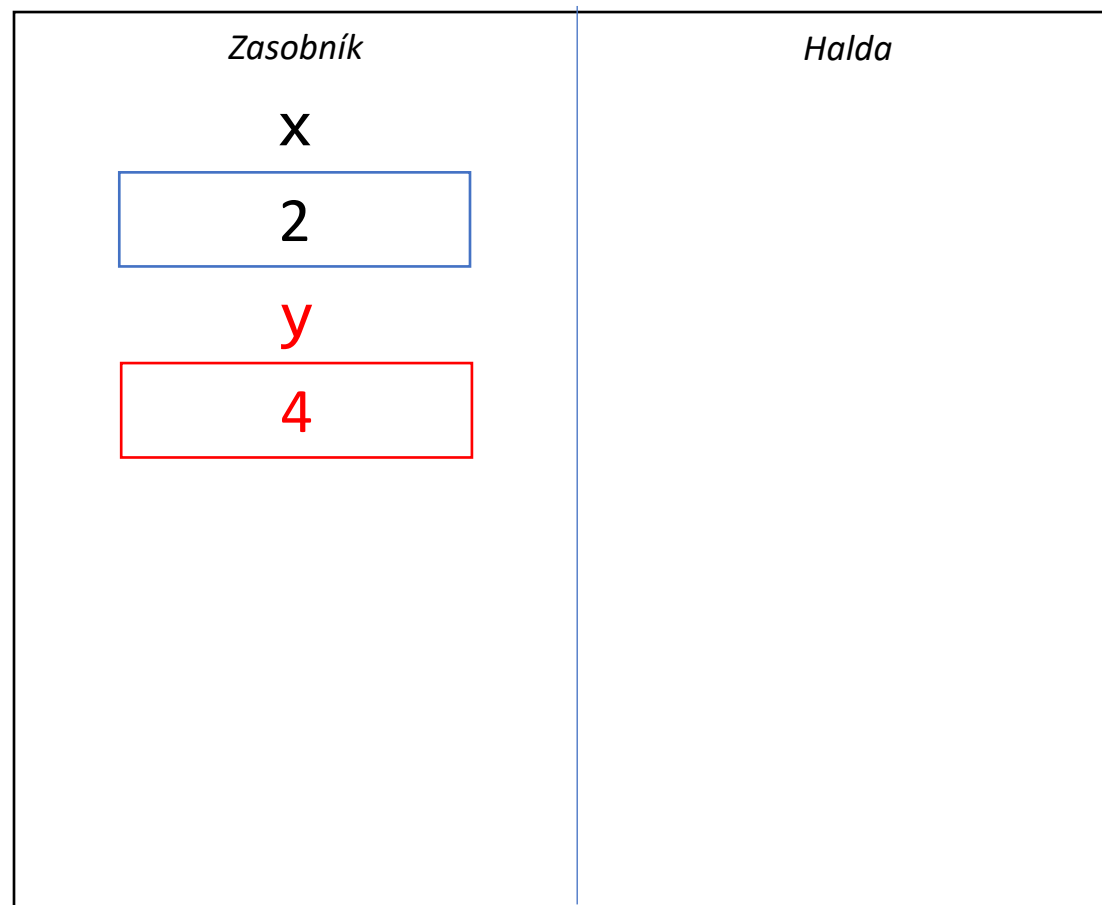
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;
```

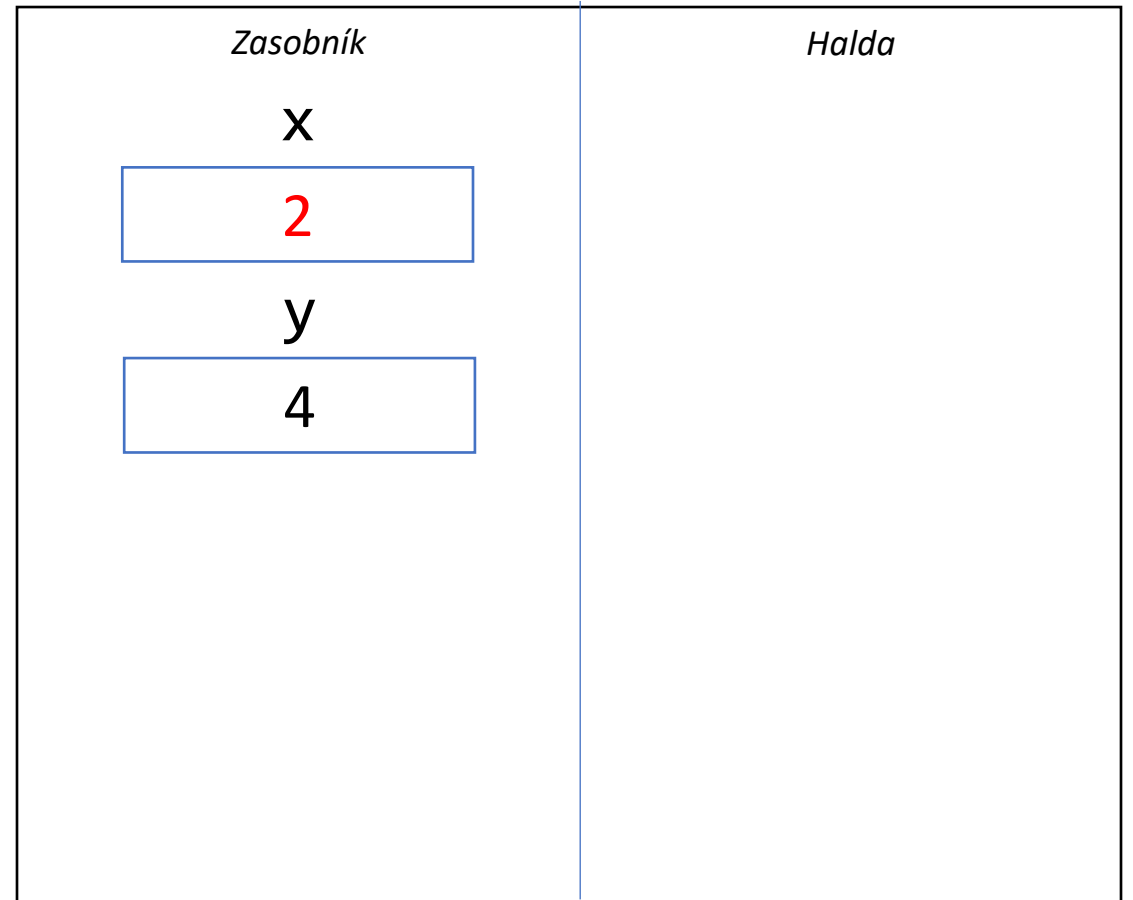
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;
```

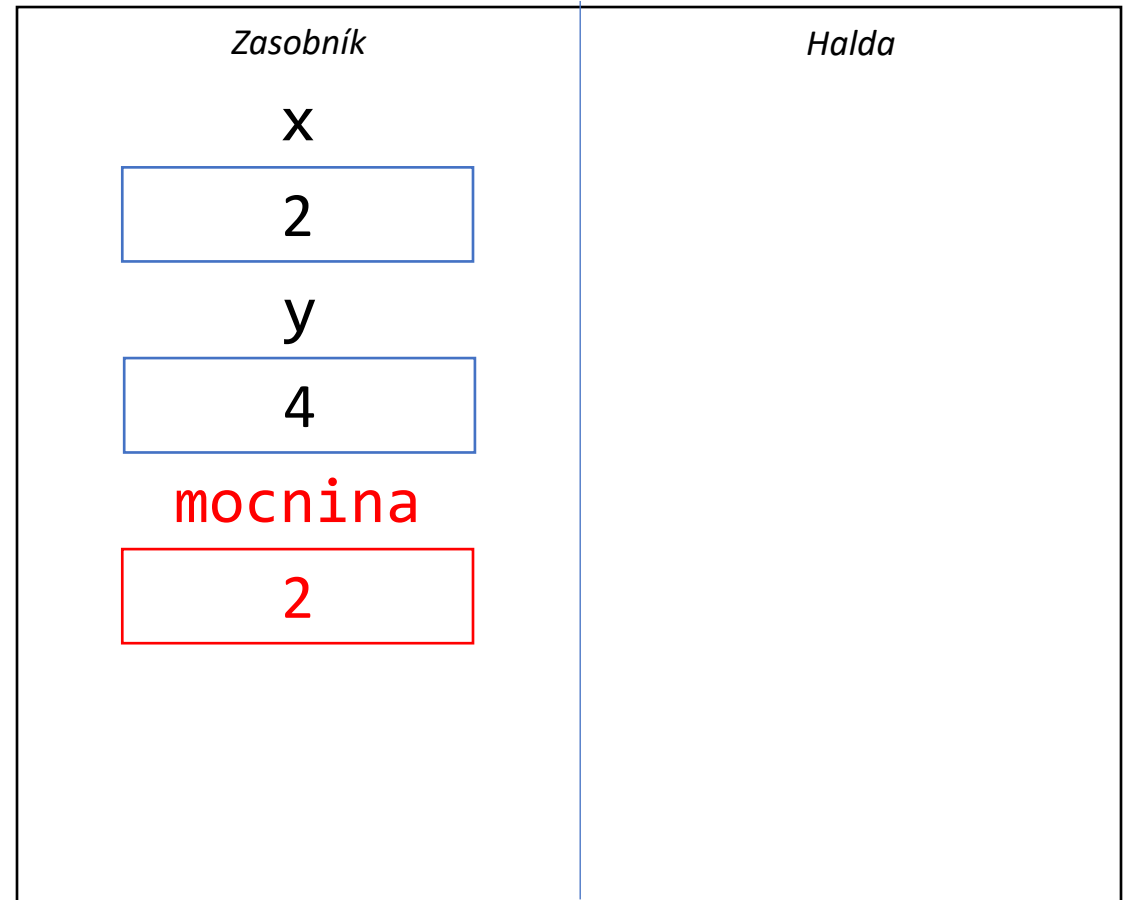
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;
```

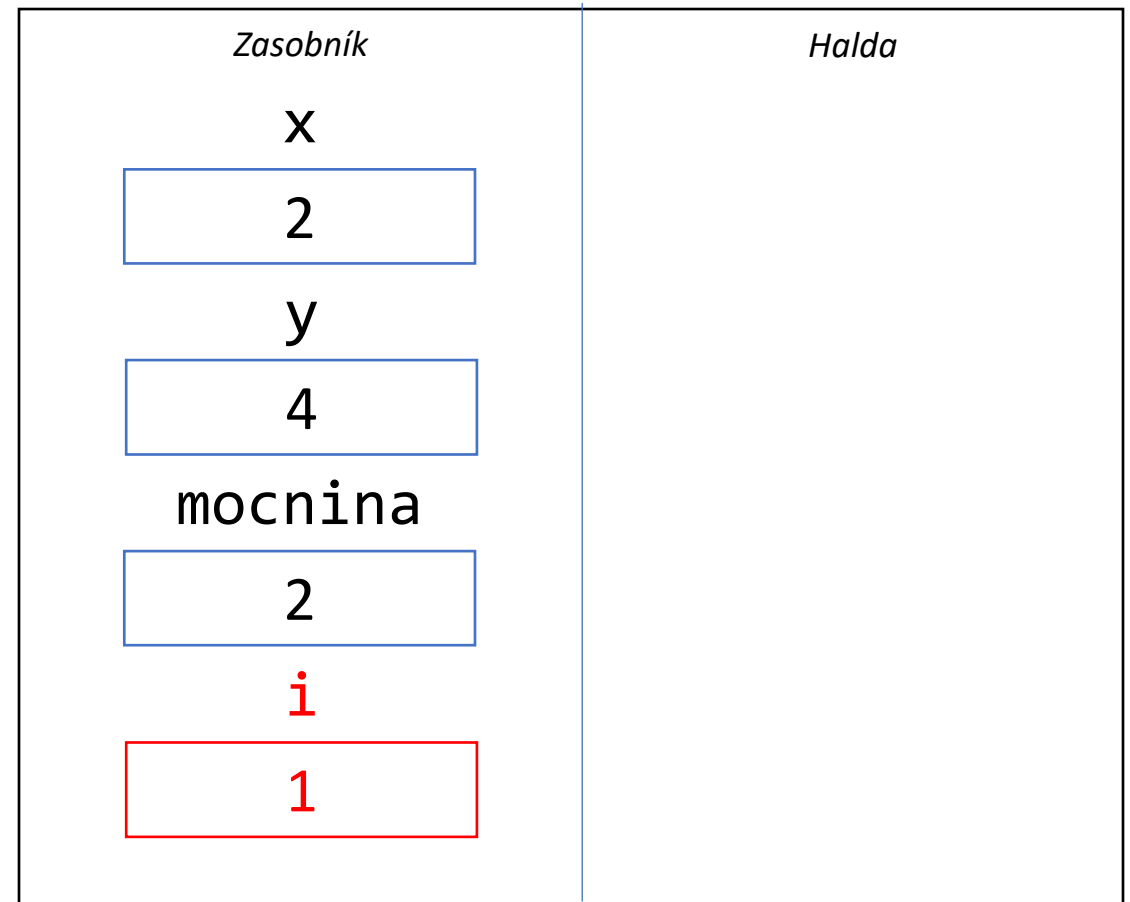
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
  
}
```

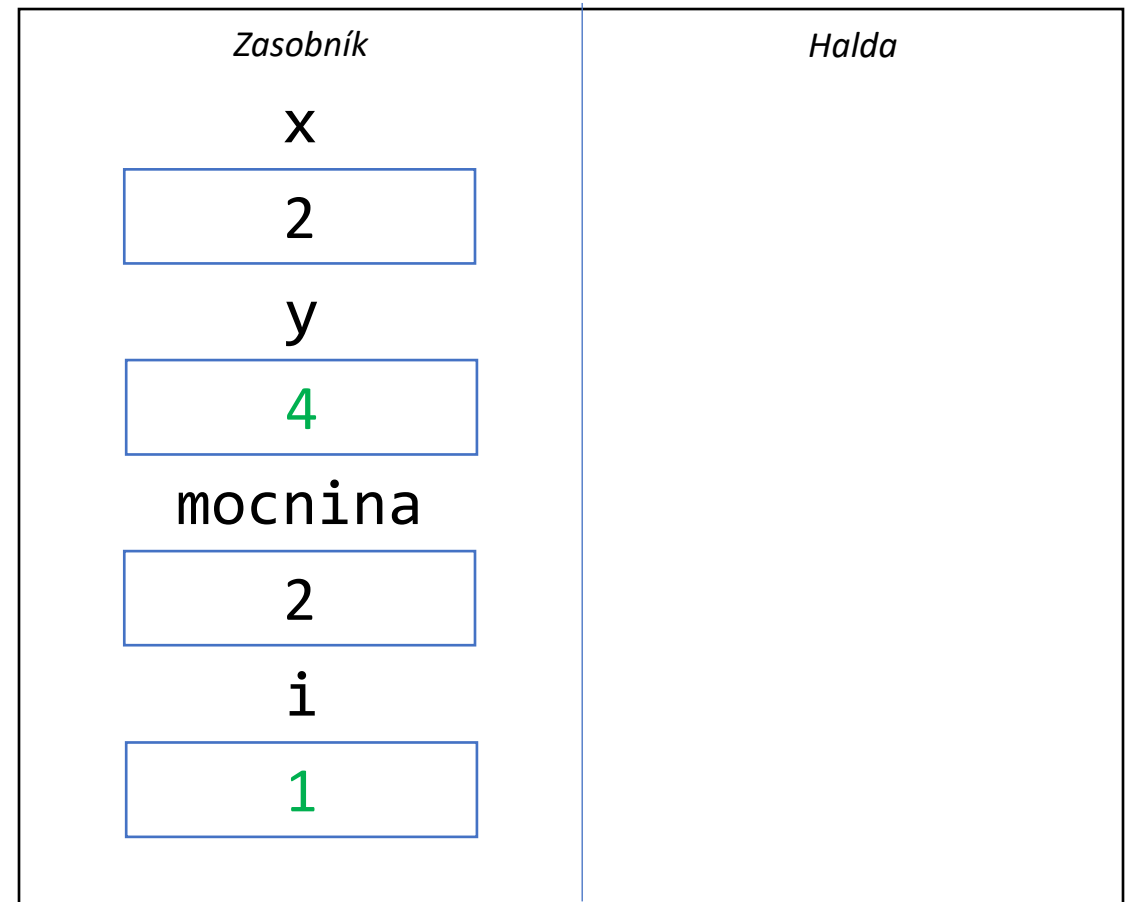
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
for (int i = 1; i < y; i++)  
{  
  
}
```

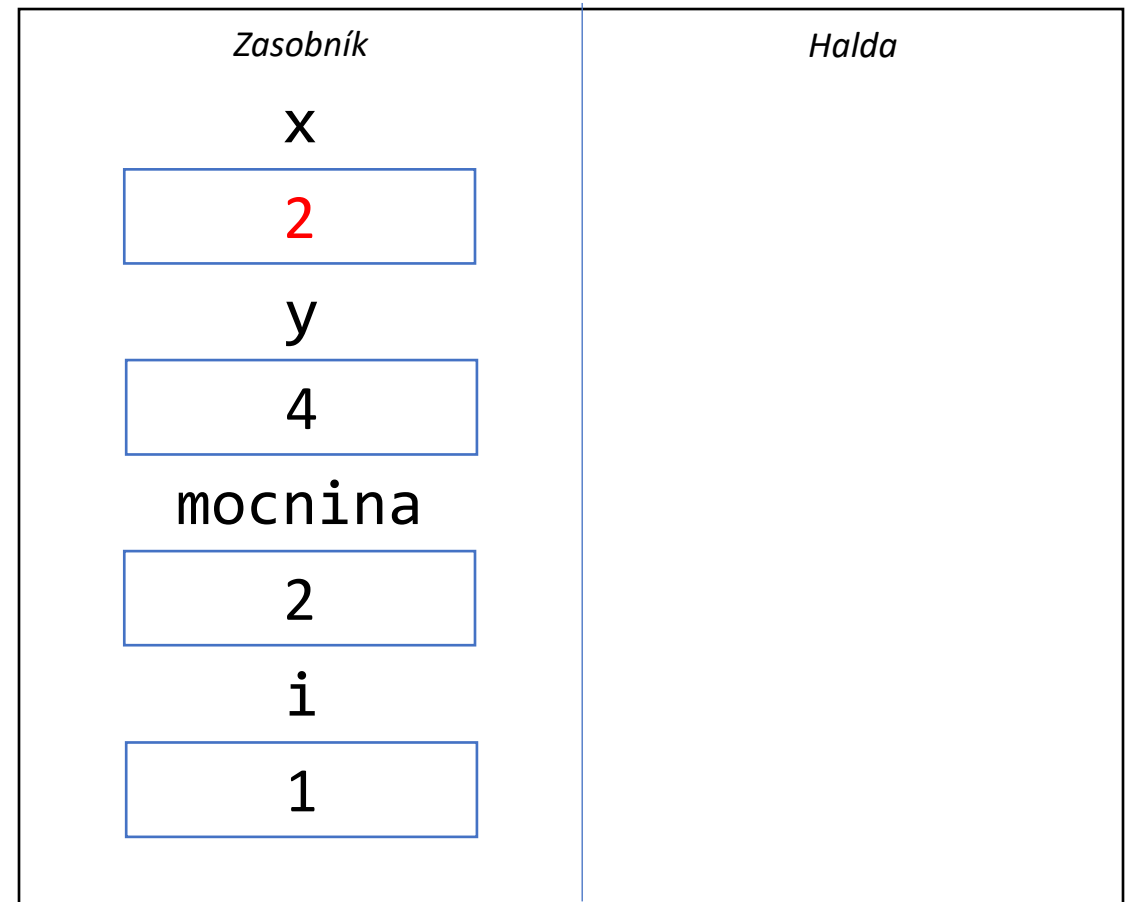
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

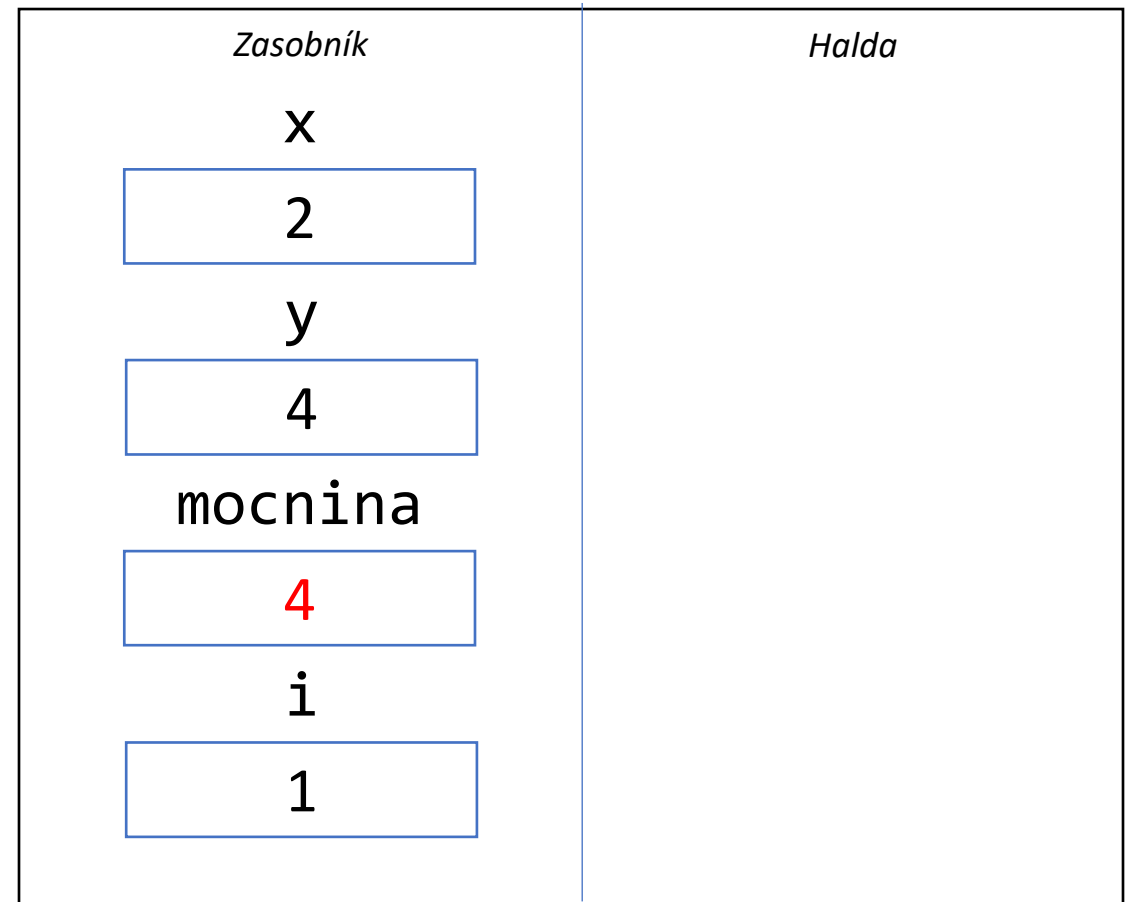
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

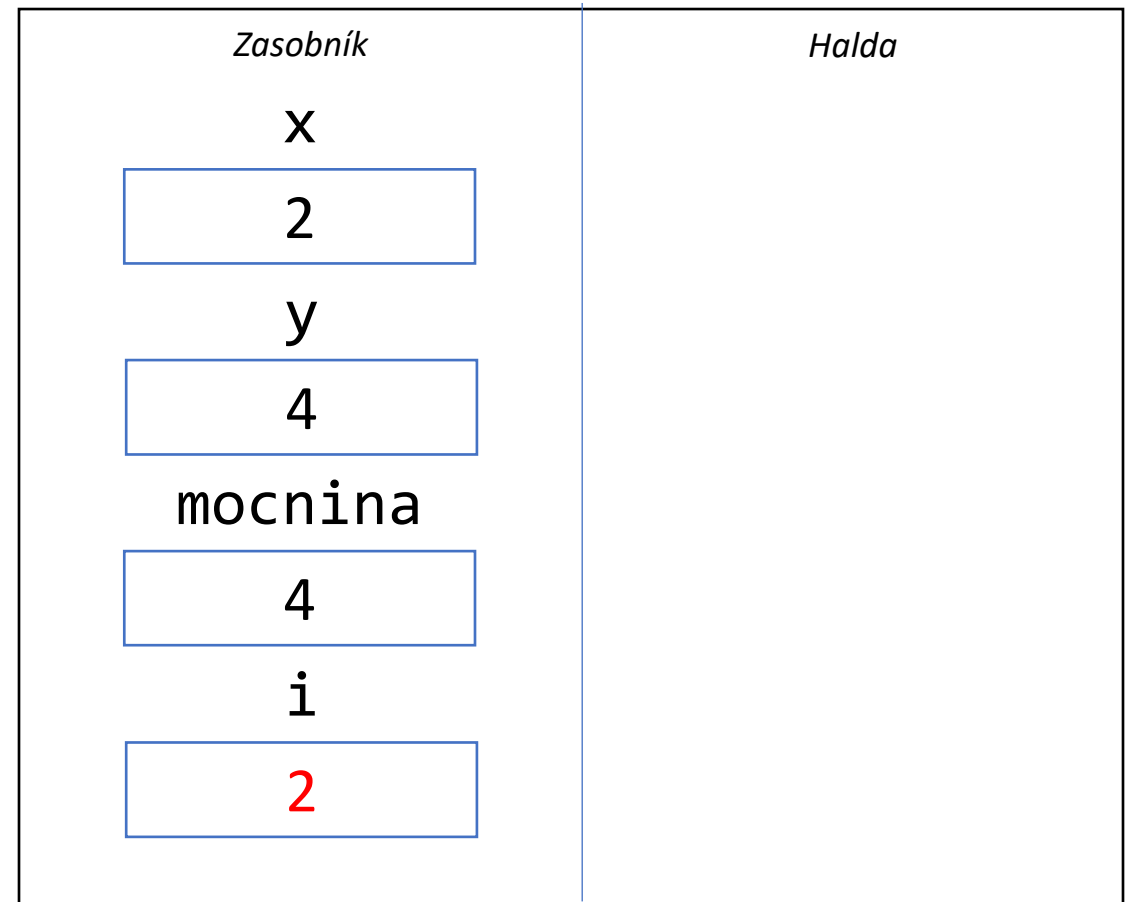
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

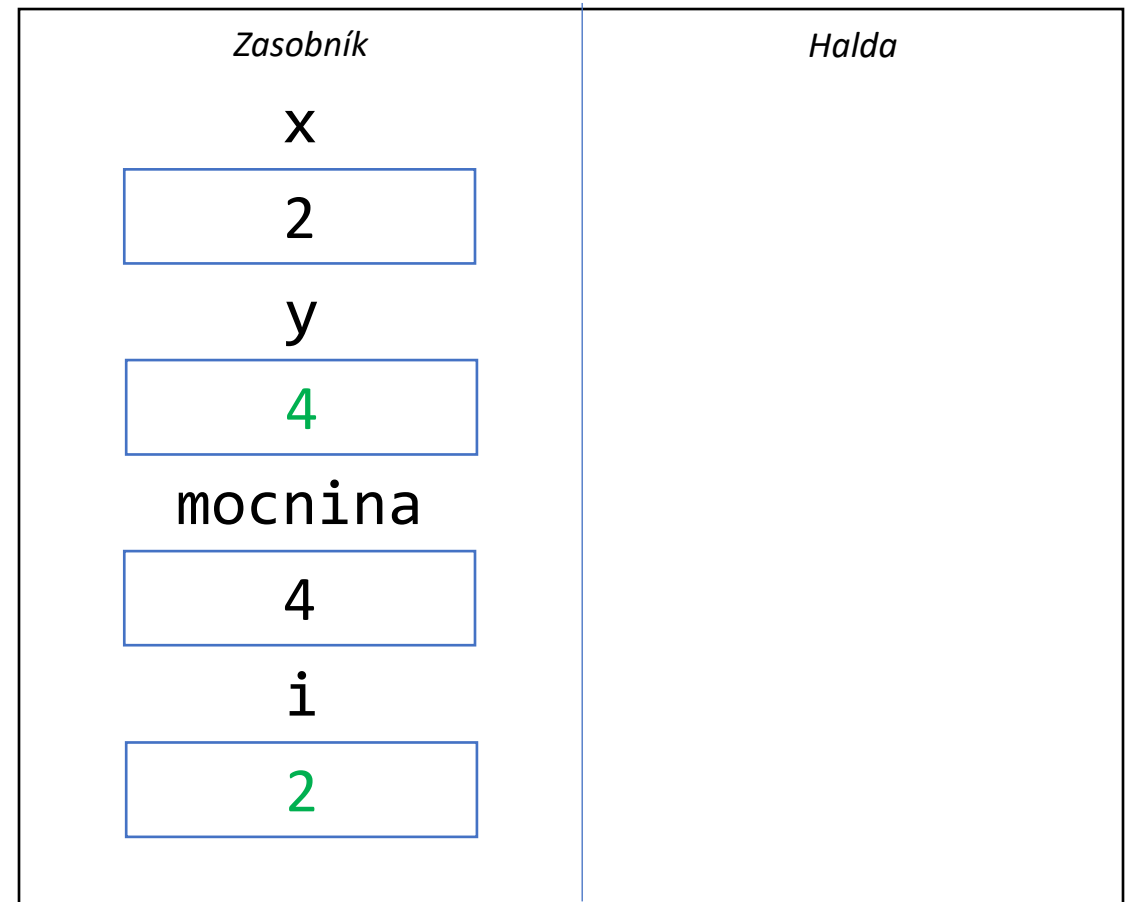
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

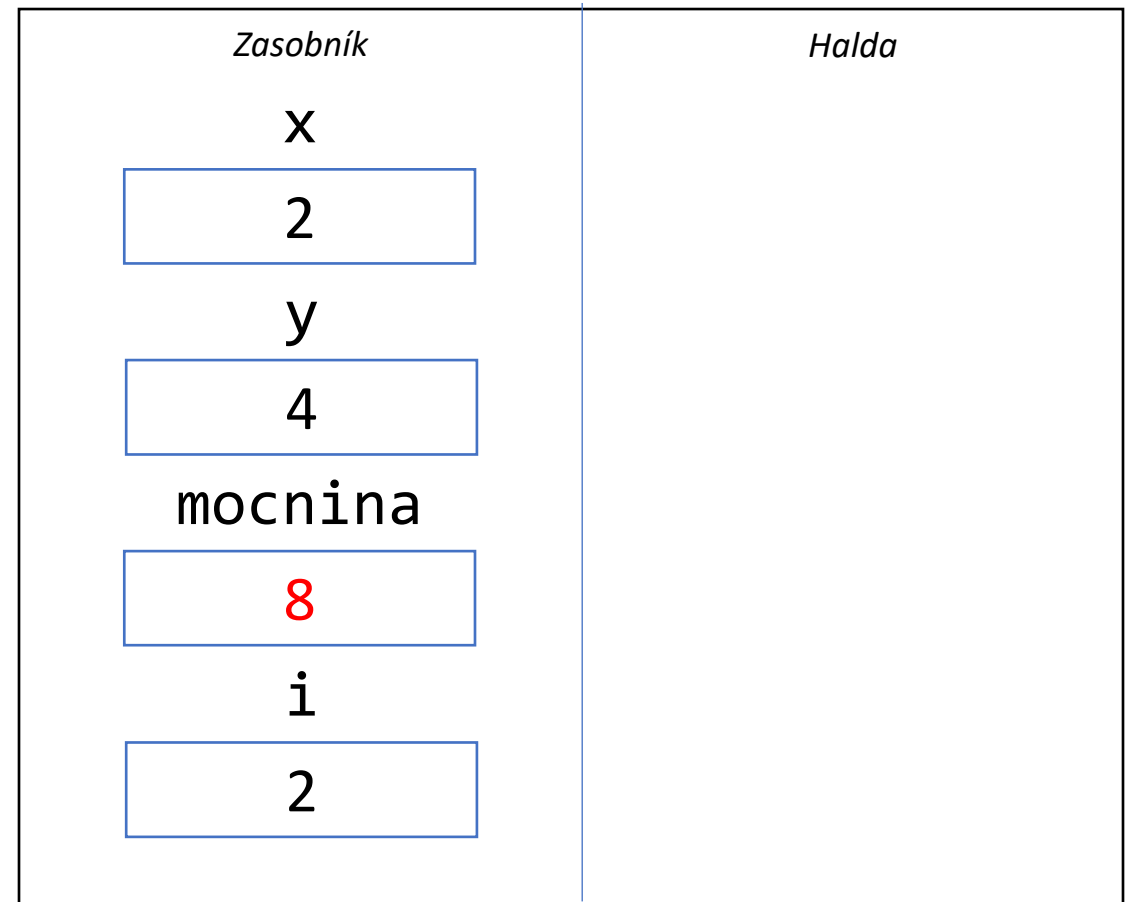
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

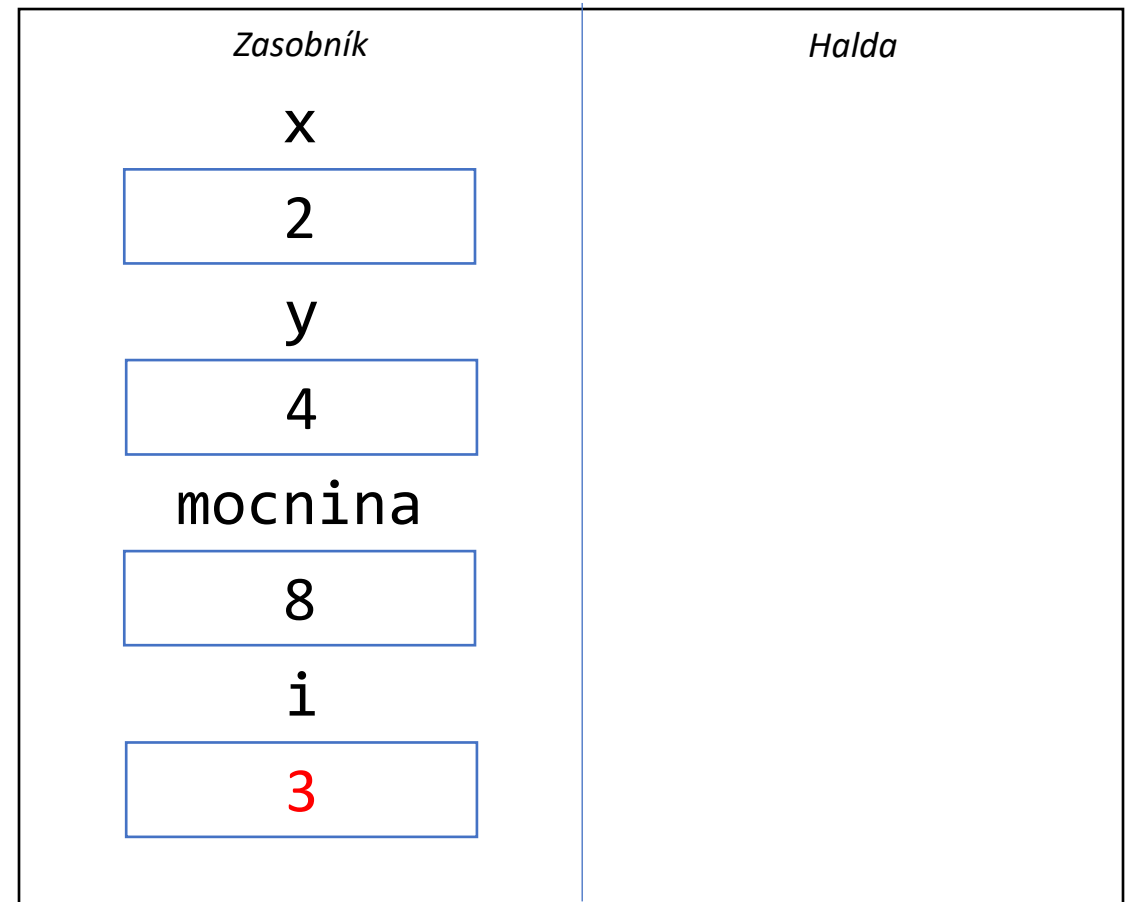
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

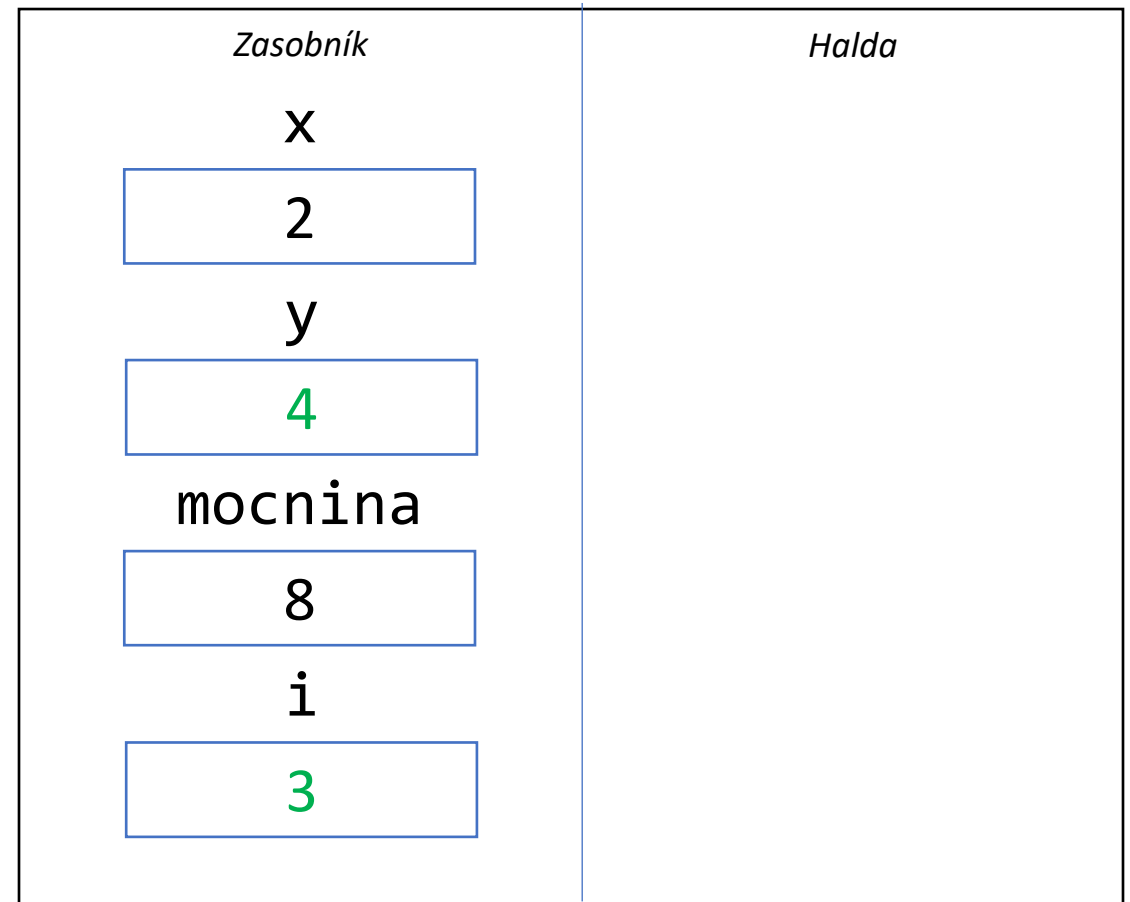
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

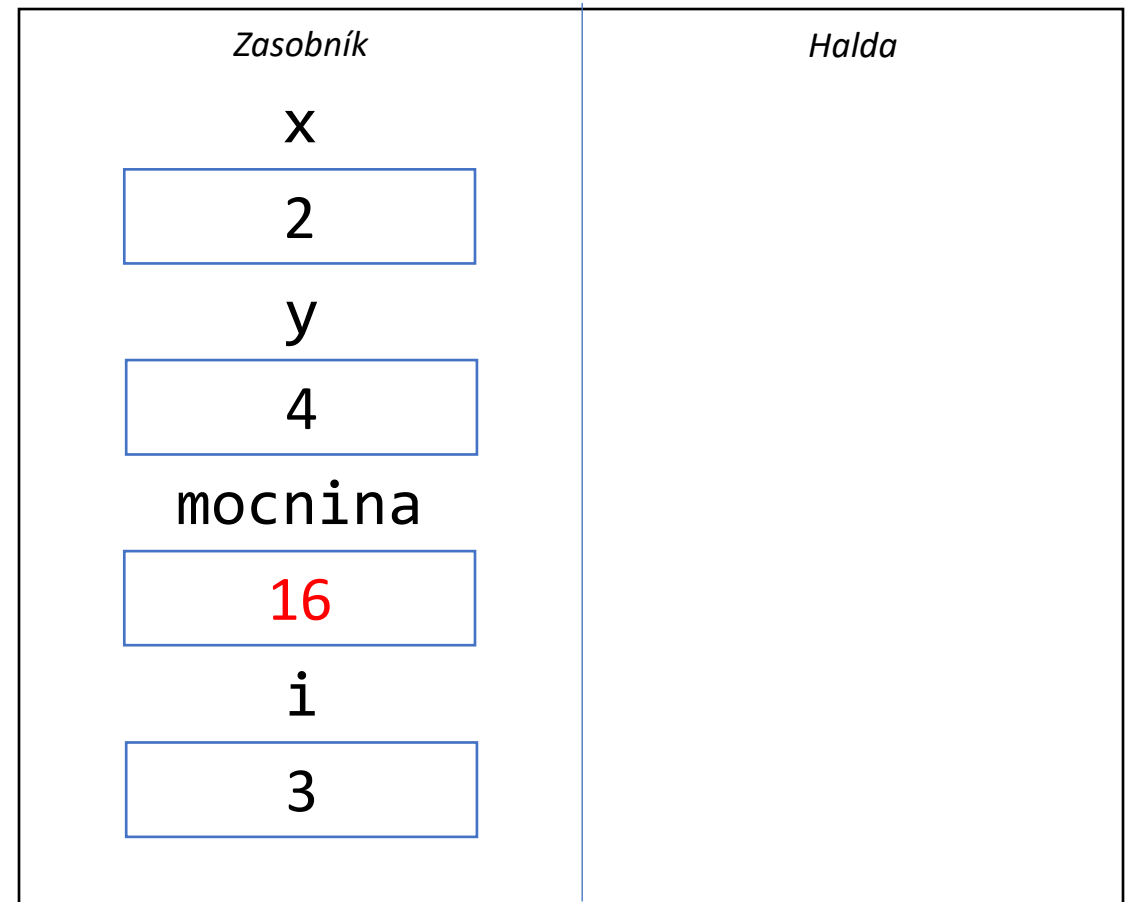
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

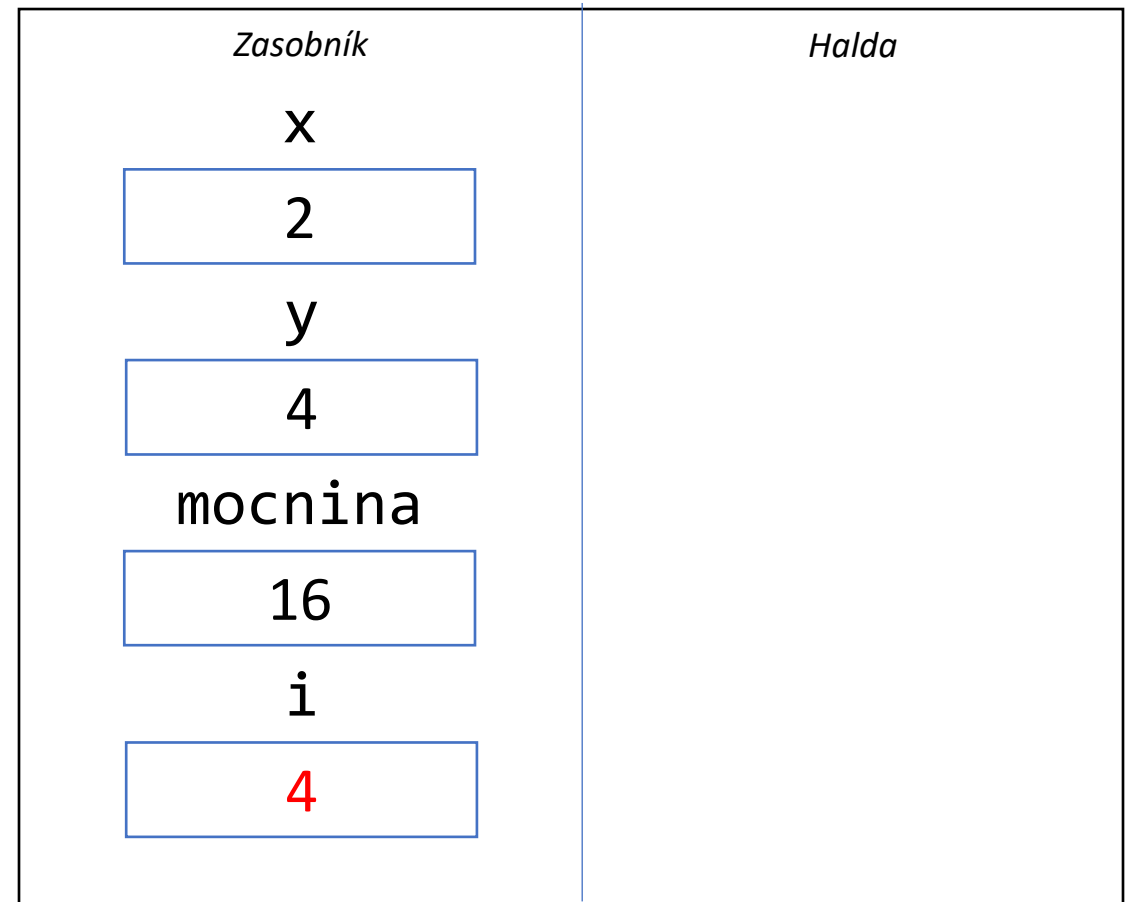
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

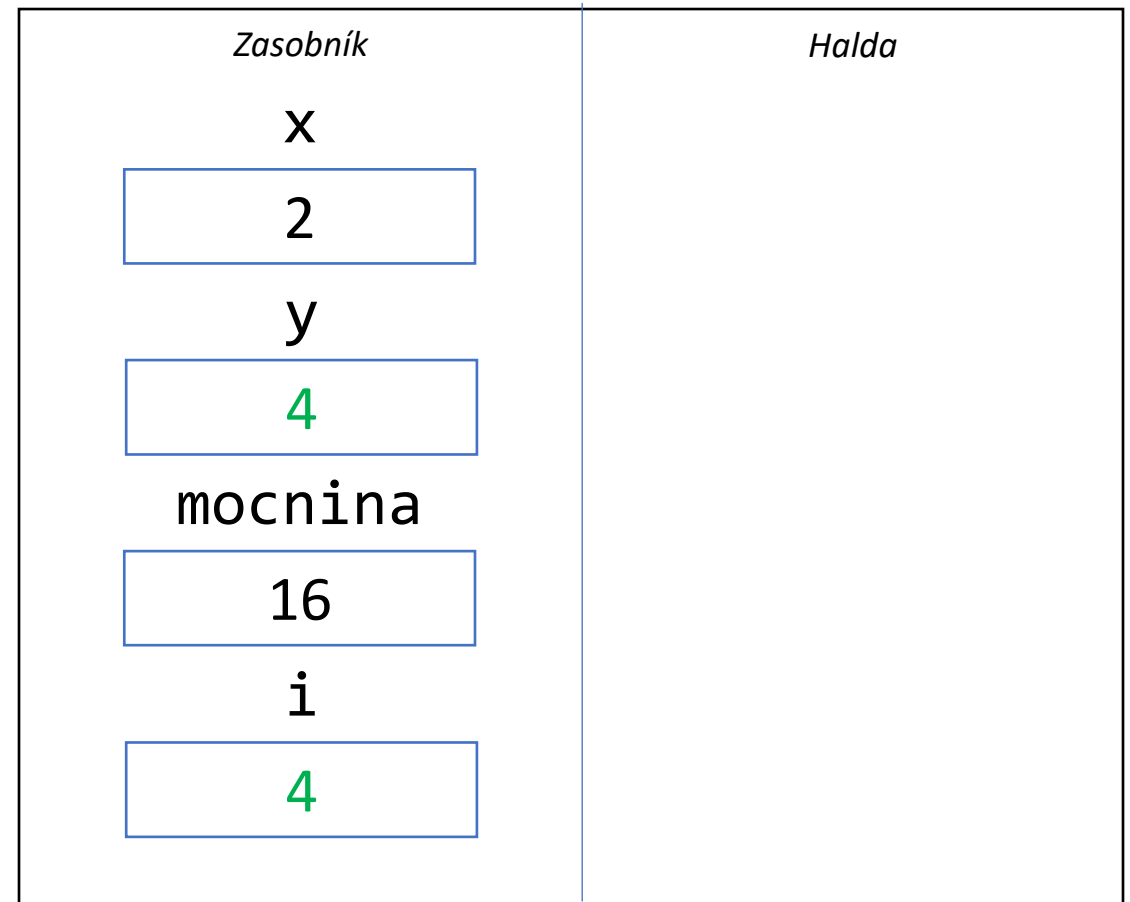
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}
```

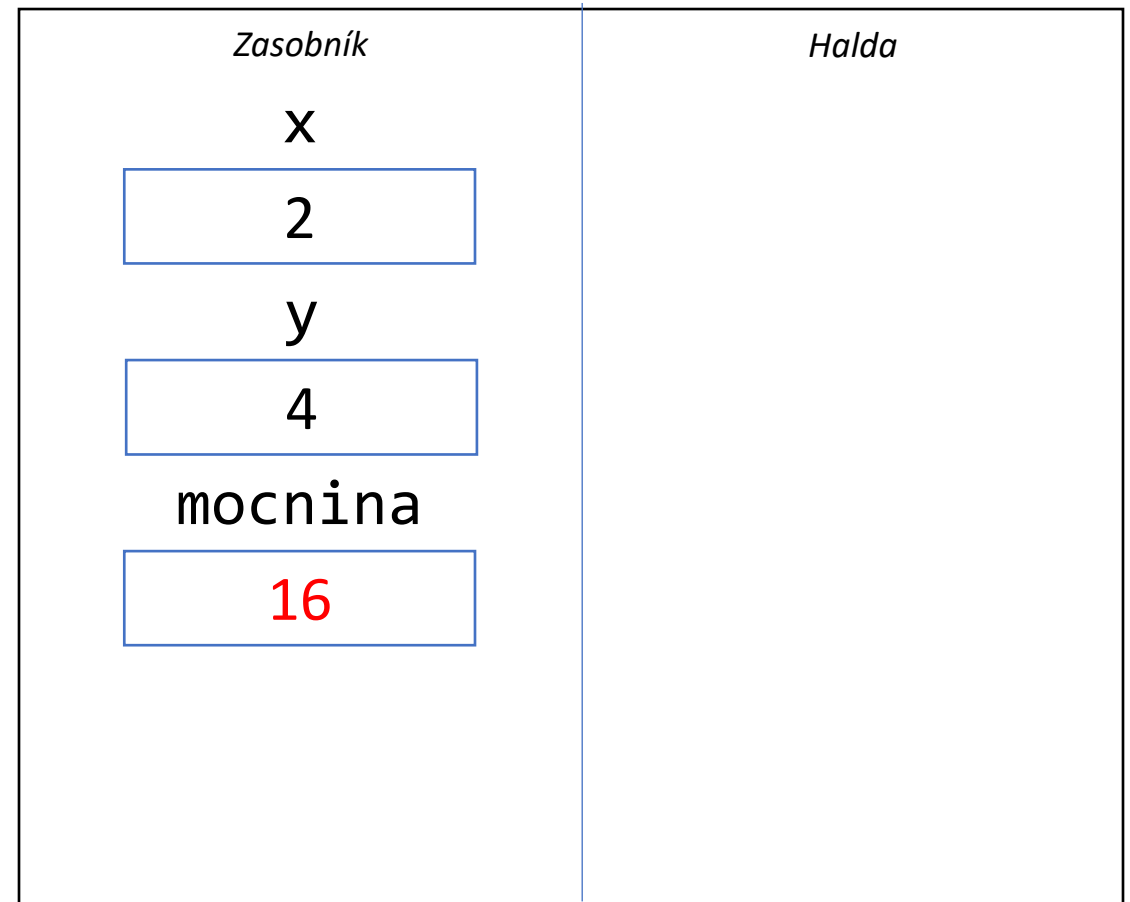
Paměť RAM



Mocnina

```
int x = 2;  
int y = 4;  
  
int mocnina = x;  
  
for (int i = 1; i < y; i++)  
{  
    mocnina *= x;  
}  
  
Console.WriteLine(x);
```

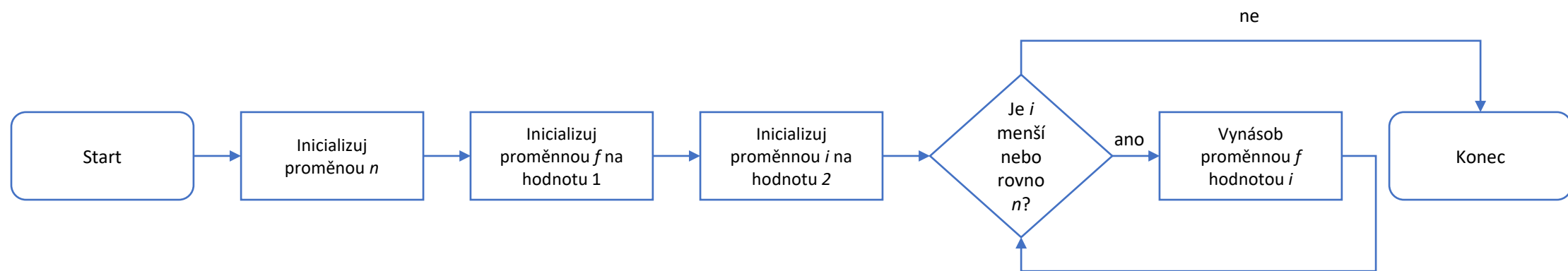
Paměť RAM



Faktoriál

- Následující algoritmus spočítá hodnotu výrazu $n!$
- Nejprve si nadefinujeme proměnnou n a potom proměnnou f inicializovanou na hodnotu 1 , která bude reprezentovat výsledek výpočtu, tedy faktoriál $n!$
- Následně v vynásobíme proměnnou f hodnotami iterační proměnné cyklu *for* z rozsahu 2 až n včetně.

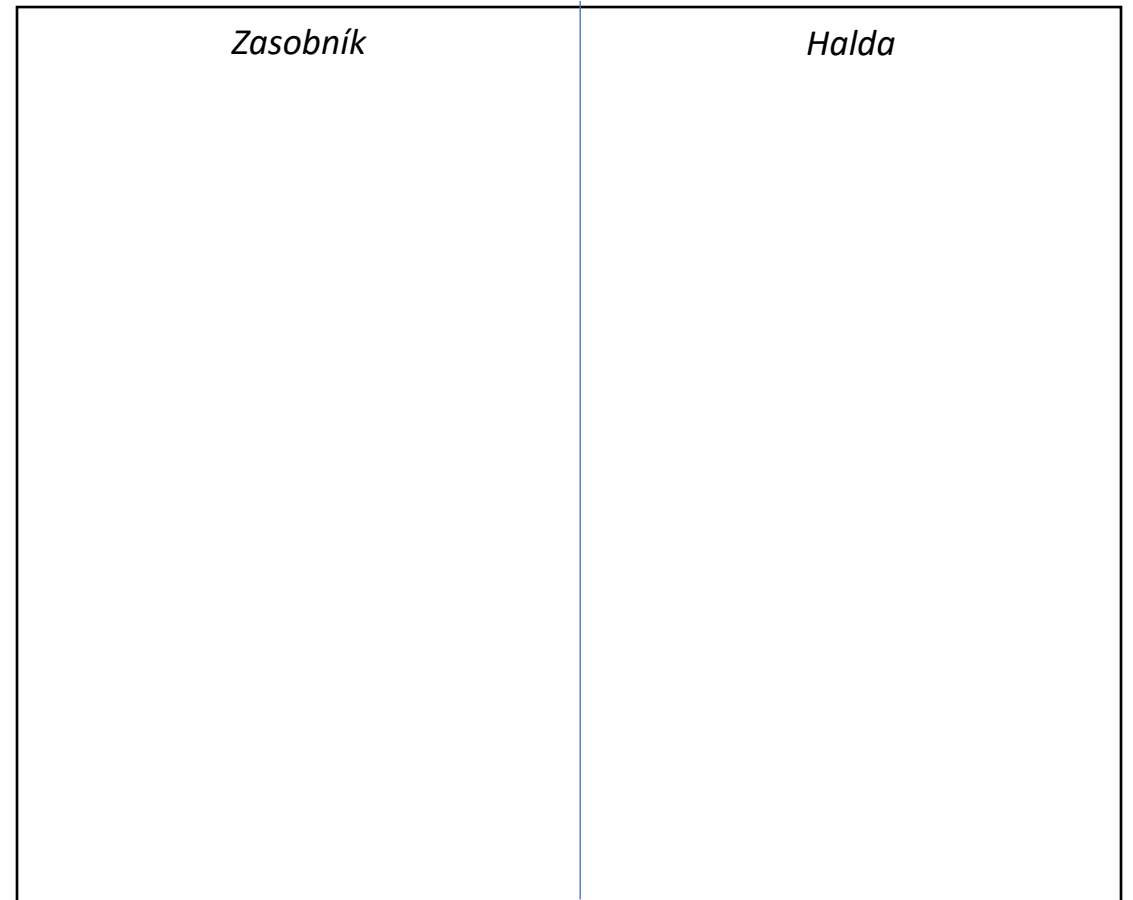
Faktoriál



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i;  
}  
  
Console.WriteLine(f);
```

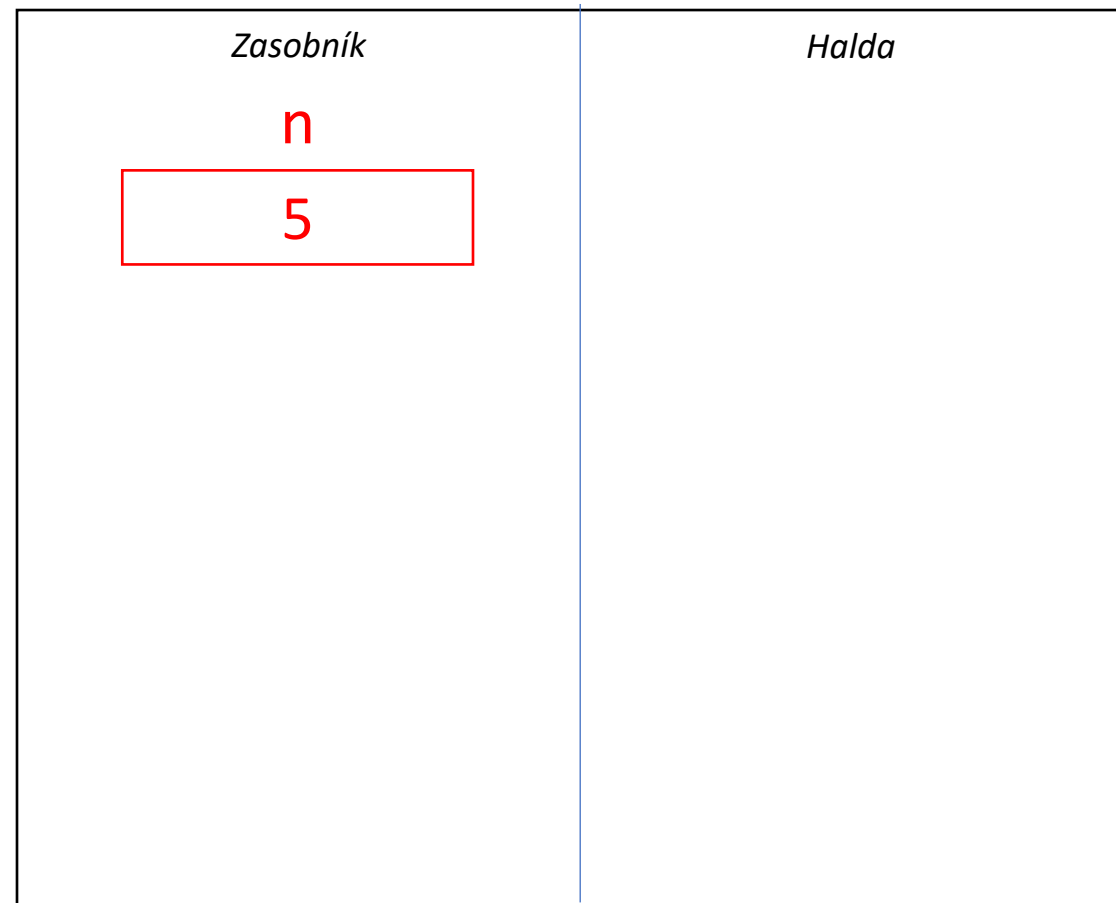
Paměť RAM



Faktoriál

```
int n = 5;
```

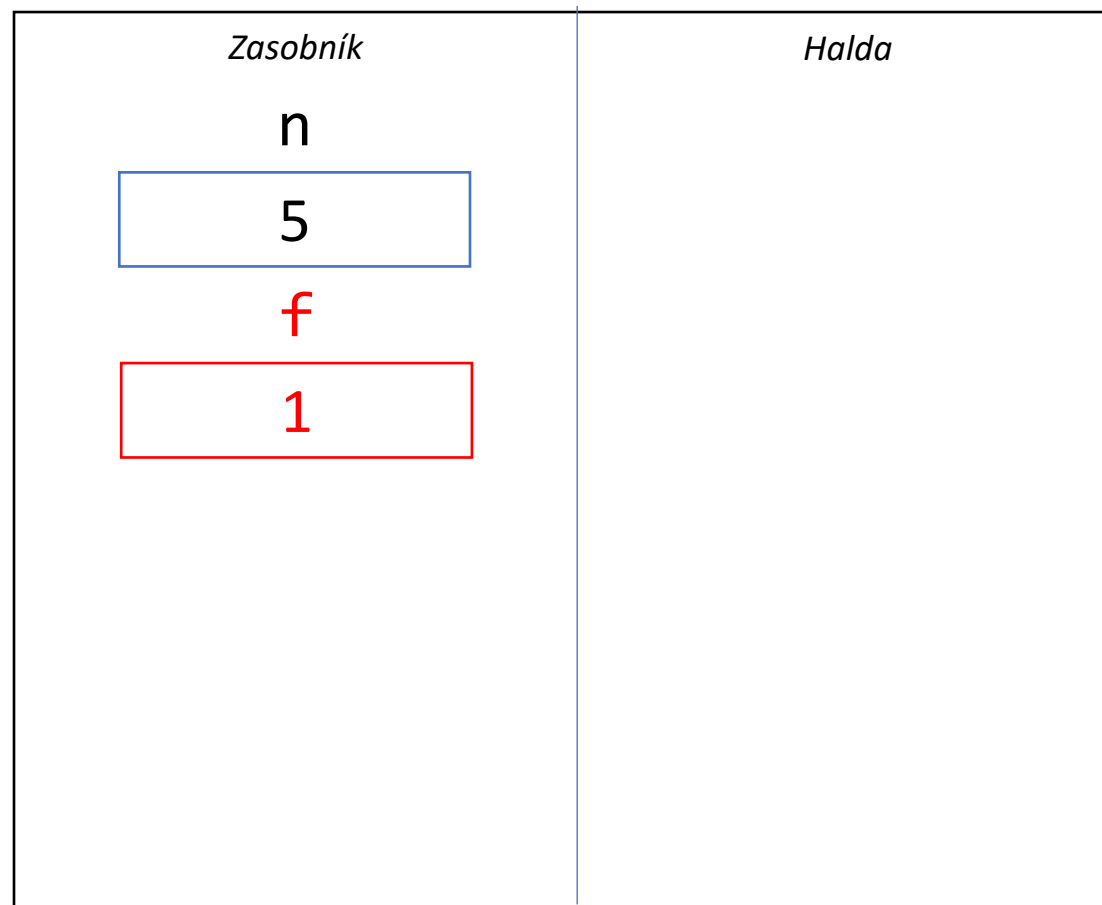
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;
```

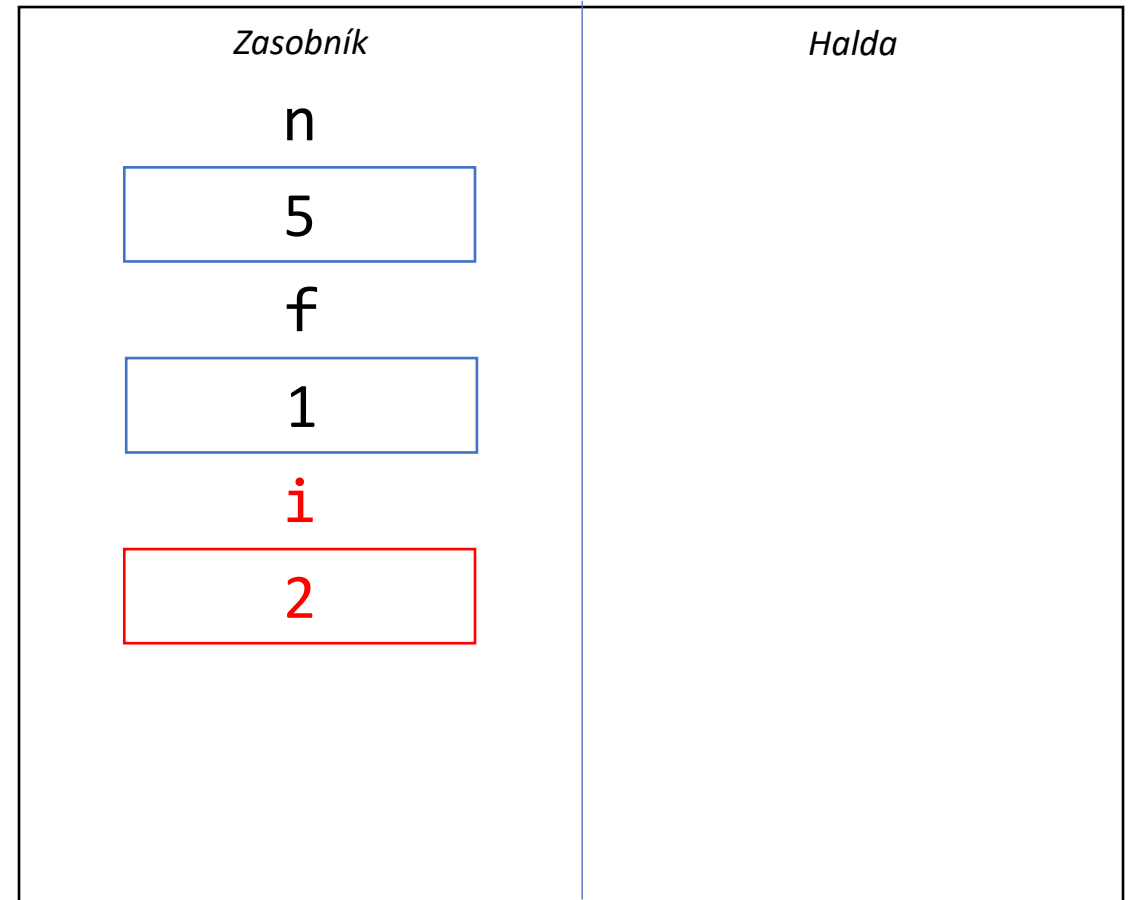
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
  
}
```

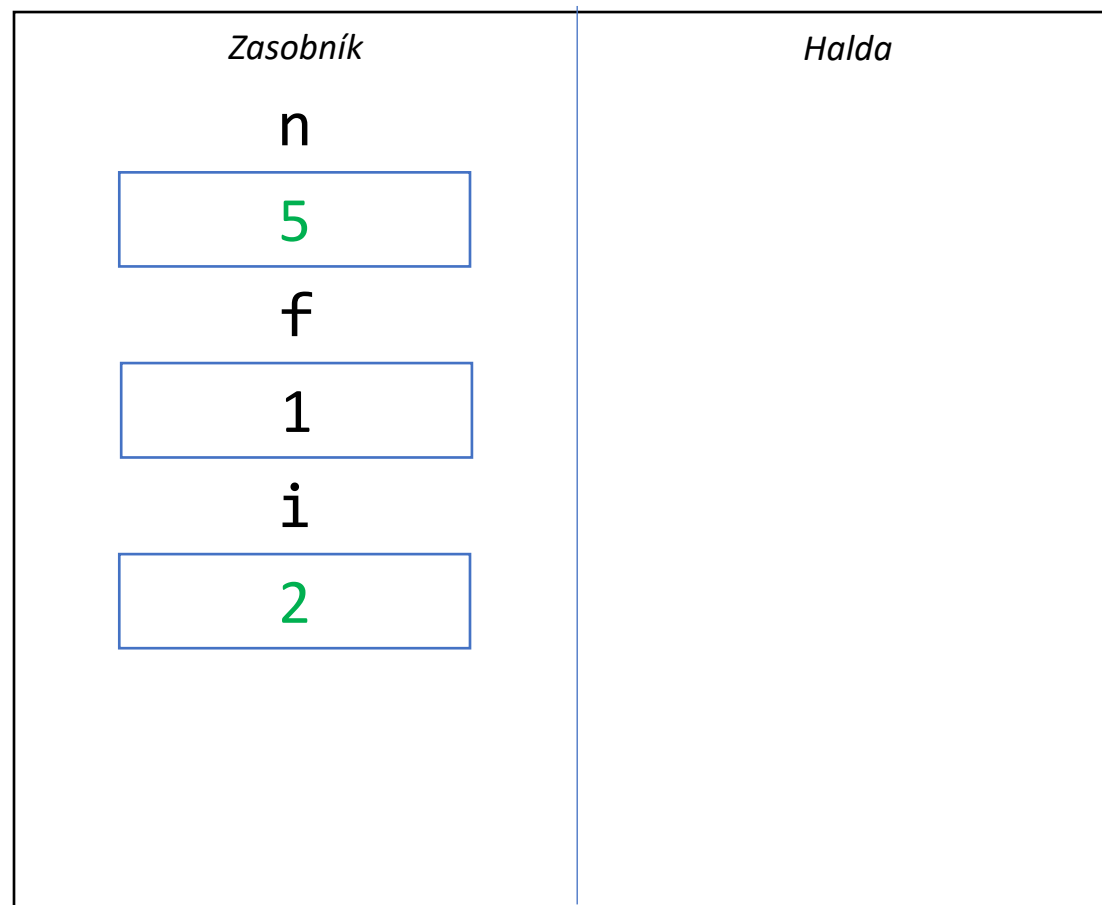
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
  
}
```

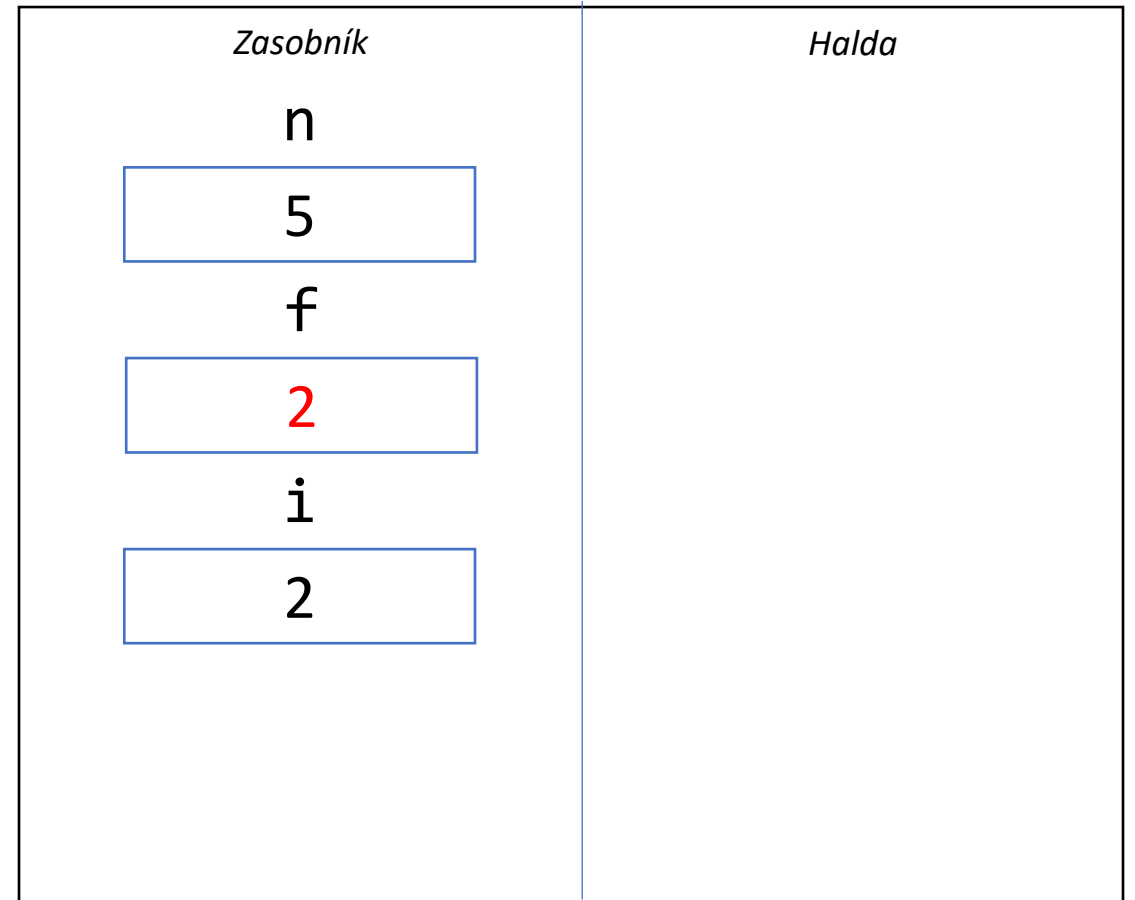
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2  
}
```

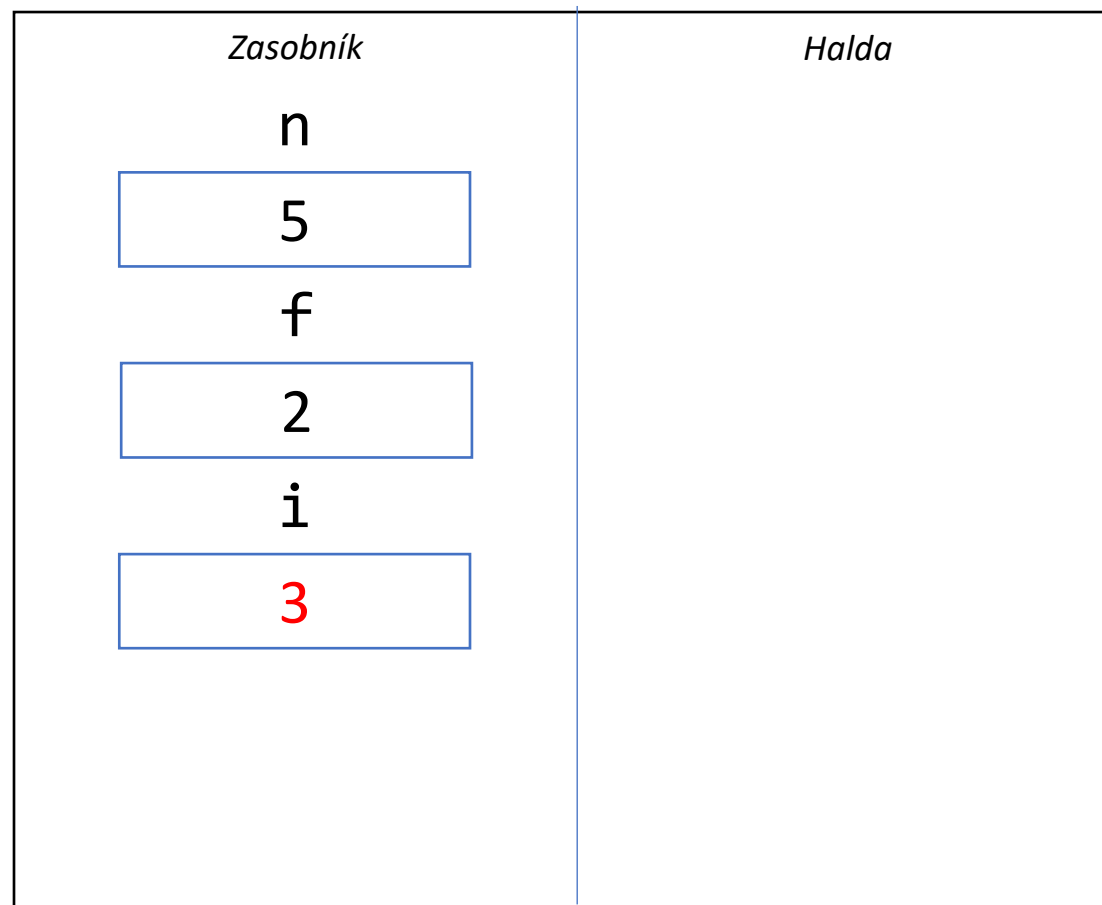
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2  
}
```

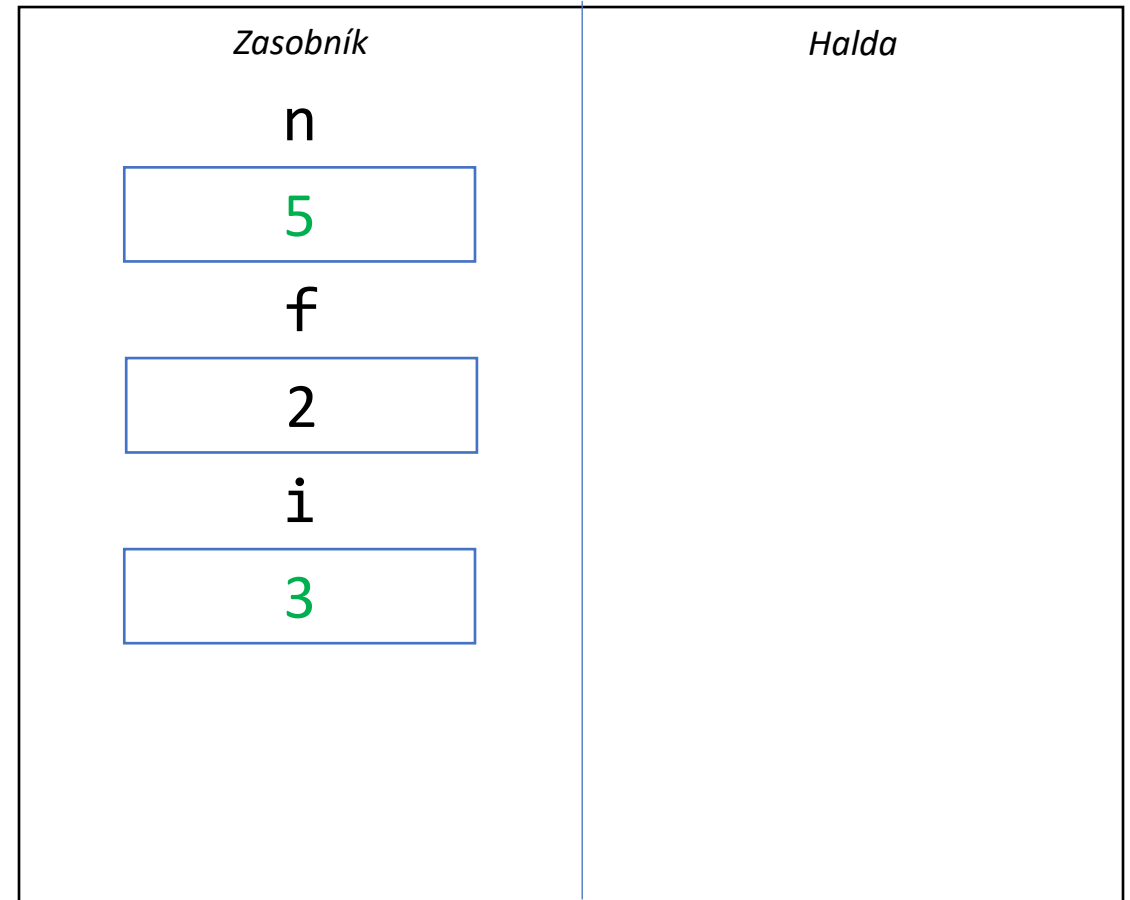
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2  
}
```

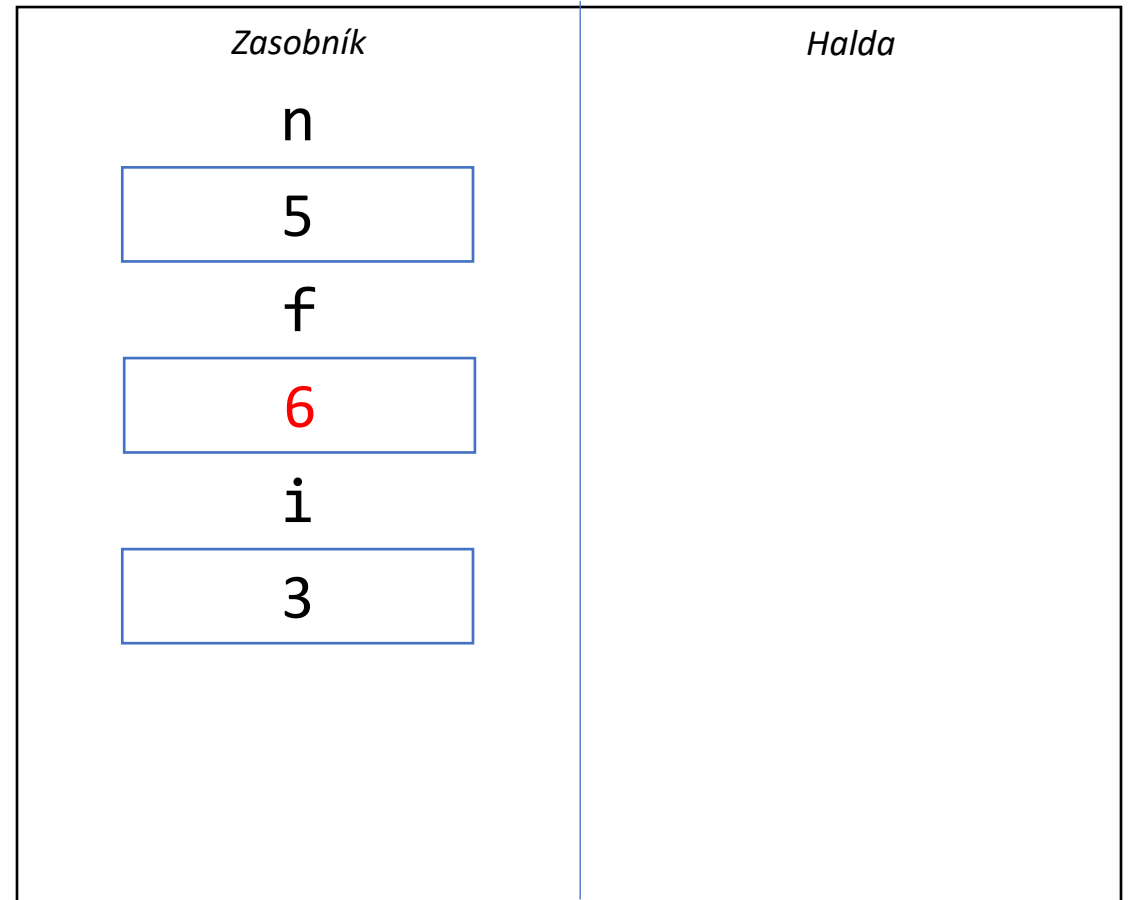
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2 * 3  
}
```

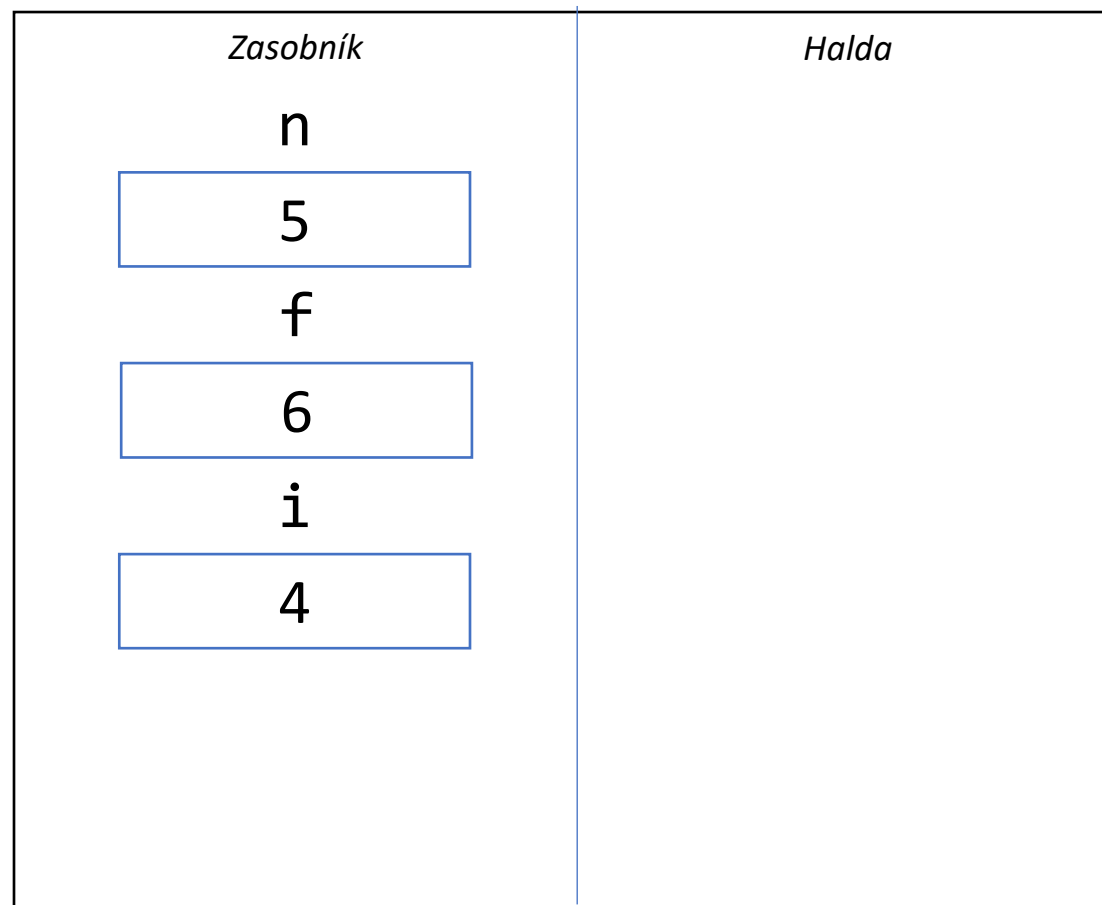
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2 * 3  
}
```

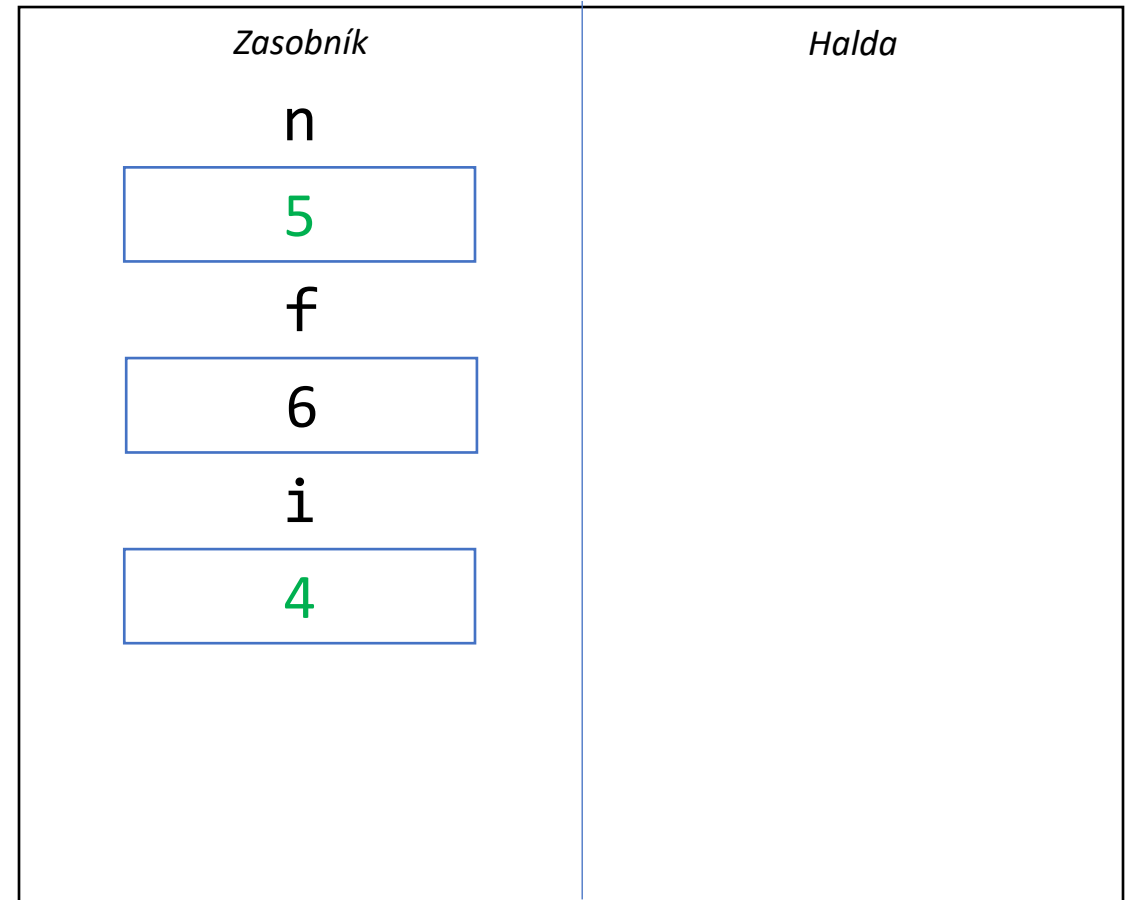
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i 4 <= 5 n; i++)  
{  
    f *= i; // 1 * 2 * 3  
}
```

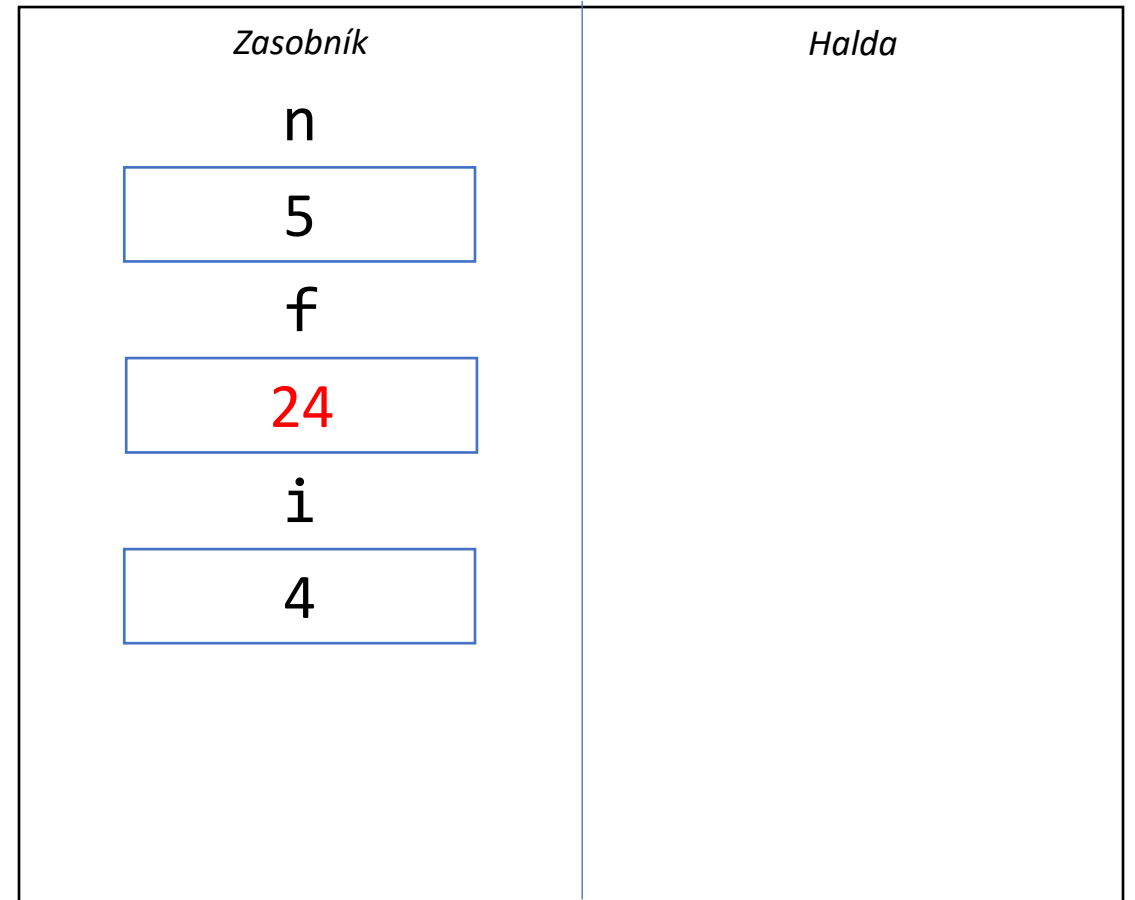
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2 * 3 * 4  
}
```

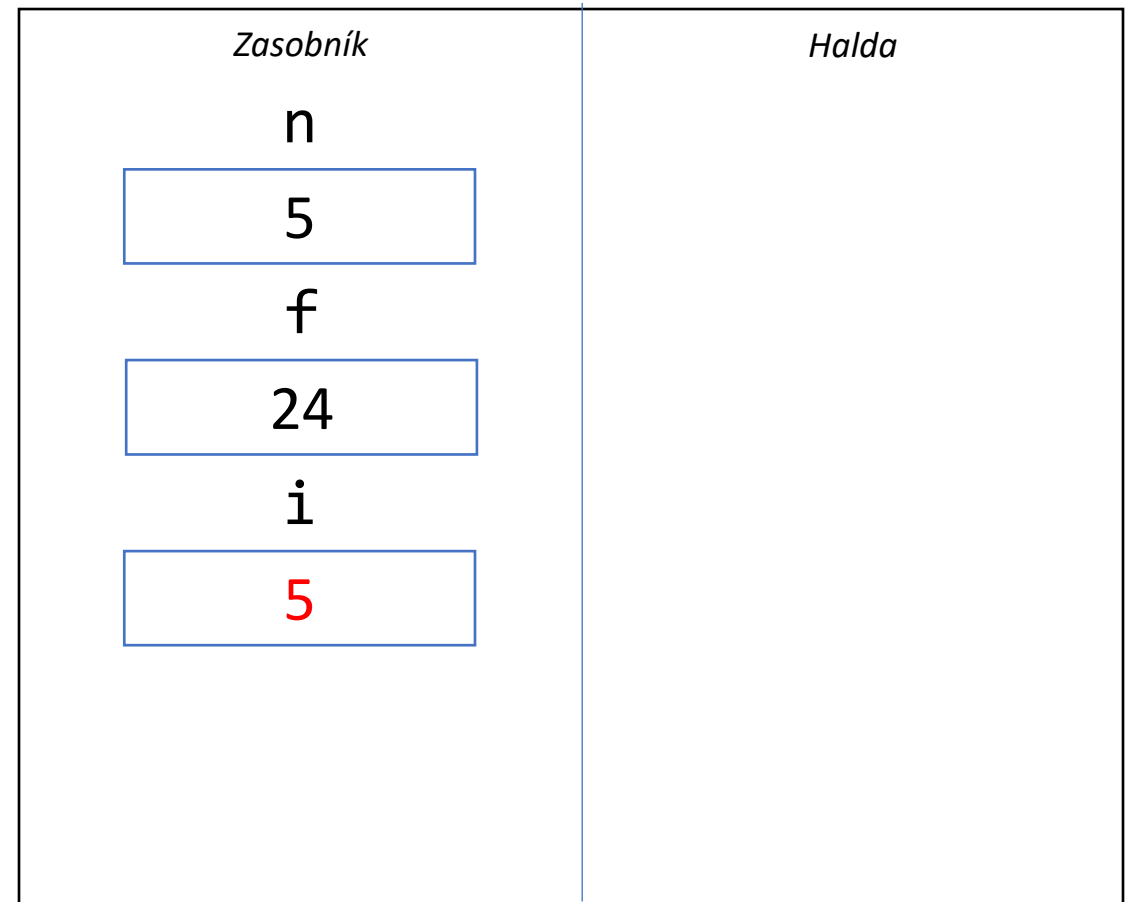
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2 * 3 * 4  
}
```

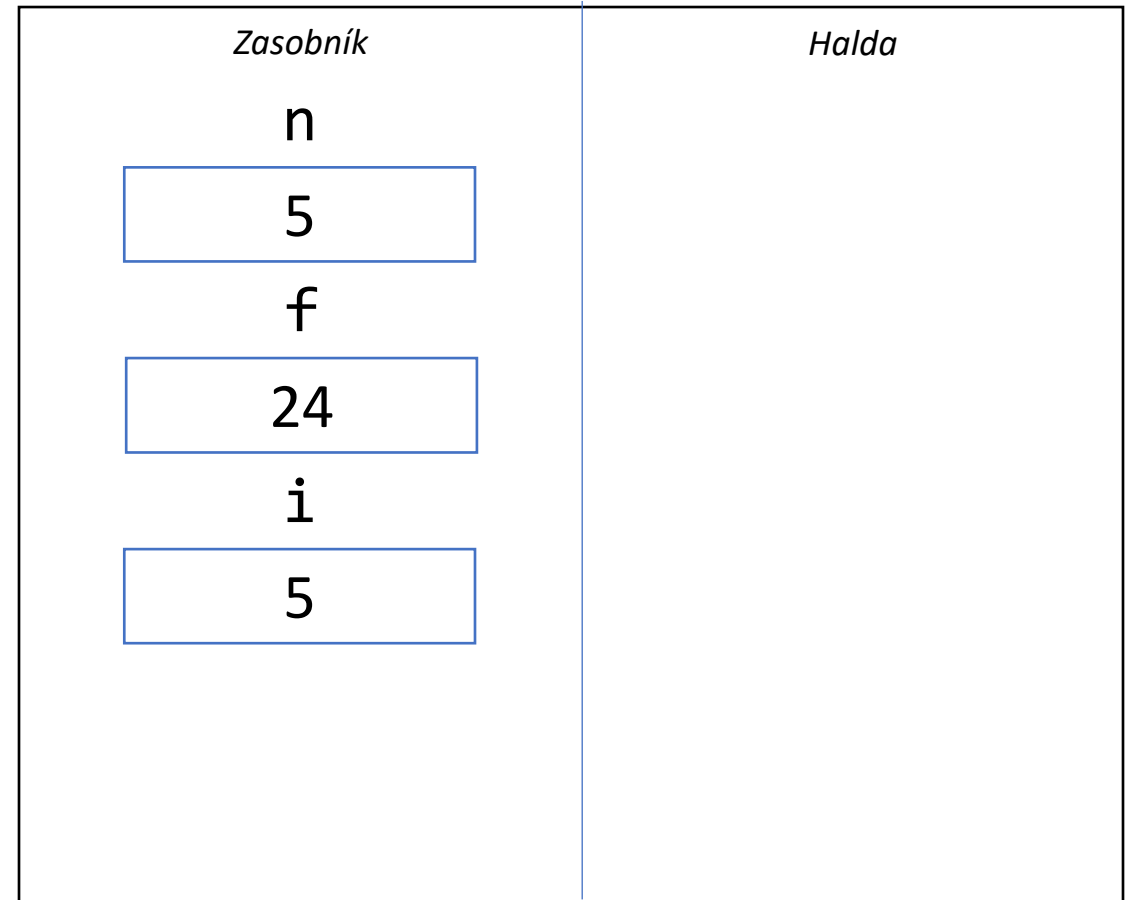
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2 * 3 * 4  
}
```

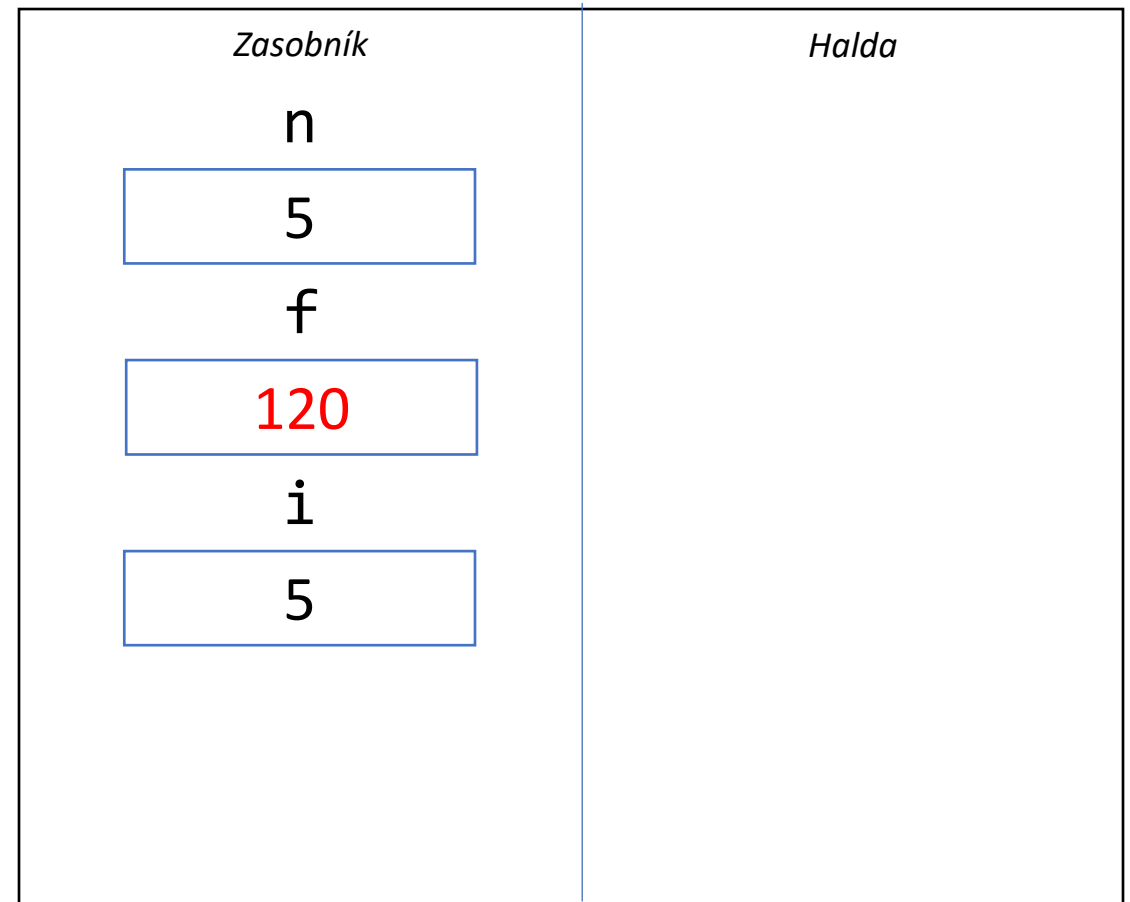
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2 * 3 * 4 * 5  
}
```

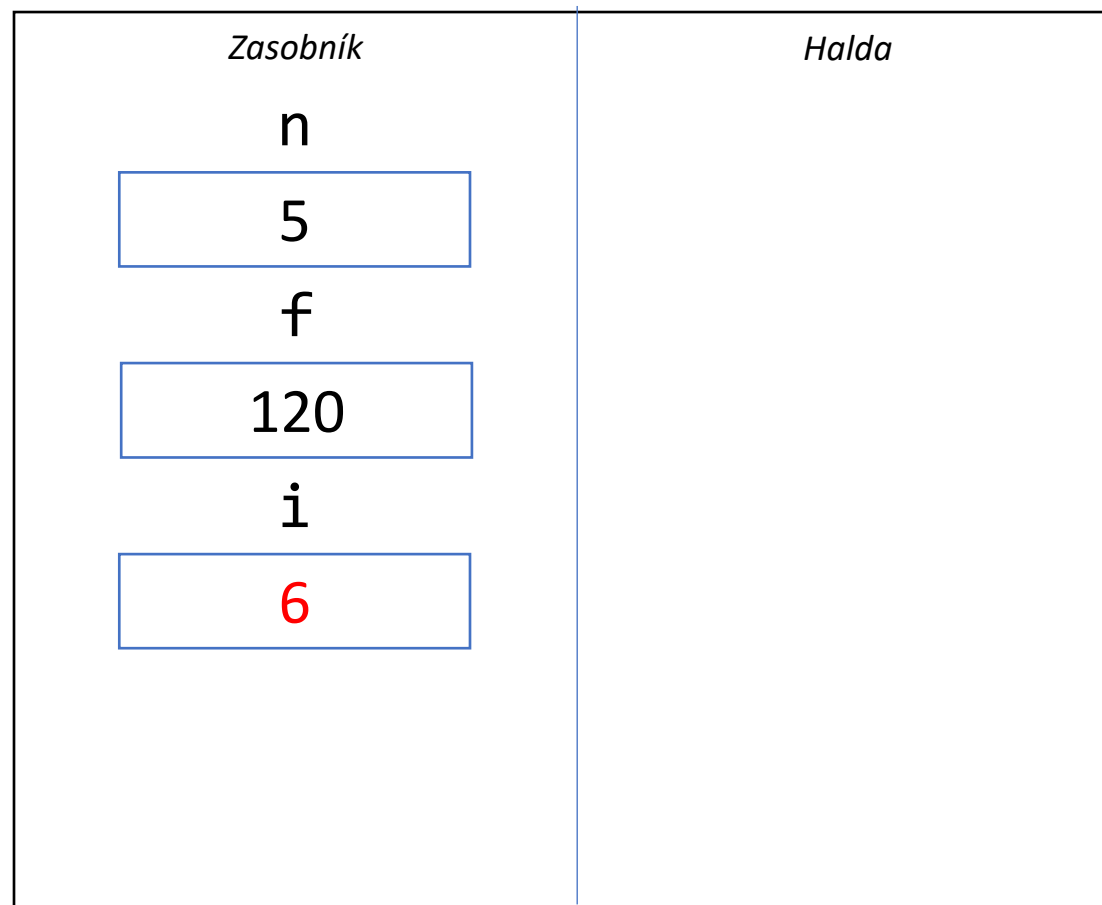
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2 * 3 * 4 * 5  
}
```

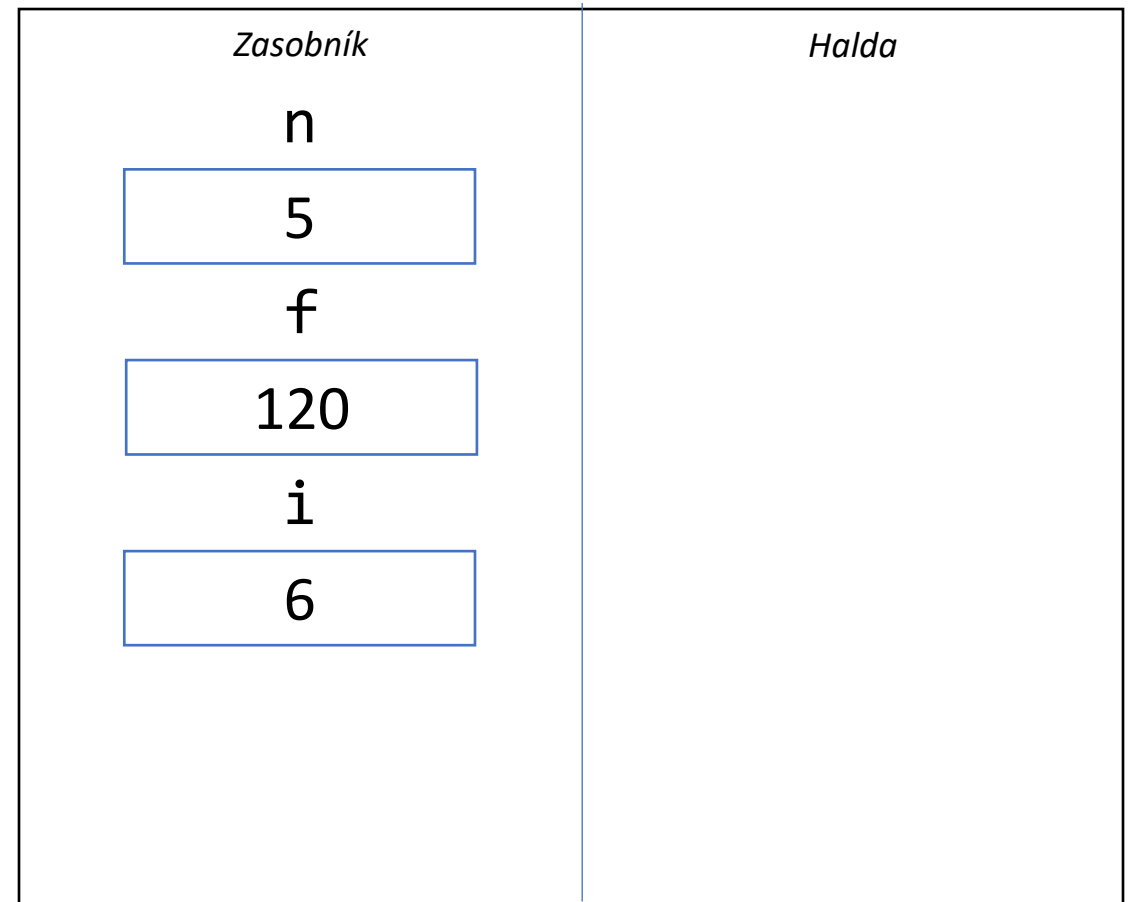
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2 * 3 * 4 * 5  
}
```

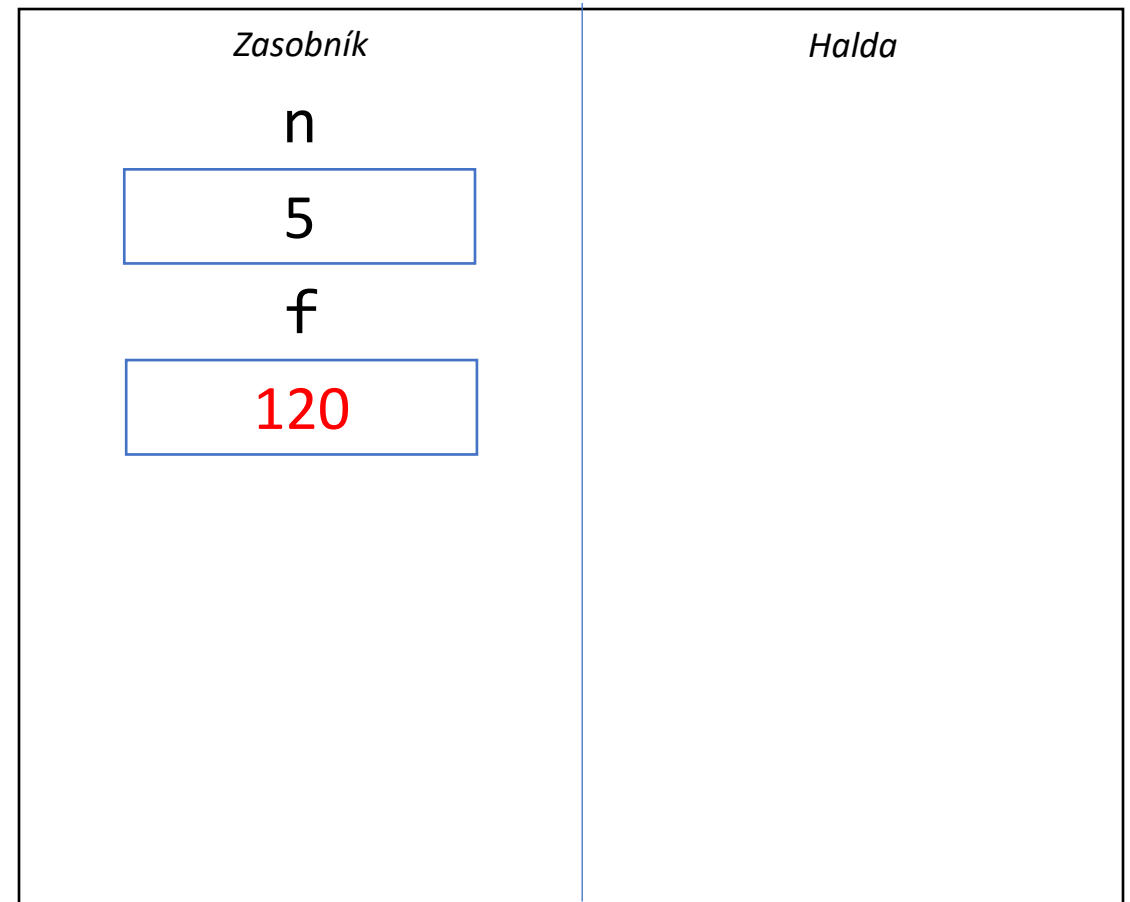
Paměť RAM



Faktoriál

```
int n = 5;  
int f = 1;  
  
for (int i = 2; i <= n; i++)  
{  
    f *= i; // 1 * 2 * 3 * 4 * 5  
}  
  
Console.WriteLine(f);
```

Paměť RAM



Použité zdroje

[1] for statement - C# reference | Microsoft Docs. [online]. Copyright © Microsoft 2021 [cit. 08.02.2021]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/for>



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání



Programování a algoritmizace

Děkuji za pozornost

Strategický projekt UTB ve Zlíně, reg. č. CZ.02.2.69/0.0/0.0/16_015/0002204



Ing. et Ing. Erik Král, Ph.D.
FAI, ÚPKS