

# OS 2024 | Cvičenie V

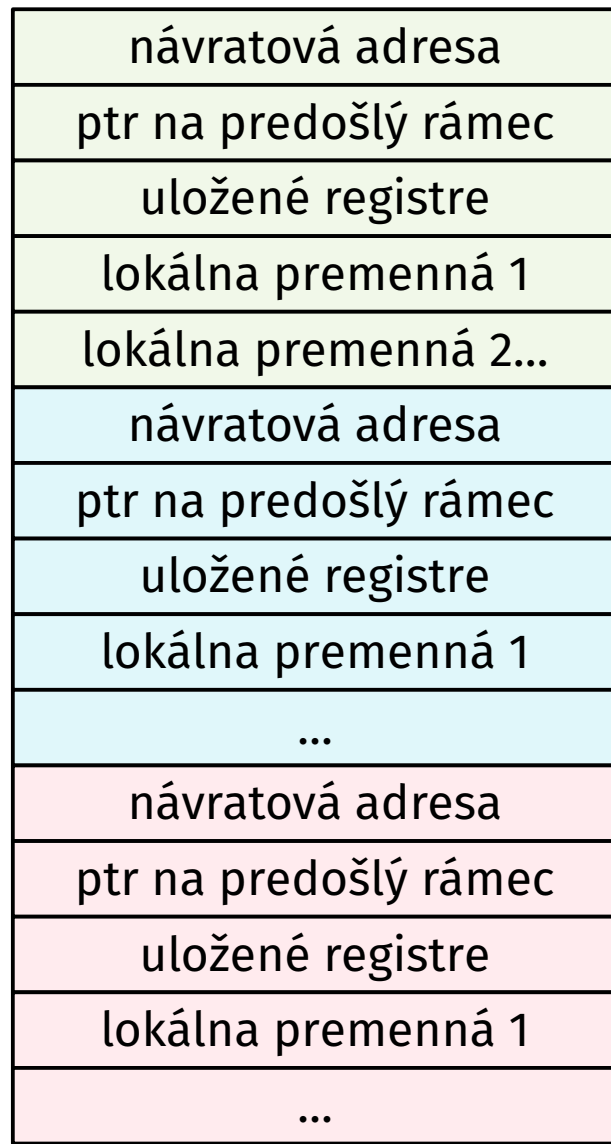
Program:

1) Zásobník a volacie konvencie

zásobníkový  
rámec

fp →

sp →



vysoké adresy

nízke adresy

```
int main() {  
    int a;  
    ...  
    f1(a)  
    ...  
}
```

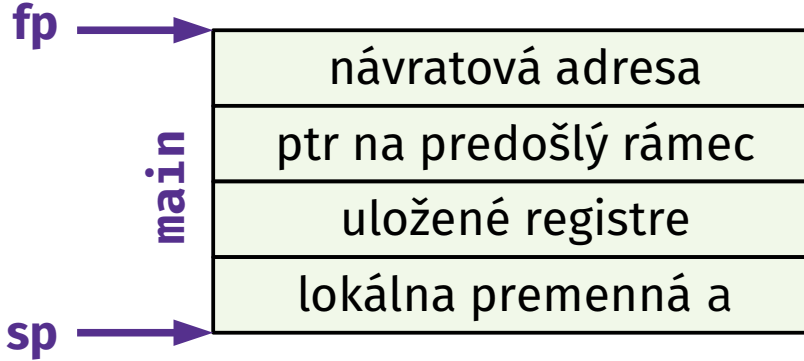
```
int f1(int x) {  
    int p, q;  
    ...  
    f2()  
    ...  
}
```

```
int f2() {  
    ...  
}
```

```
int main() {  
    int a;  
    ...  
    f1(a)  
    ...  
}
```

```
int f1(int x) {  
    int p, q;  
    ...  
    f2()  
    ...  
}
```

```
int f2() {  
    ...  
}
```



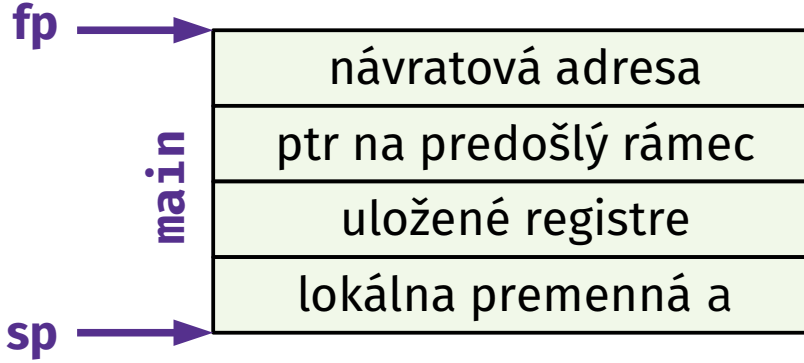
vysoké adresy

nízke adresy

```
int main() {  
    int a;  
    ...  
    pc → f1(a)  
    ...  
}
```

```
int f1(int x) {  
    int p, q;  
    ...  
    f2()  
    ...  
}
```

```
int f2() {  
    ...  
}
```



vysoké adresy

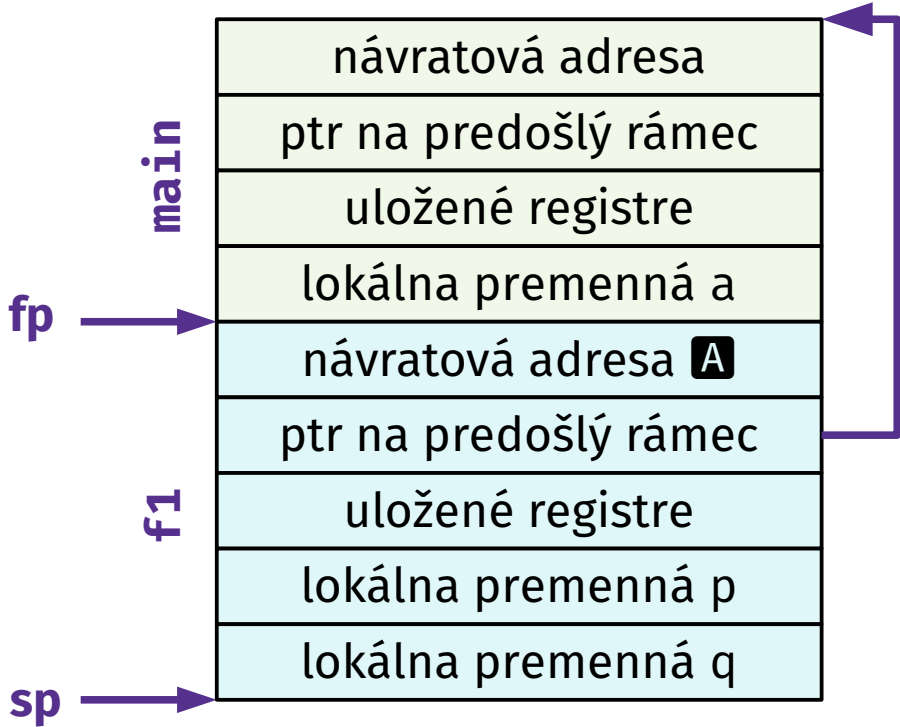
nízke adresy

```
int main() {
    int a;
    ...
    f1(a)
    ...
}

int f1(int x) {
    int p, q;
    ...
    f2()
    ...
}

int f2() {
    ...
}
```

**A** →



vysoké adresy

nízke adresy

```

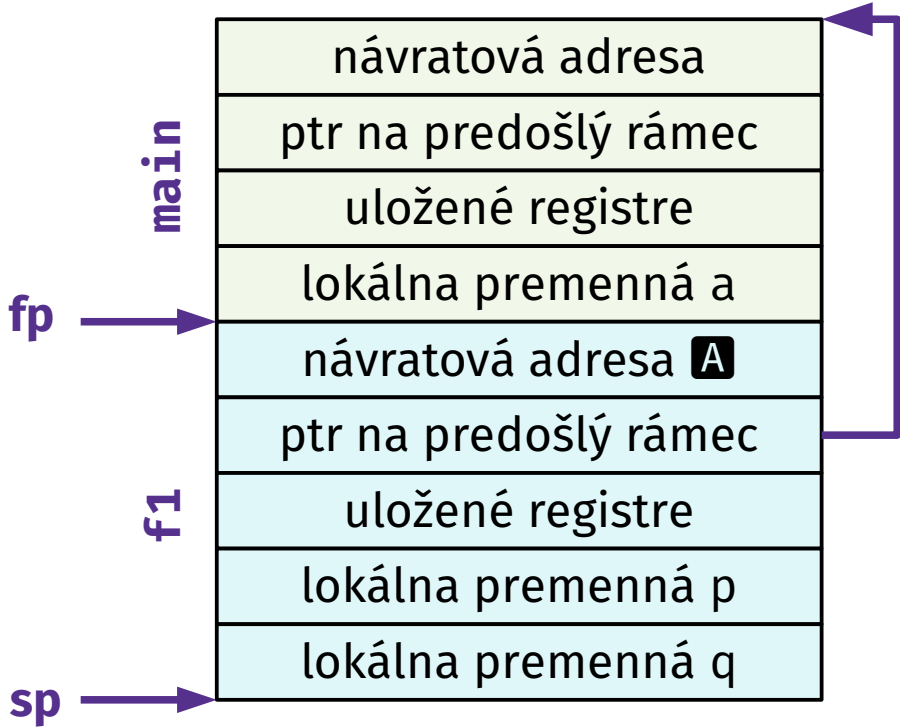
int main() {
    int a;
    ...
    f1(a)
    ...
}

int f1(int x) {
    int p, q;
    ...
    f2()
    ...
}

int f2() {
    ...
}

```

**A** →



vysoké adresy

nízke adresy

```
int main() {
    int a;
    ...
    f1(a)
    ...
}
```

```
int f1(int x) {
    int p, q;
    ...
    f2()
    ...
}
```

```
int f2() {
    ...
}
```

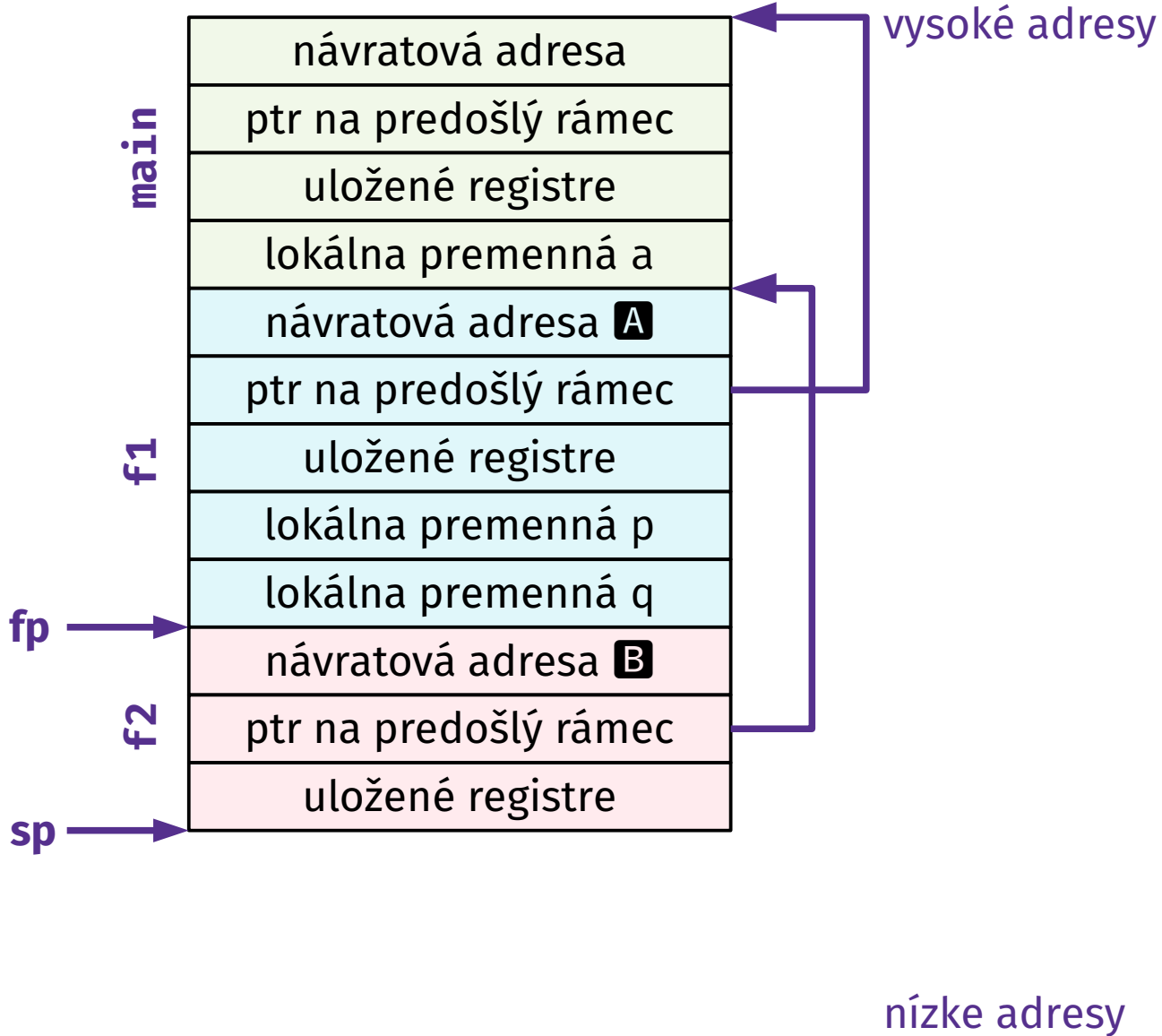
**A**



**B**



**pc**





# Čo si zapamätať o zásobníku

- zavolaním funkcie sa vytvorí nový zásobníkový rámec
- zásobník rastie z vyšších adries k nižším
- register sp ukazuje vždy na vrchol zásobníka
- register fp ukazuje na začiatok aktívneho rámca
- pri volaní funkcie sa na zásobník sa najprv uloží návratová adresa a potom smerník na predchádzajúci rámec
- na zásobník sa okrem tohto ukladajú aj argumenty, ktoré sa nezmestili do registrov určených pre argumenty; registre, ktoré musí uložiť *volaná* funkcia a lokálne premenné funkcie

# Prológ a epilóg funkcie

- Ukladanie registrov a premenných na zásobník musí zabezpečiť kompilátor podľa zvolenej **volacej konvencie**.
- ukážka:

## prológ:

```
addi sp, sp, -24
sd   ra, 16(sp)
sd   s0, 8(sp)
sd   s1, 0(sp)
sd   s1, 0(sp)
```

## epilóg:

```
ld   ra, 16(sp)
ld   s0, 8(sp)
ld   s1, 0(sp)
addi sp, sp, 24
```

- Na zásobník sa ukladajú registre uložené volanou funkciou (*callee-saved*), ktoré musia byť zachované naprieč volaniami funkcií.