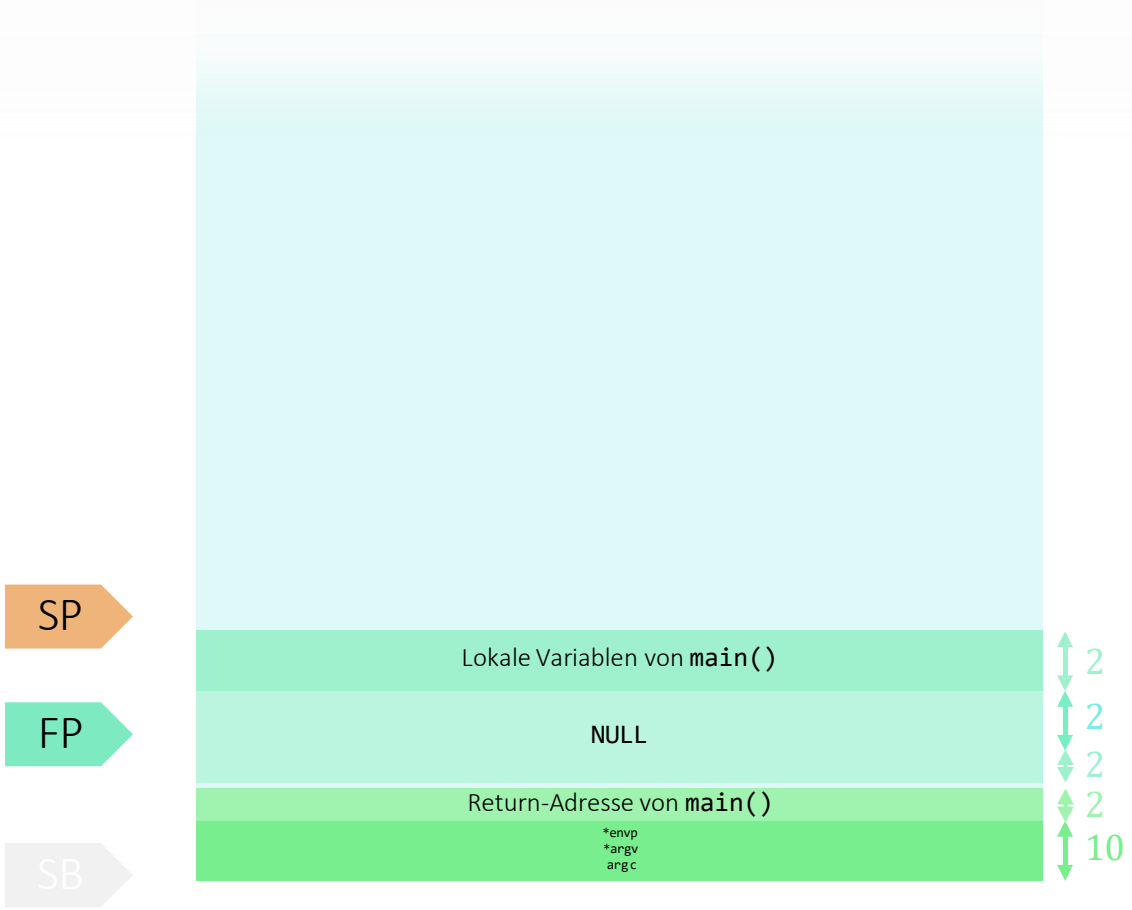
Stack am Beispiel erklärt

Stack am Beispiel erklärt

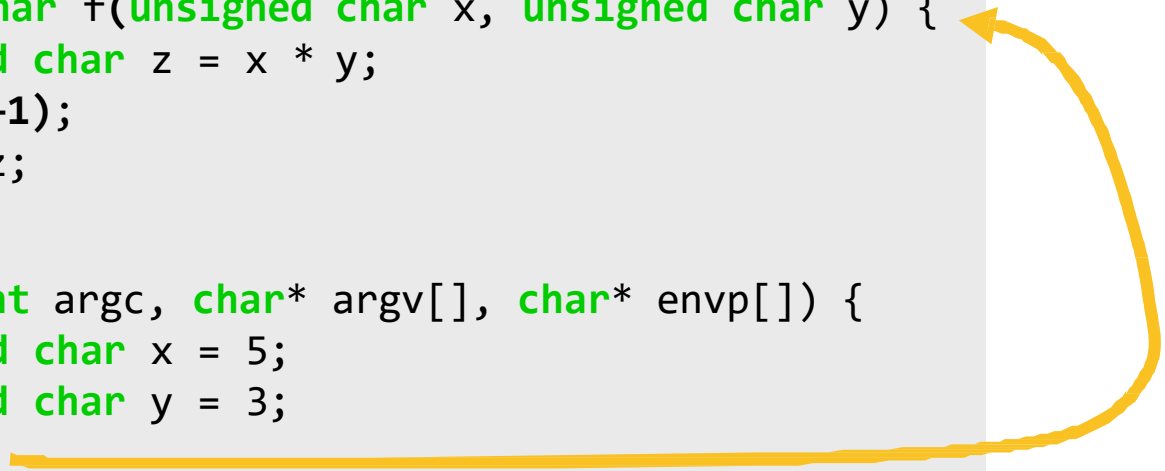
```
unsigned char g(unsigned char z) {
    unsigned char t = z / 2;
    return t;
}

unsigned char f(unsigned char x, unsigned char y) {
    unsigned char z = x * y;
    z = g(z+1);
    return z;
}

int main(int argc, char* argv[], char* envp[]) {
    unsigned char x = 5;
    unsigned char y = 3;
    f(x, y)
    return 0;
}
```

Stack nach der Initialisierung

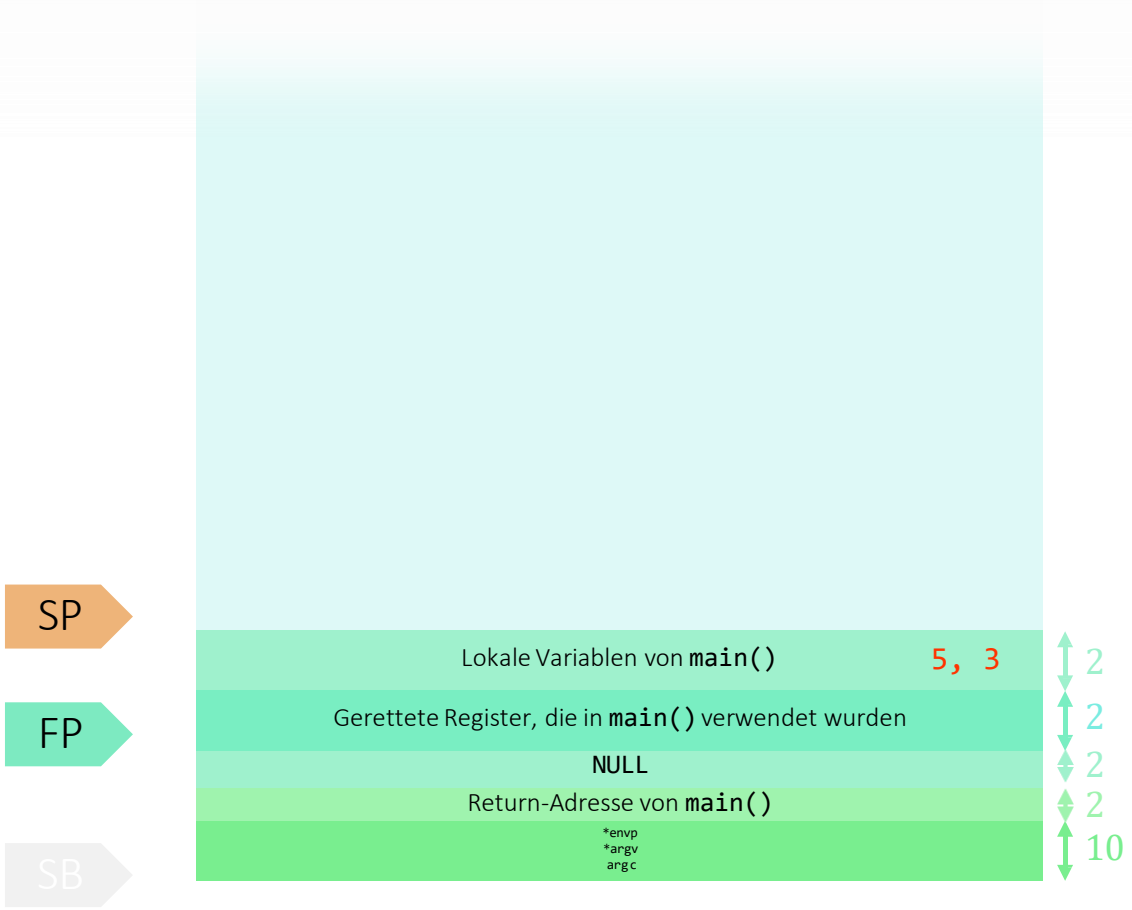


Aufruf einer Funktion

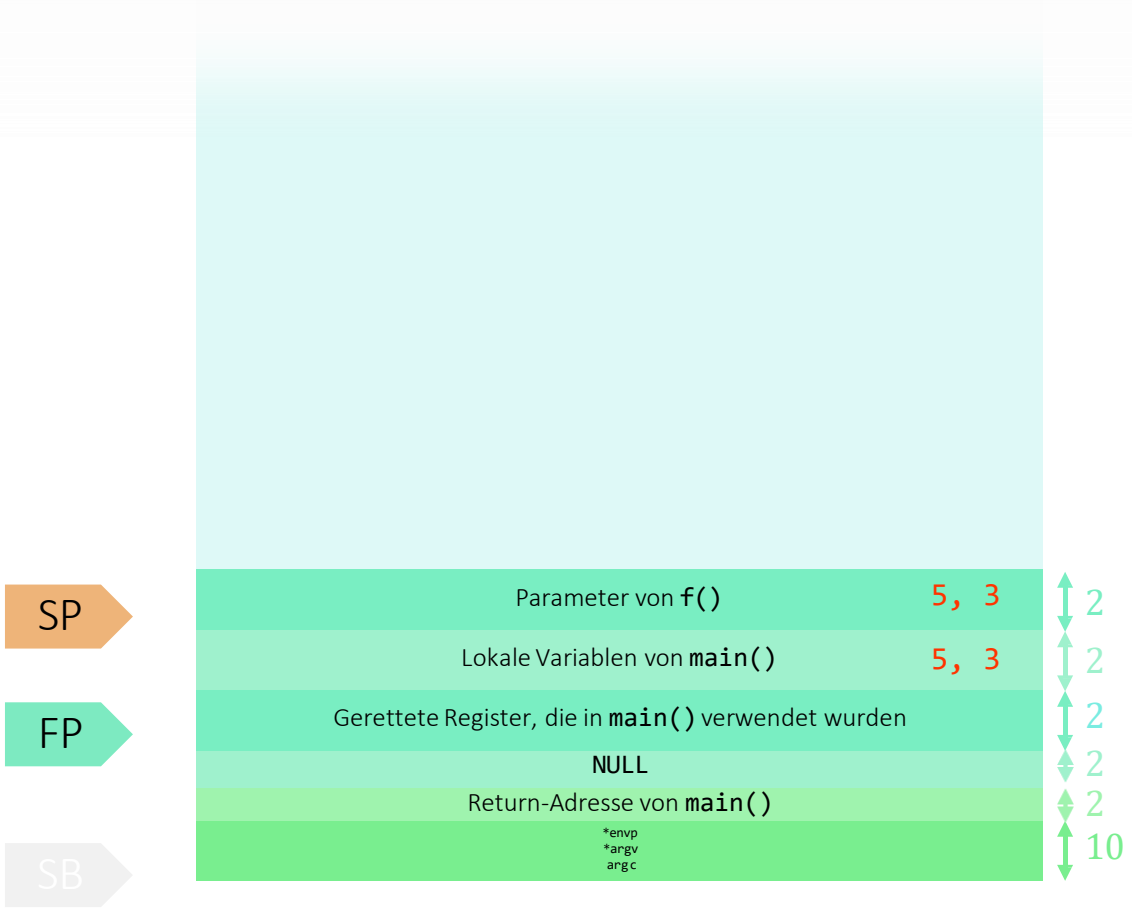
```
unsigned char g(unsigned char z) {
    unsigned char t = z / 2;
    return t;
}

unsigned char f(unsigned char x, unsigned char y) {
    unsigned char z = x * y;
    z = g(z+1);
    return z;
}

int main(int argc, char* argv[], char* envp[]) {
    unsigned char x = 5;
    unsigned char y = 3;
    f(x, y)
    return 0;
}
```



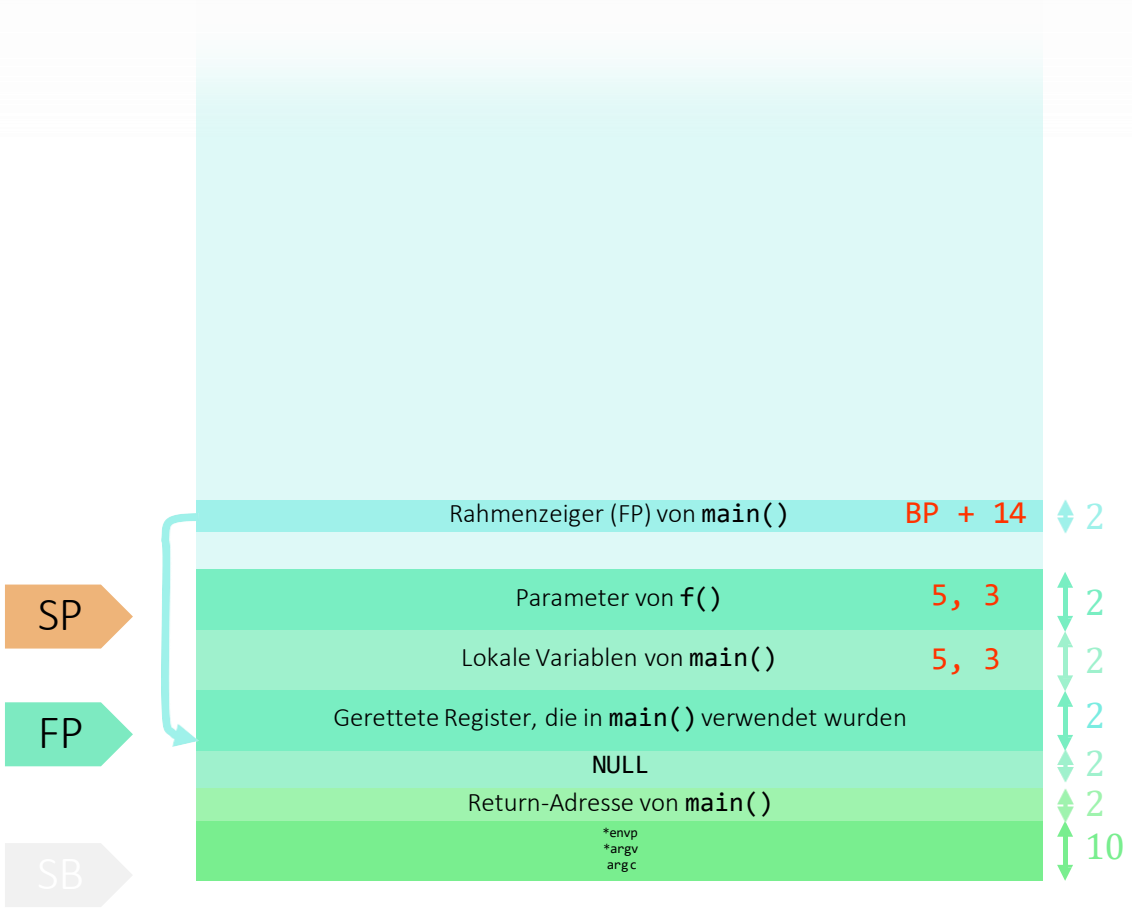
Register retten



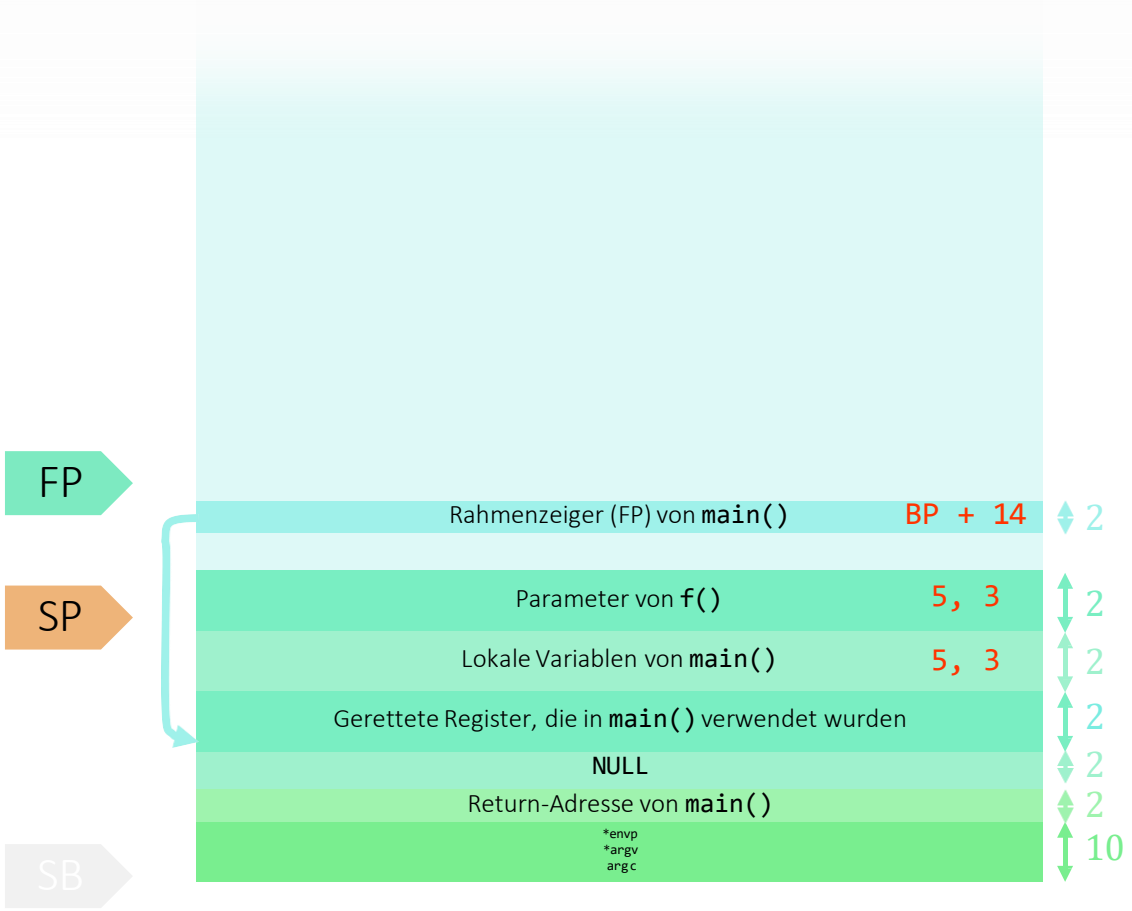
Parameter auf den Stapel legen



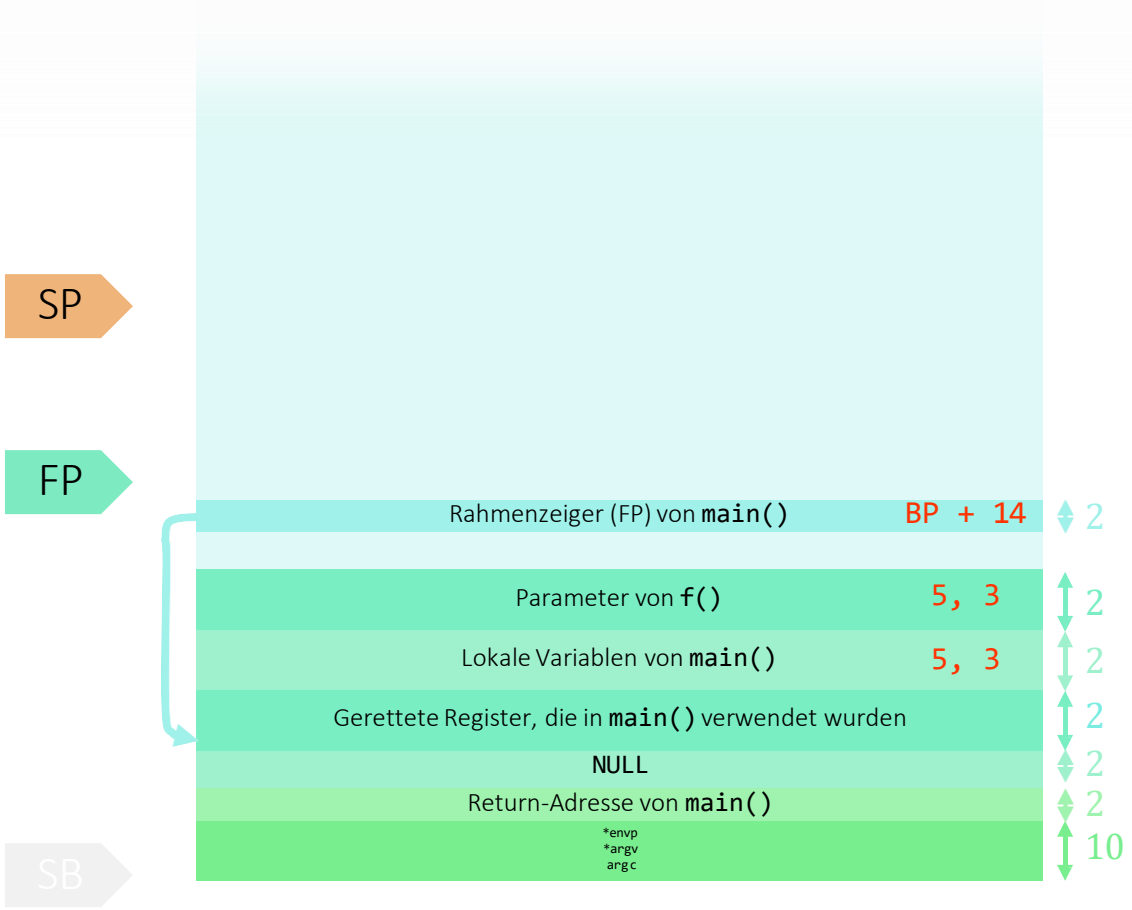
Rahmenzeiger retten



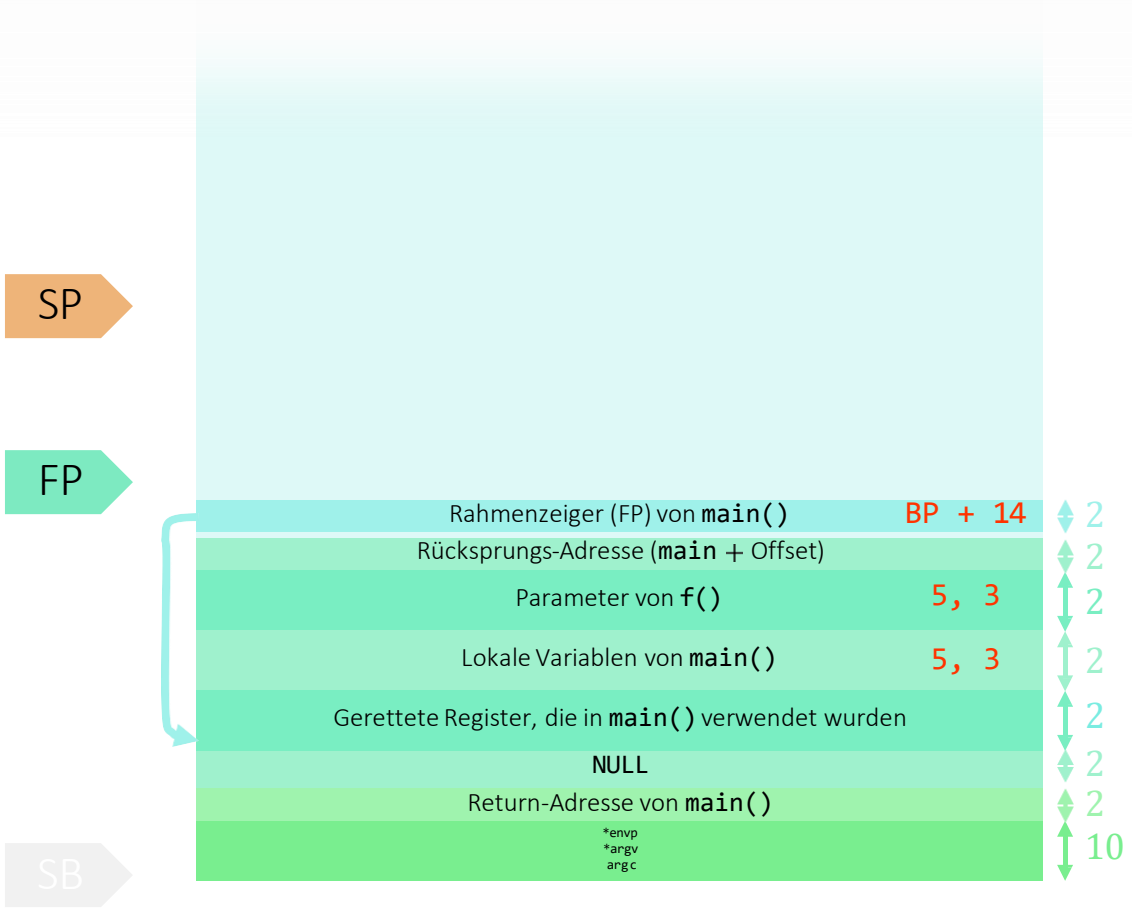
Rahmenzeiger verschieben



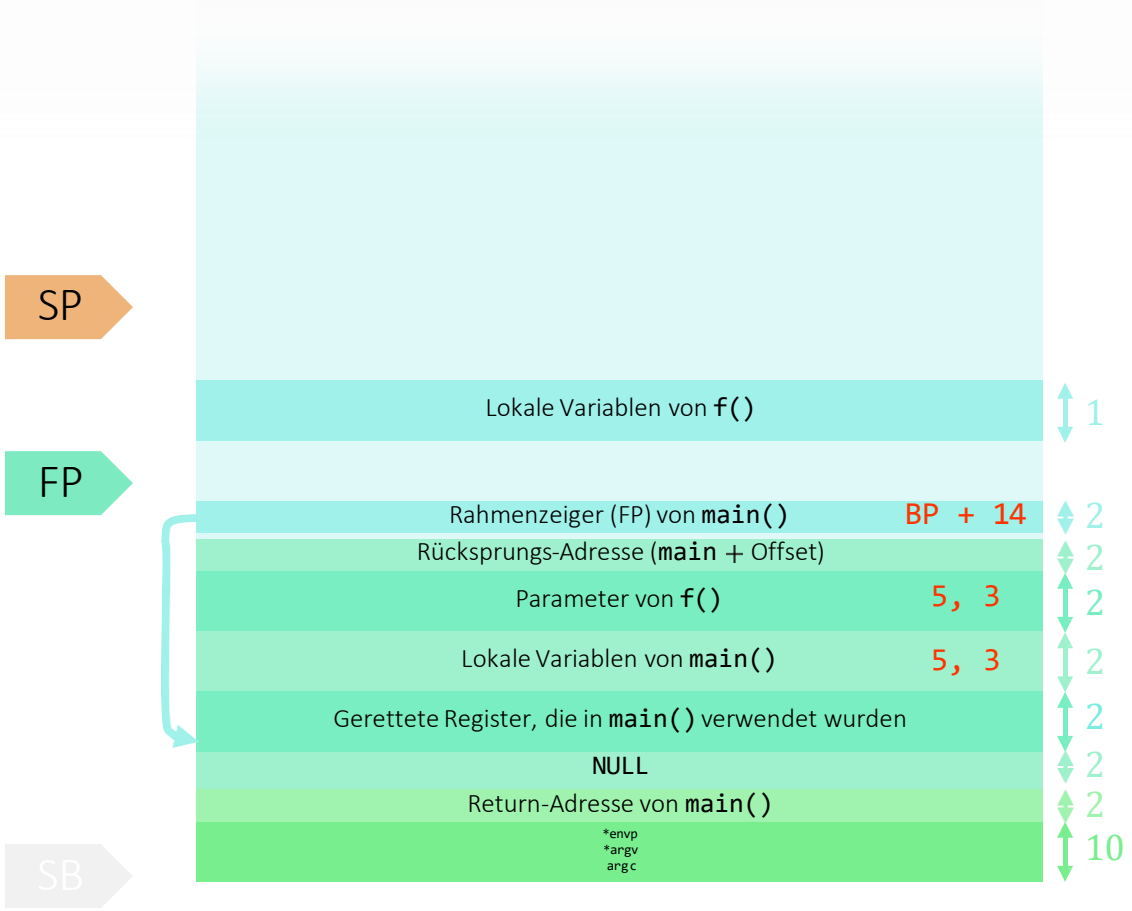
Stapelzeiger verschieben



Rücksprungsadresse speichern

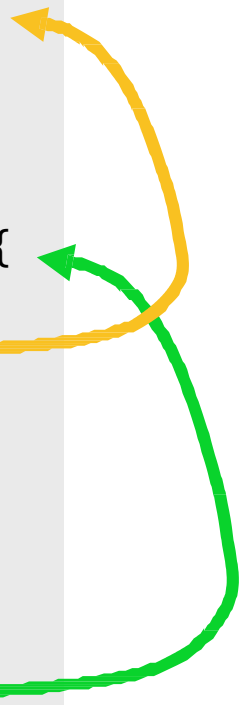


Unterprogrammaufruf

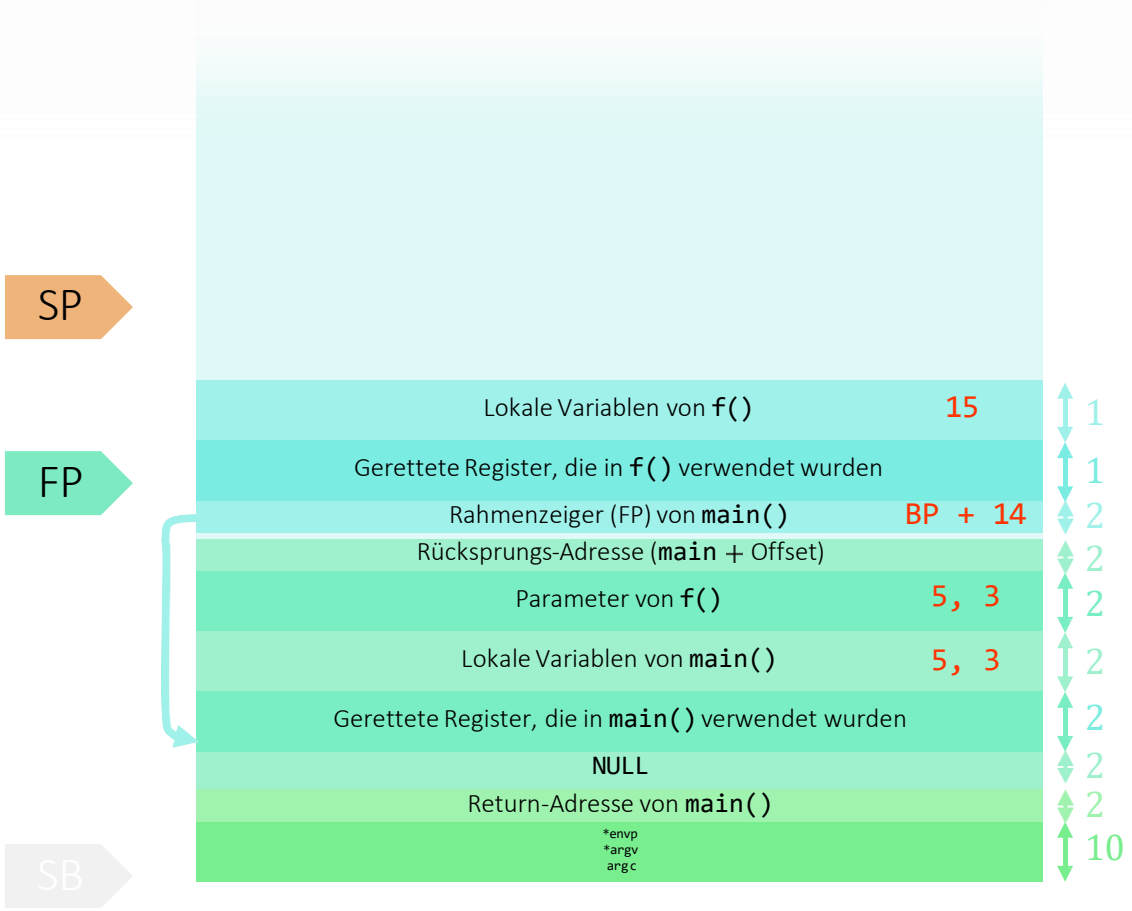


Aufruf einer weiteren Funktion

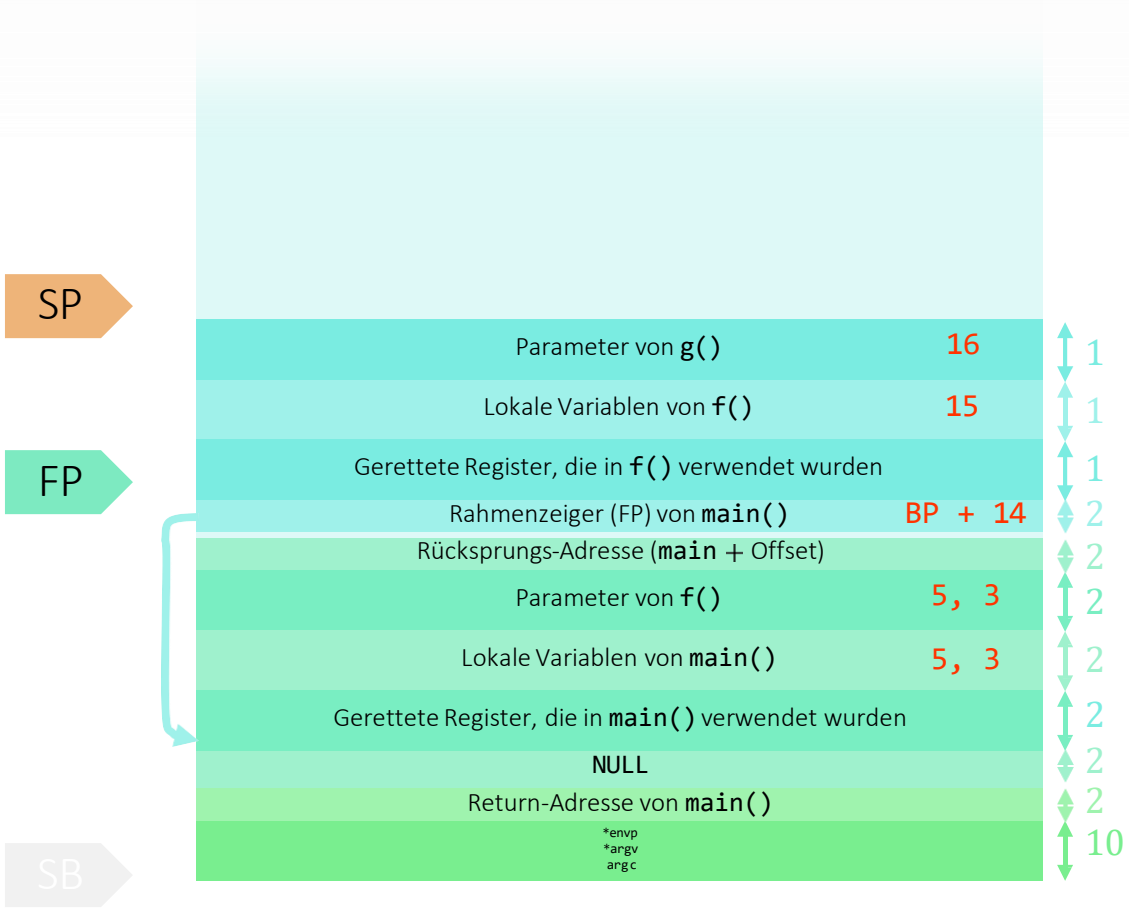
```
unsigned char g(unsigned char z) {
    unsigned char t = z / 2;
    return t;
}

unsigned char f(unsigned char x, unsigned char y) {
    unsigned char z = x * y;
    z = g(z+1);
    return z;
}

int main(int argc, char* argv[], char* envp[]) {
    unsigned char x = 5;
    unsigned char y = 3;
    f(x, y)
    return 0;
}
```



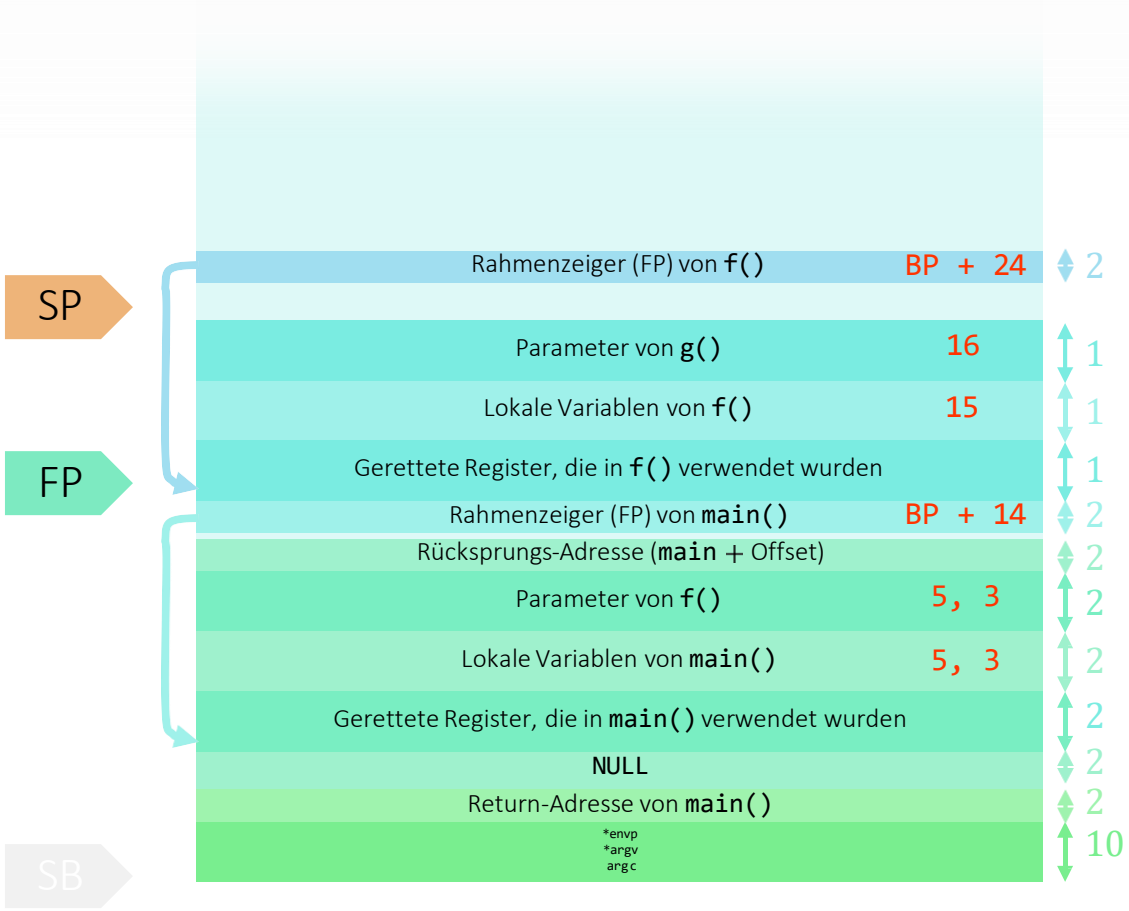
Register retten



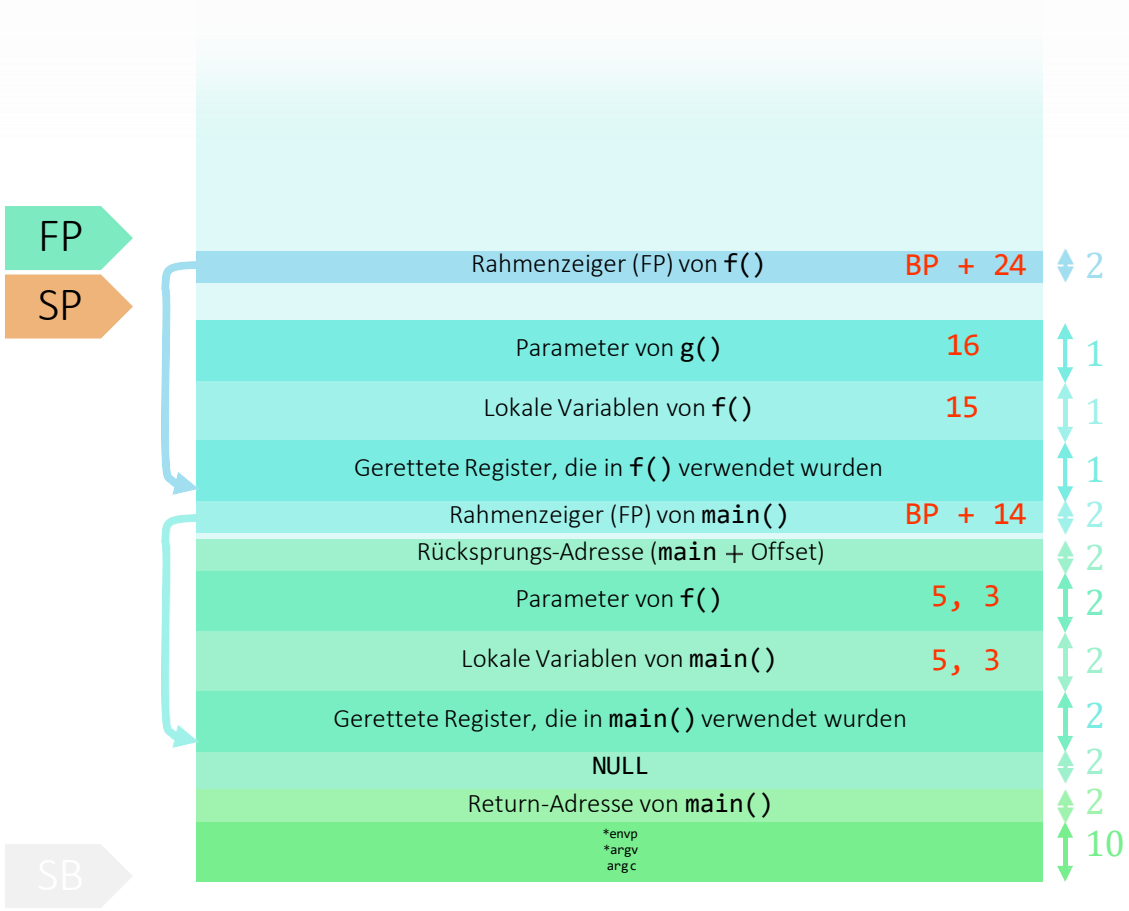
Parameter auf den Stapel legen



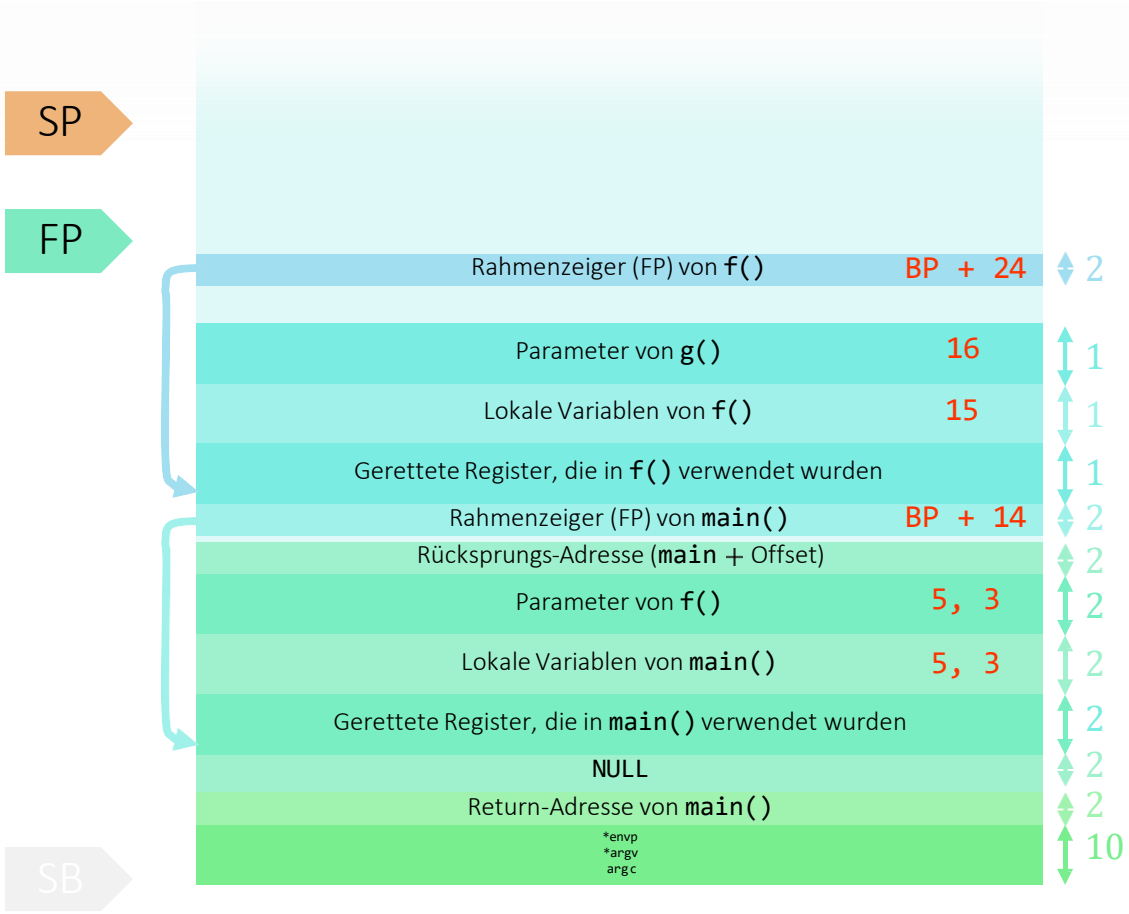
Rahmenzeiger retten



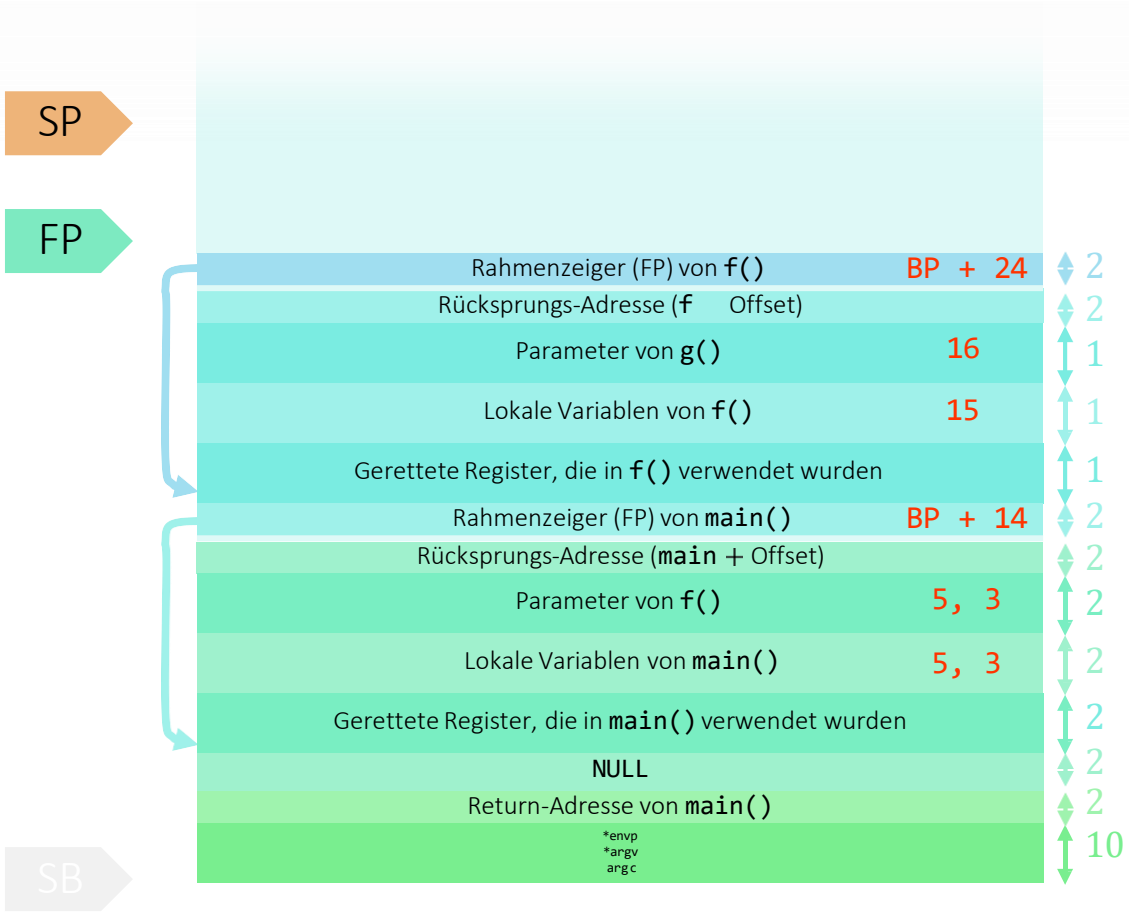
Rahmenzeiger verschieben



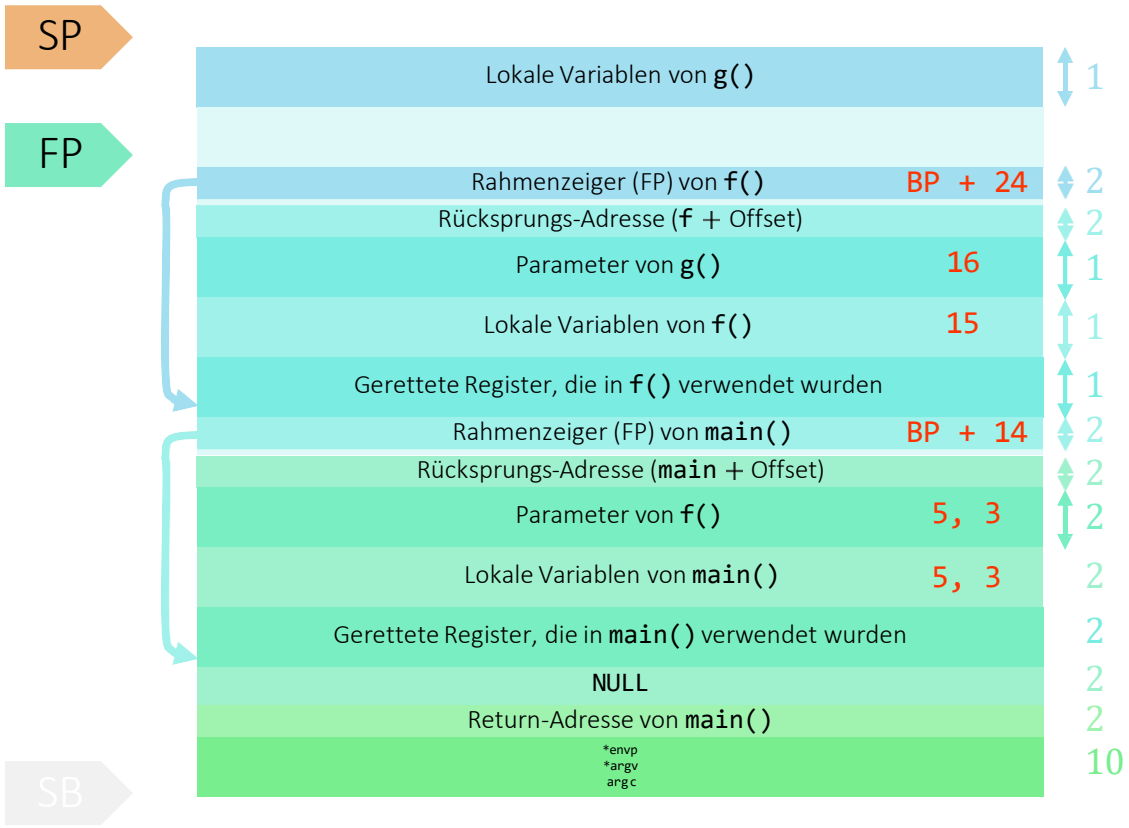
Stapelzeiger verschieben



Rücksprungadresse speichern

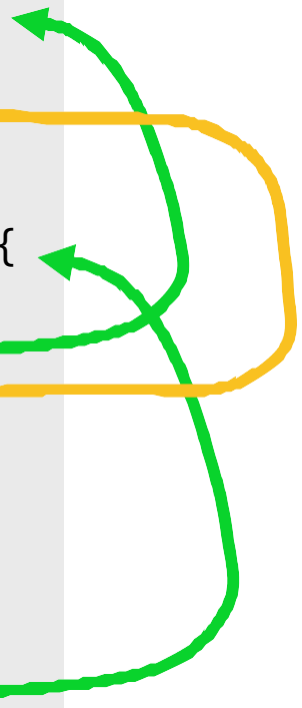


Unterprogrammaufruf

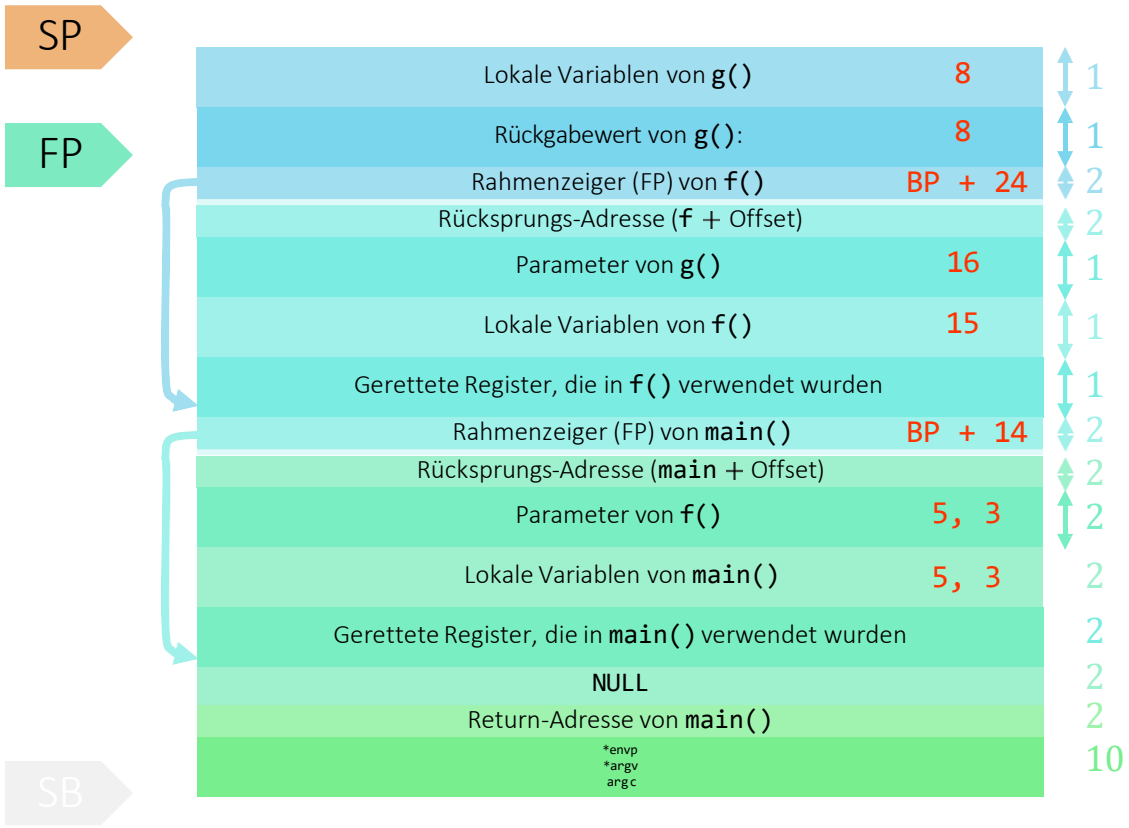


Aufruf der Return-Anweisung

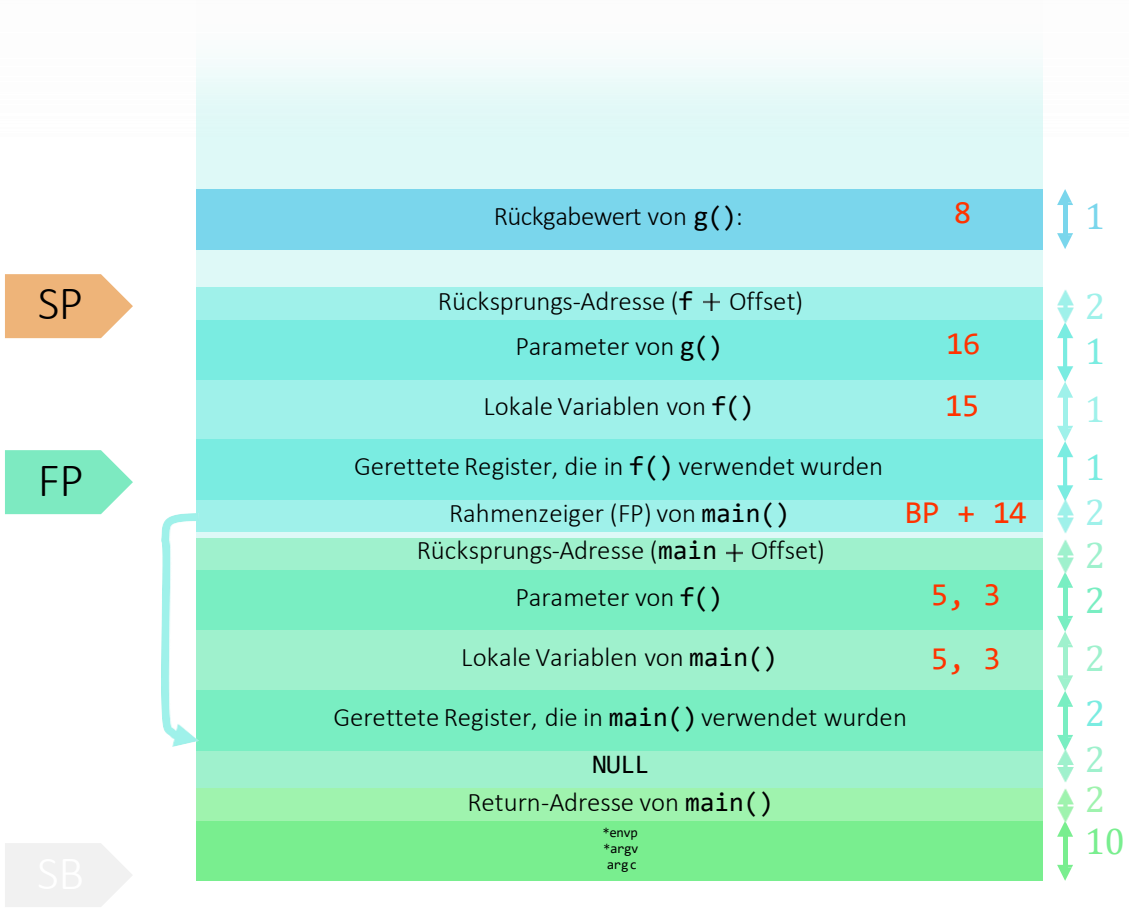
```
unsigned char g(unsigned char z) {
    unsigned char t = z / 2;
    return t;
}

unsigned char f(unsigned char x, unsigned char y) {
    unsigned char z = x * y;
    z = g(z+1);
    return z;
}

int main(int argc, char* argv[], char* envp[]) {
    unsigned char x = 5;
    unsigned char y = 3;
    f(x, y)
    return 0;
}
```



Erzeugen des Rückgabewertes

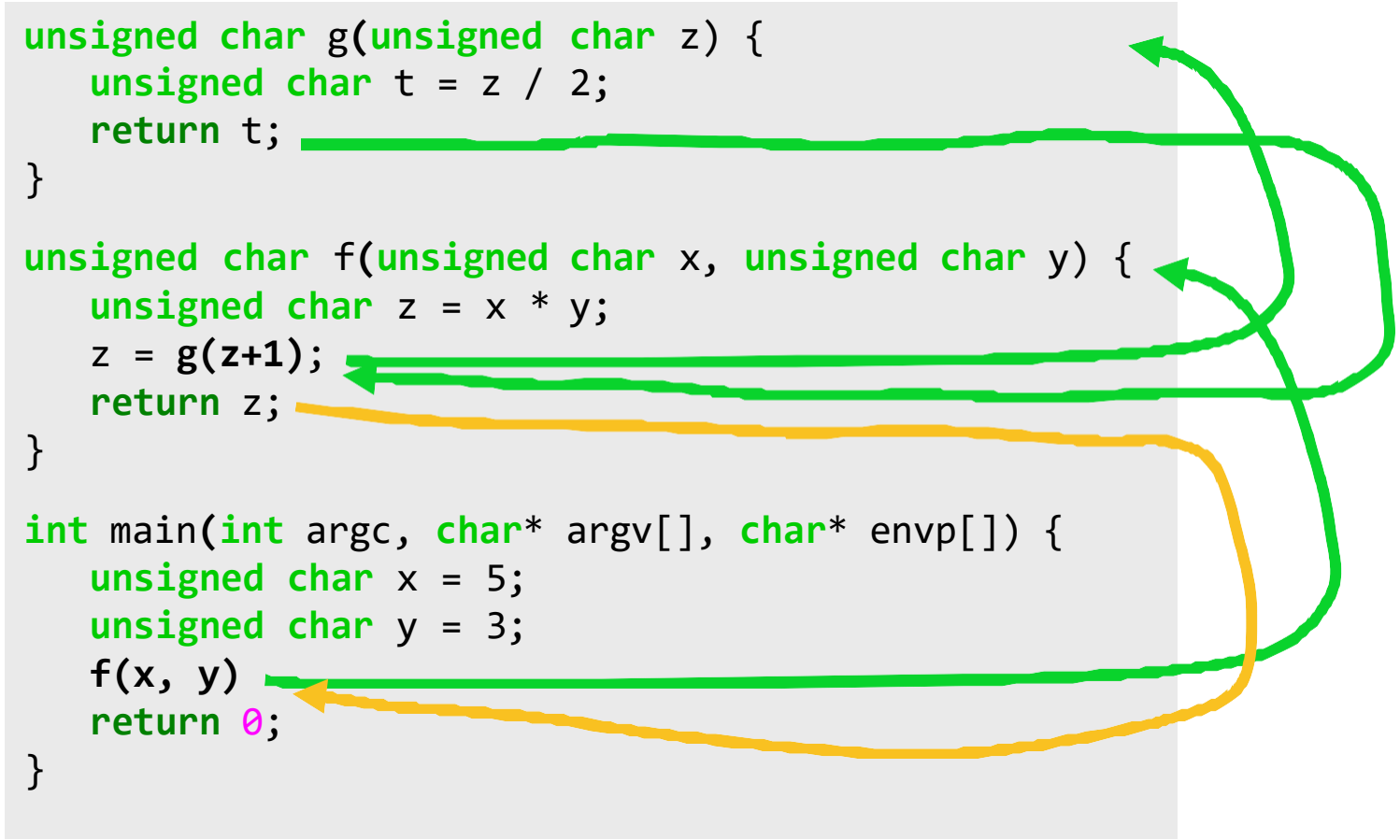


Stapel- und Rahmenzeiger wiederherstellen

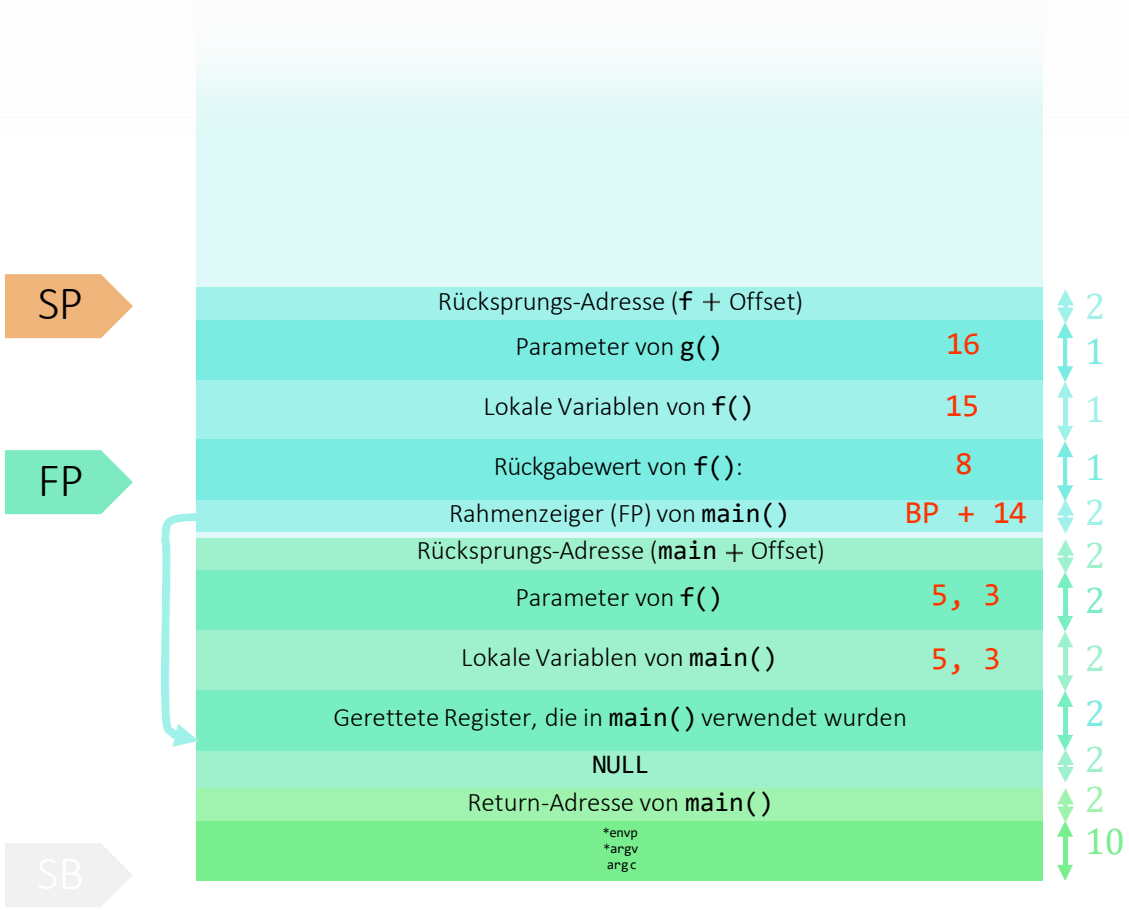


Rückkehr in die `main`

```
unsigned char g(unsigned char z) {
    unsigned char t = z / 2;
    return t;
}

unsigned char f(unsigned char x, unsigned char y) {
    unsigned char z = x * y;
    z = g(z+1);
    return z;
}

int main(int argc, char* argv[], char* envp[]) {
    unsigned char x = 5;
    unsigned char y = 3;
    f(x, y)
    return 0;
}
```



Erzeugen des Rückgabewertes



Stapel- und Rahmenzeiger wiederherstellen