Verlässliche Echtzeitsysteme

Übungen zur Vorlesung

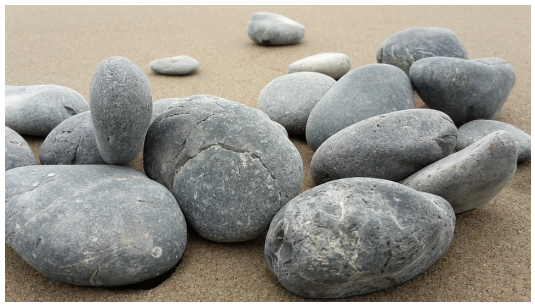
Dynamische Stackbedarfsanalyse

Phillip Raffeck, Florian Schmaus, Simon Schuster

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)
https://www4.cs.fau.de

Sommersemester 2020

Analyse Harter Echtzeitsysteme



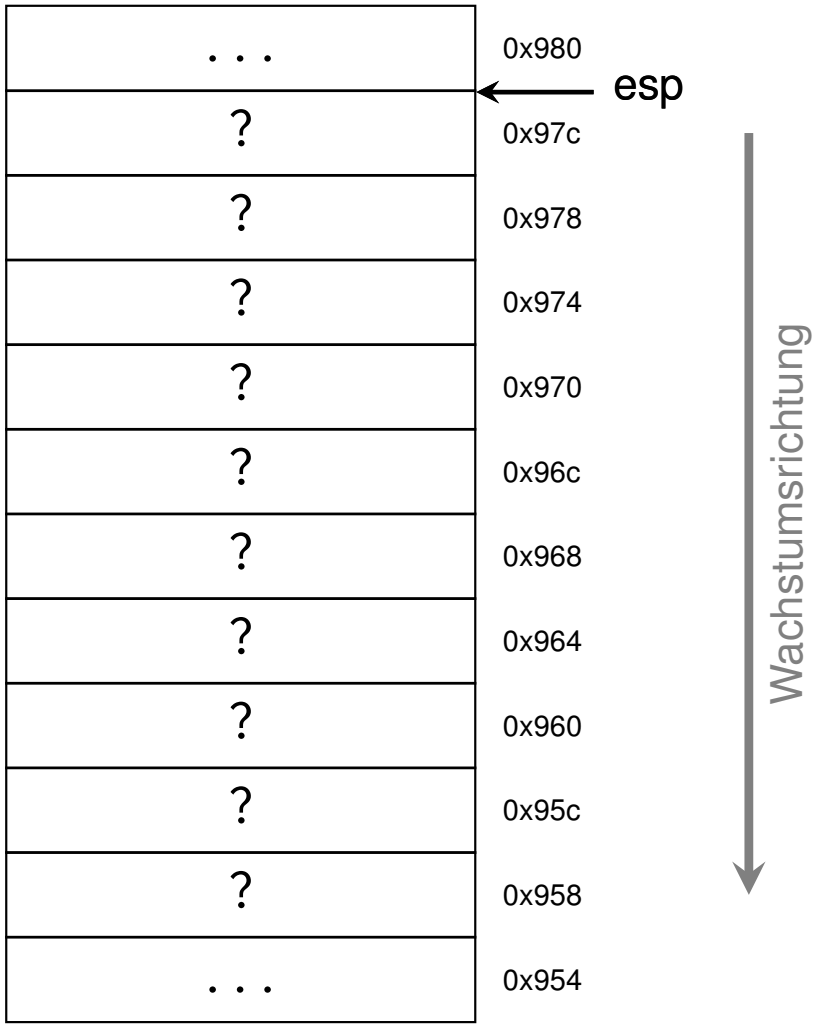
- Harte, verlässliche Echtzeitsysteme
  - Garantien über Ressourcenbedarf notwendig
    ☞ statische Analyse unabdingbar
- Mögliche Ressourcen: Speicherbedarf, Laufzeit, etc.
- Übung: Analyse des Stackbedarfs einer Bibliothek
- Stack-Analyse

1. Dynamisch: Wasserstandstechnik
2. Statisch: „Eigenbau“ und aiT (Stack-Analyzer der a^3 Suite)

Beispiel: Programmstapel

```
int main(void) {
  return f(4, 2);
}
```

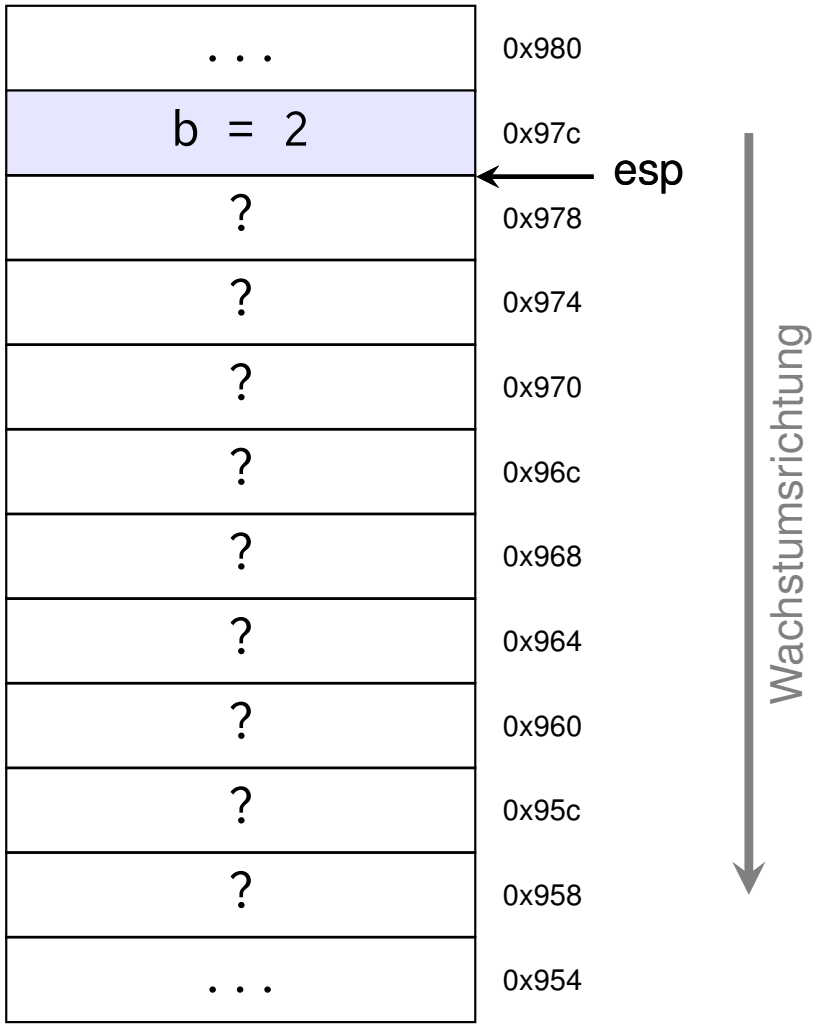
```
52a:    push ebp
52b:    mov ebp,esp
52d:    lea ...
534:    or ...
538:    lea ...
53f:    push 0x2
541:    push 0x4
543:    call 4fd <f>
548:    add esp,0x8
54b:    leave
54c:    ret
```



Beispiel: Programmstapel

```
int main(void) {
  return f(4, 2);
}
```

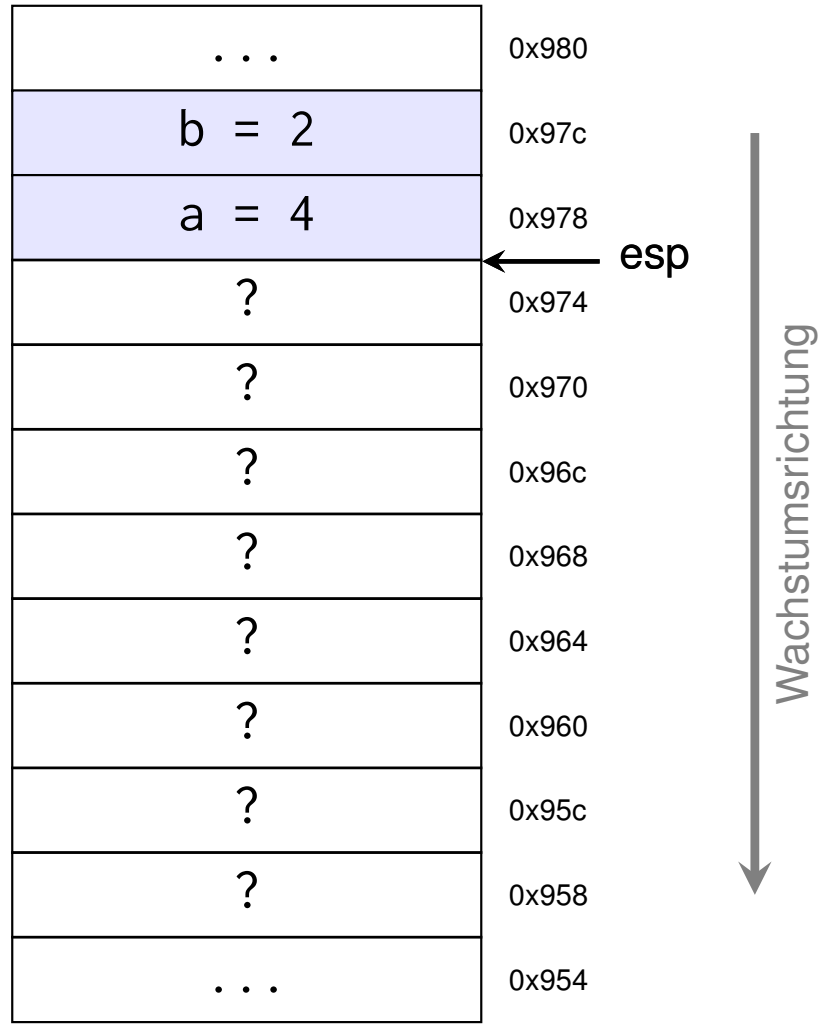
```
52a:    push ebp
52b:    mov ebp,esp
52d:    lea ...
534:    or ...
538:    lea ...
53f:    push 0x2
541:    push 0x4
543:    call 4fd <f>
548:    add esp,0x8
54b:    leave
54c:    ret
```



Beispiel: Programmstapel

```
int main(void) {
  return f(4, 2);
}
```

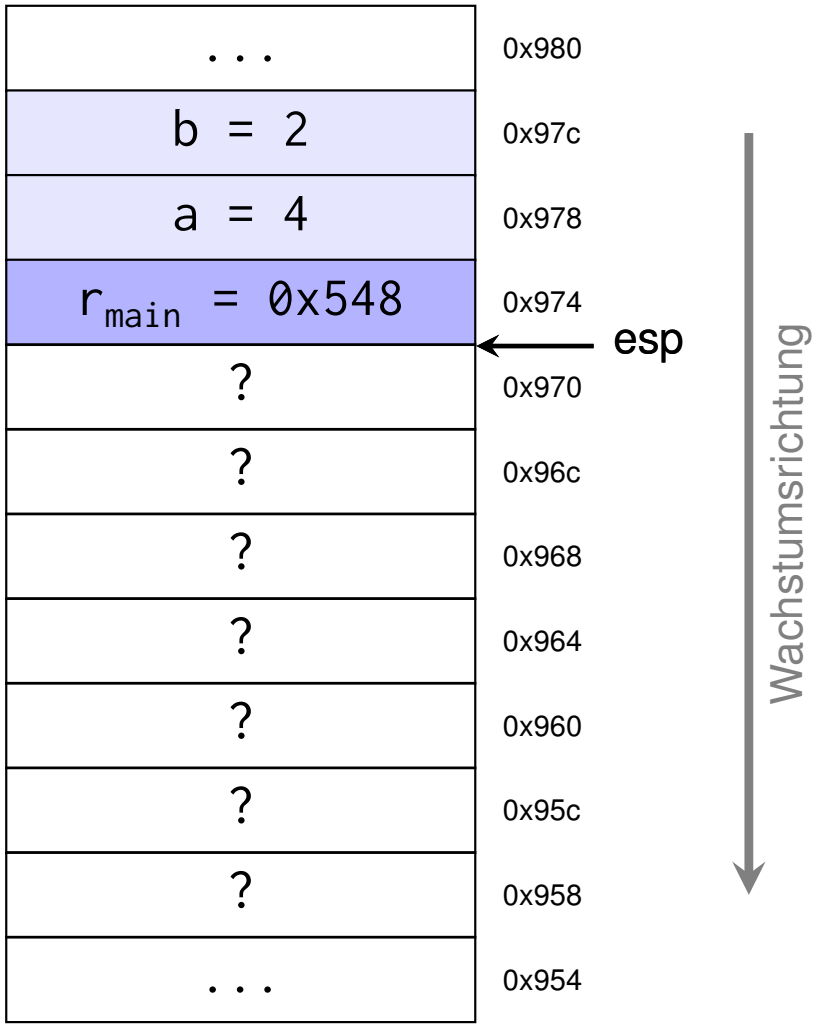
```
52a:    push ebp
52b:    mov ebp,esp
52d:    lea ...
534:    or ...
538:    lea ...
53f:    push 0x2
541:    push 0x4
543:    call 4fd <f>
548:    add esp,0x8
54b:    leave
54c:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

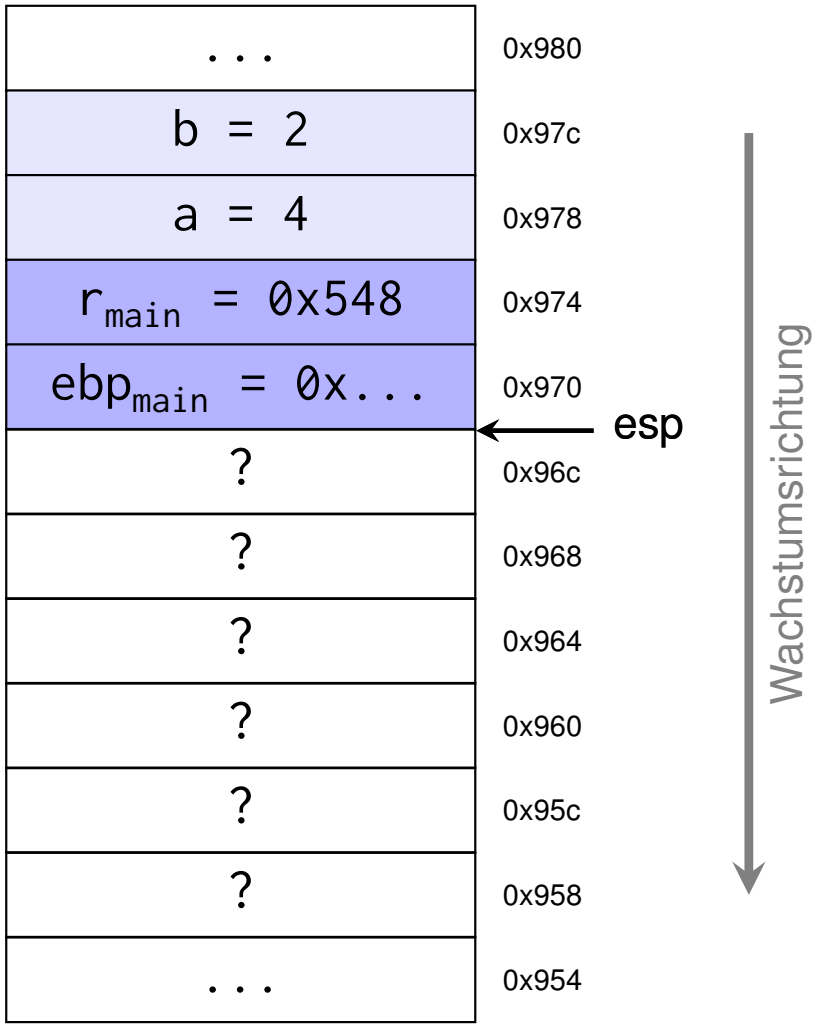
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

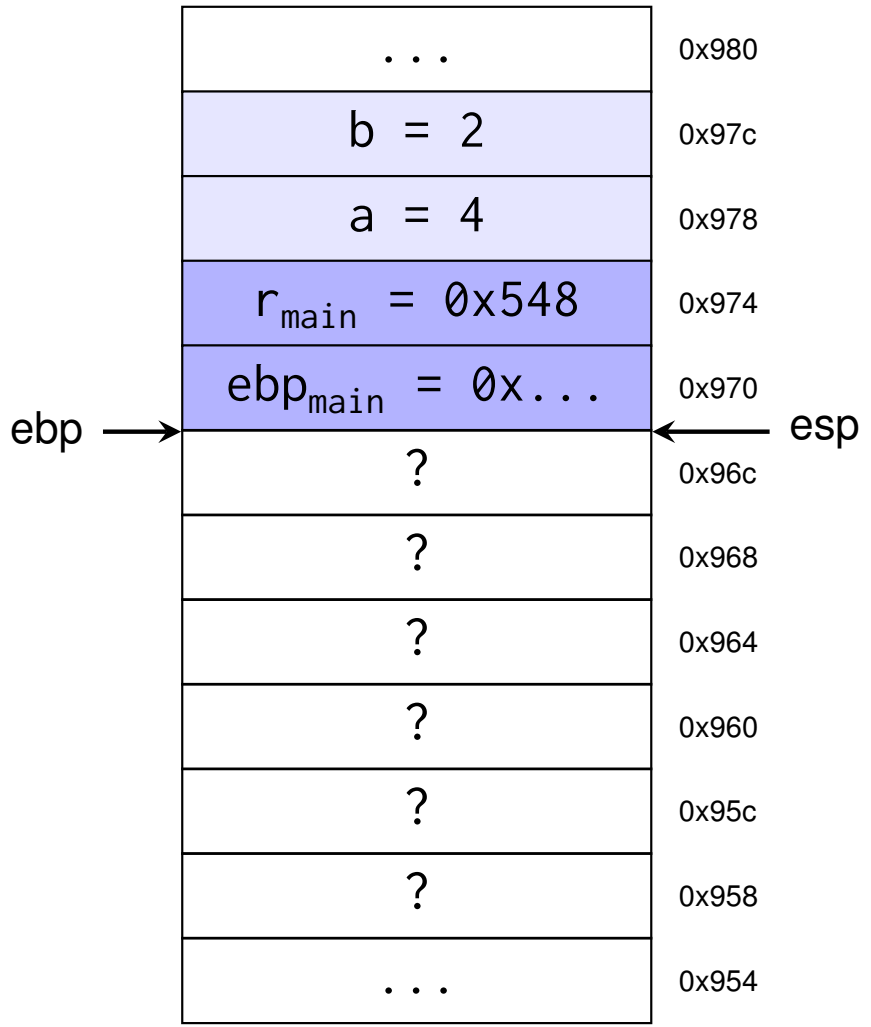
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

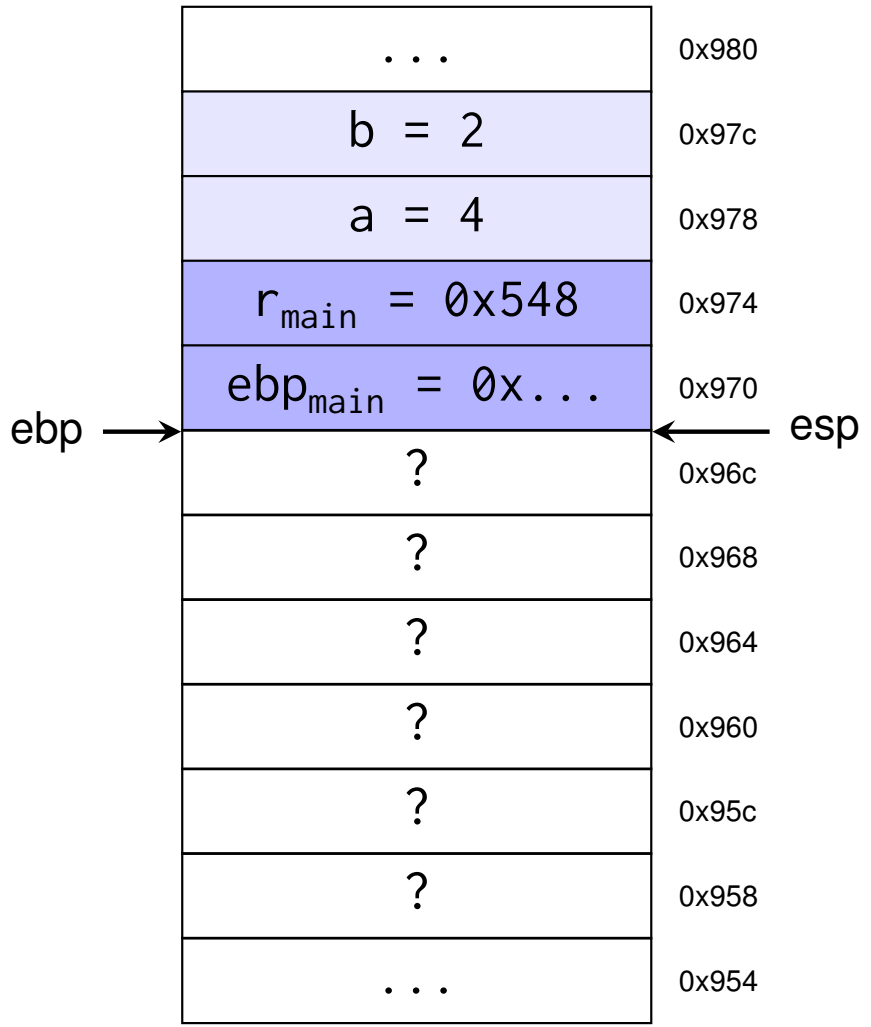
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

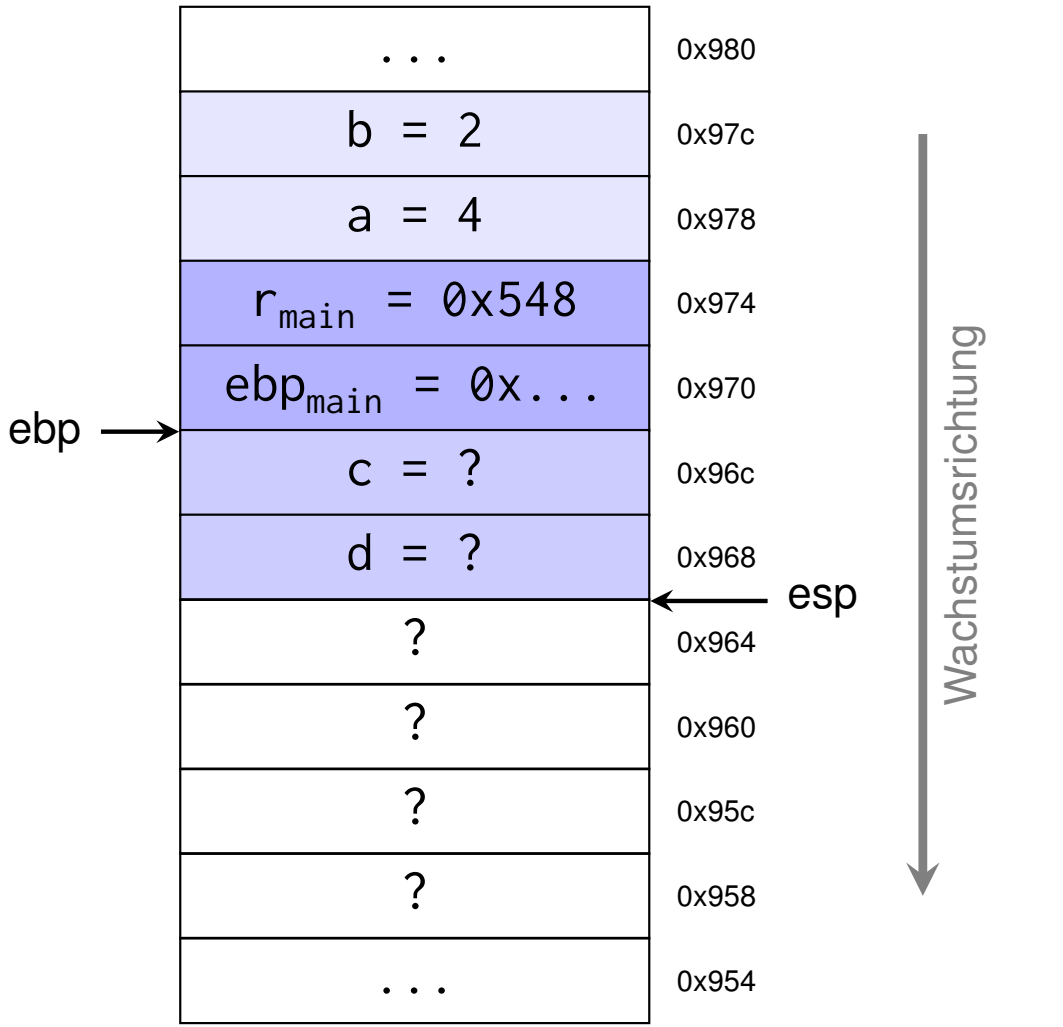
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

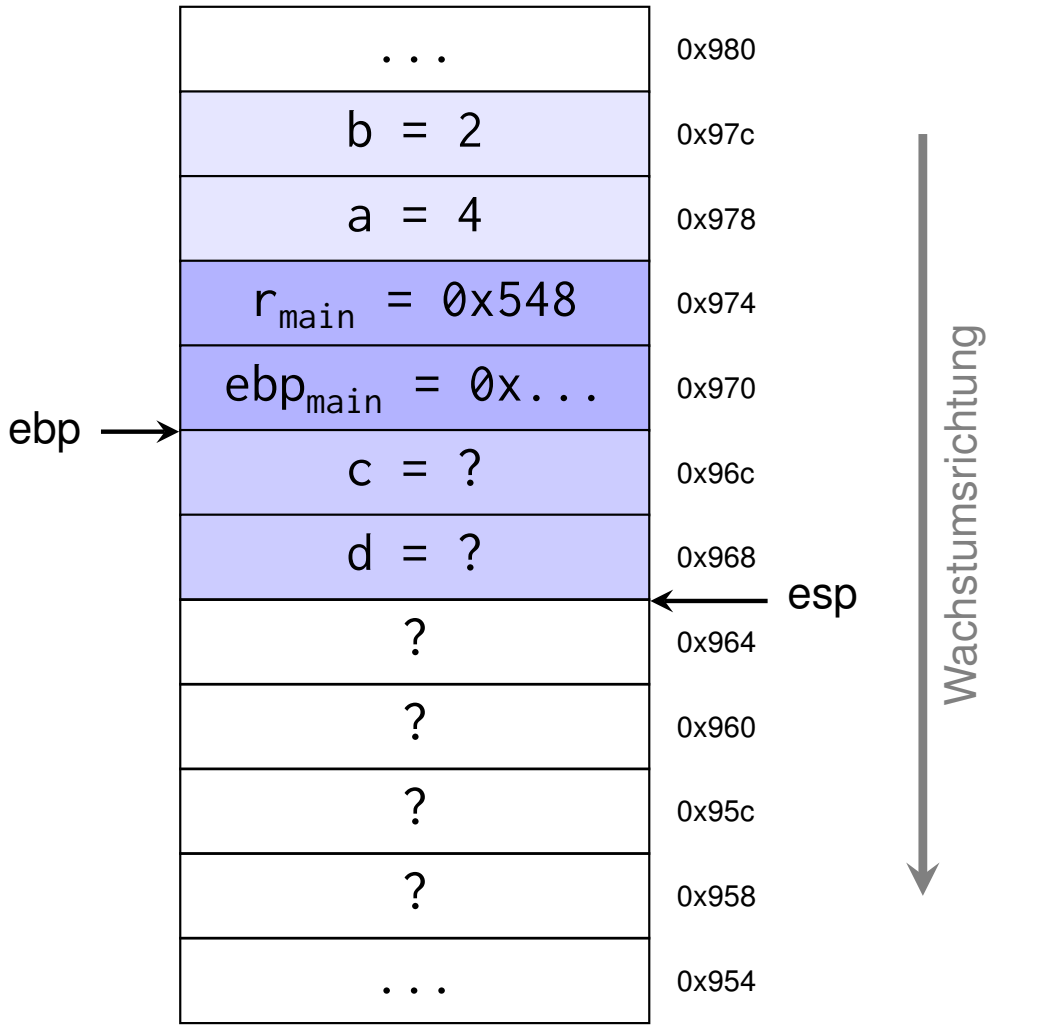
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

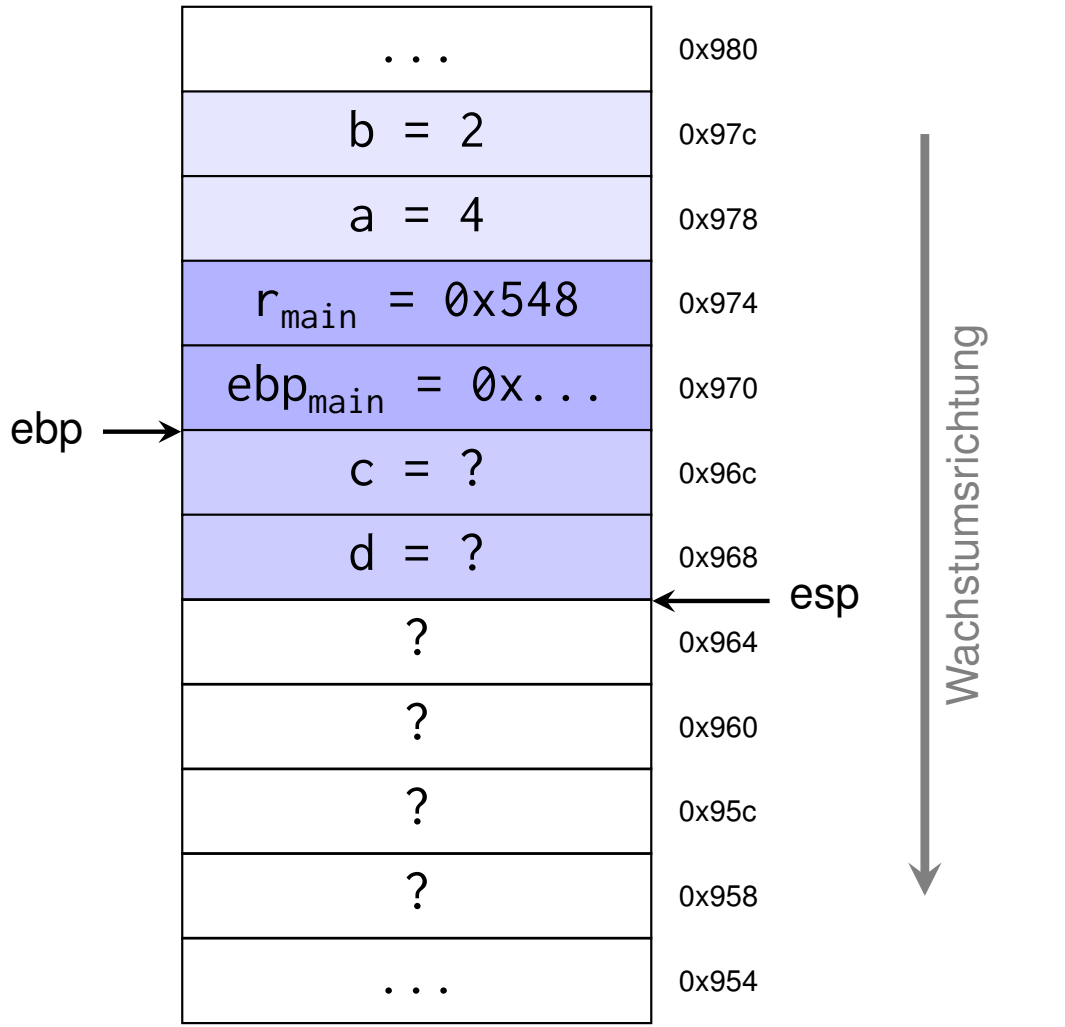
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

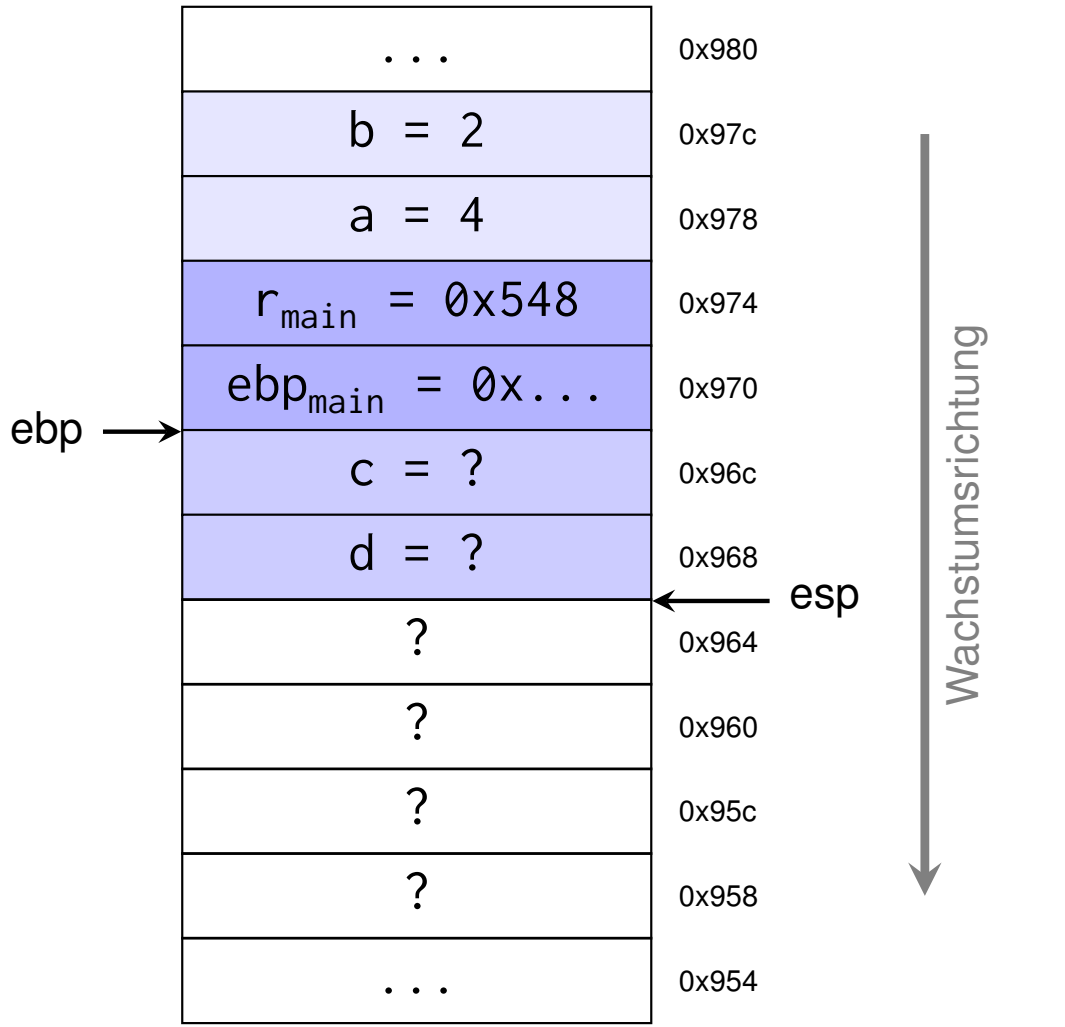
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

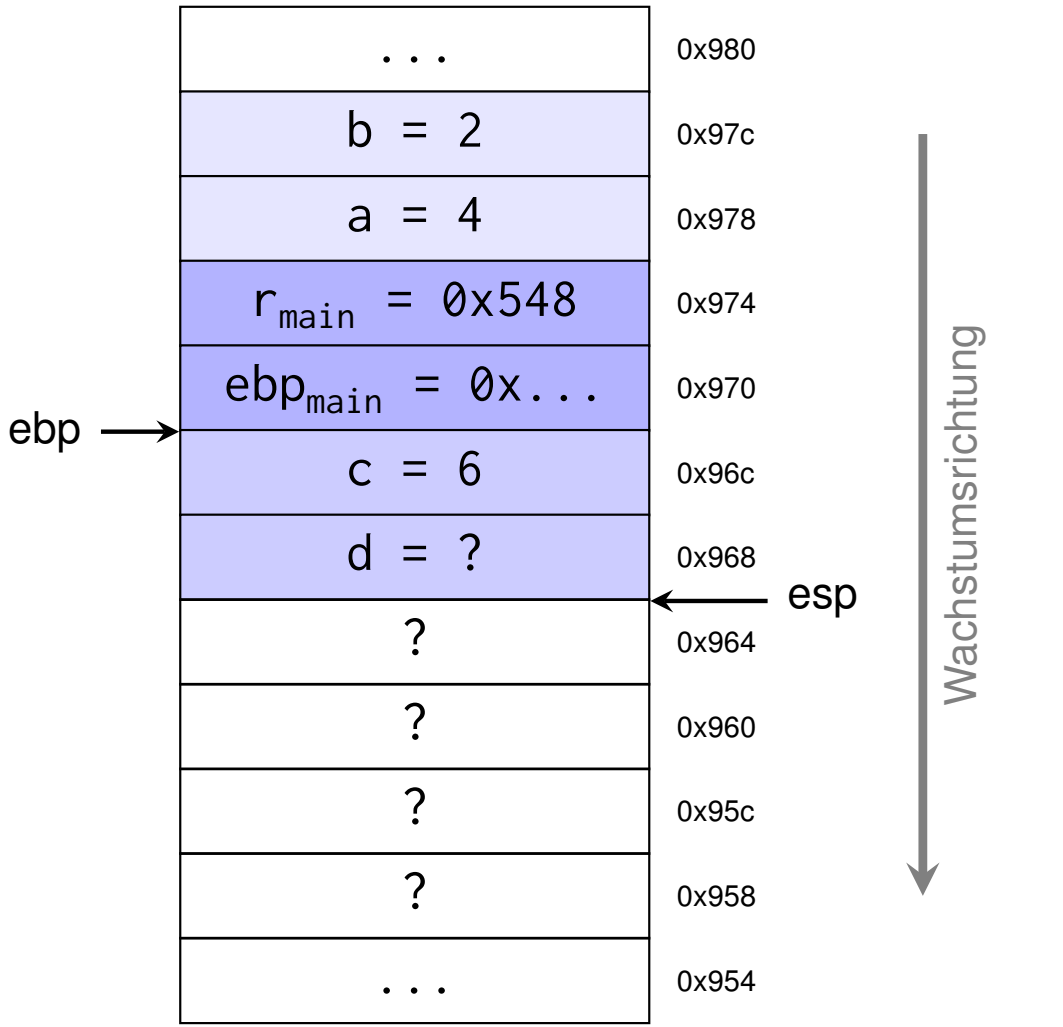
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

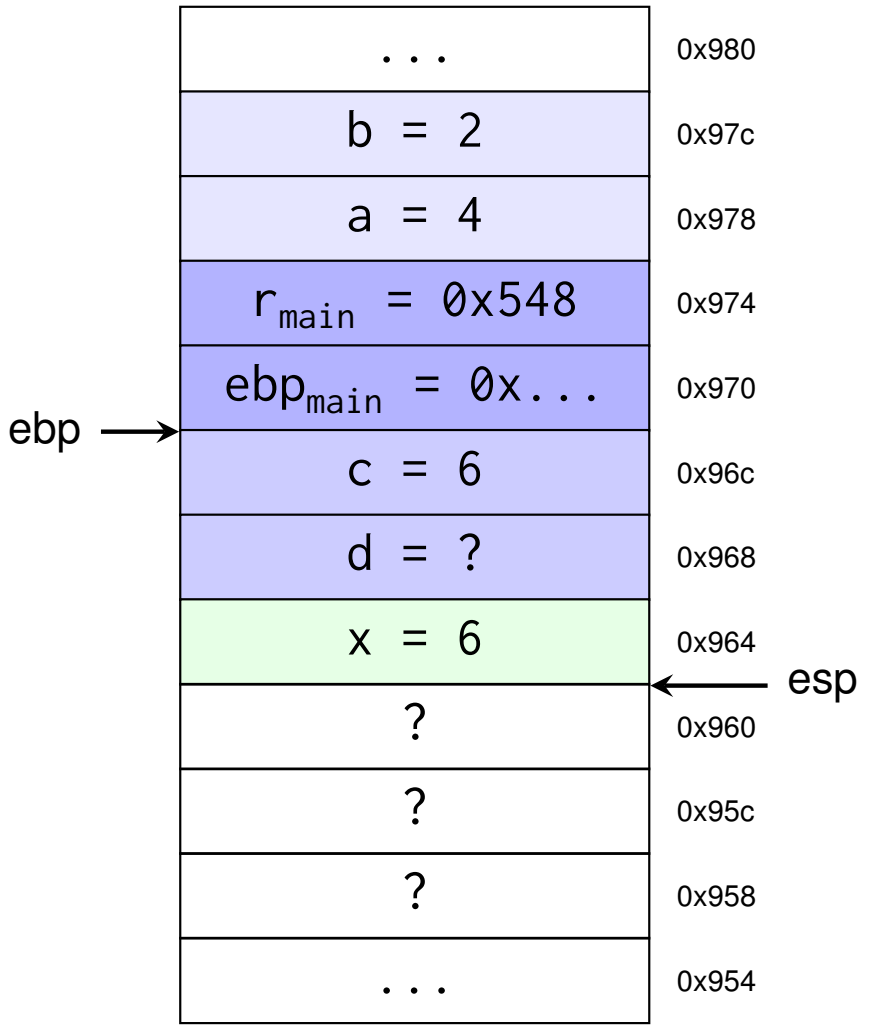
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

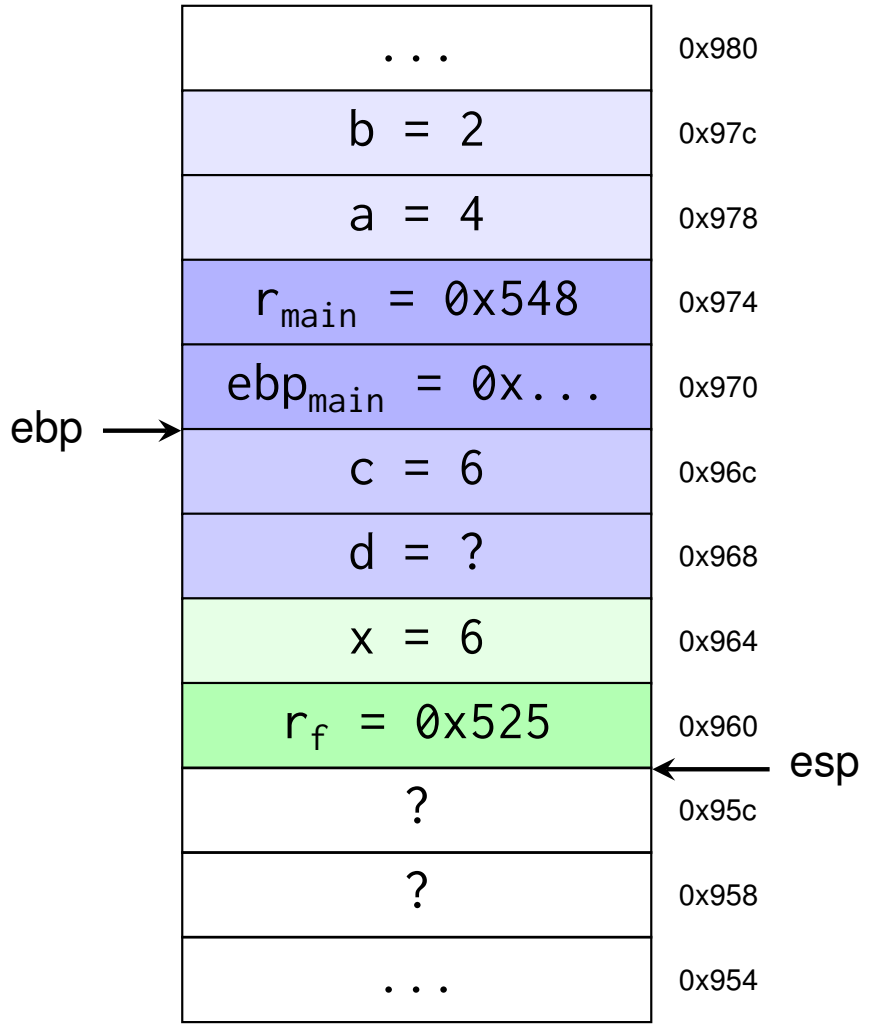
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int g(int x) {
  int y = x + 1;
  return y;
}
```

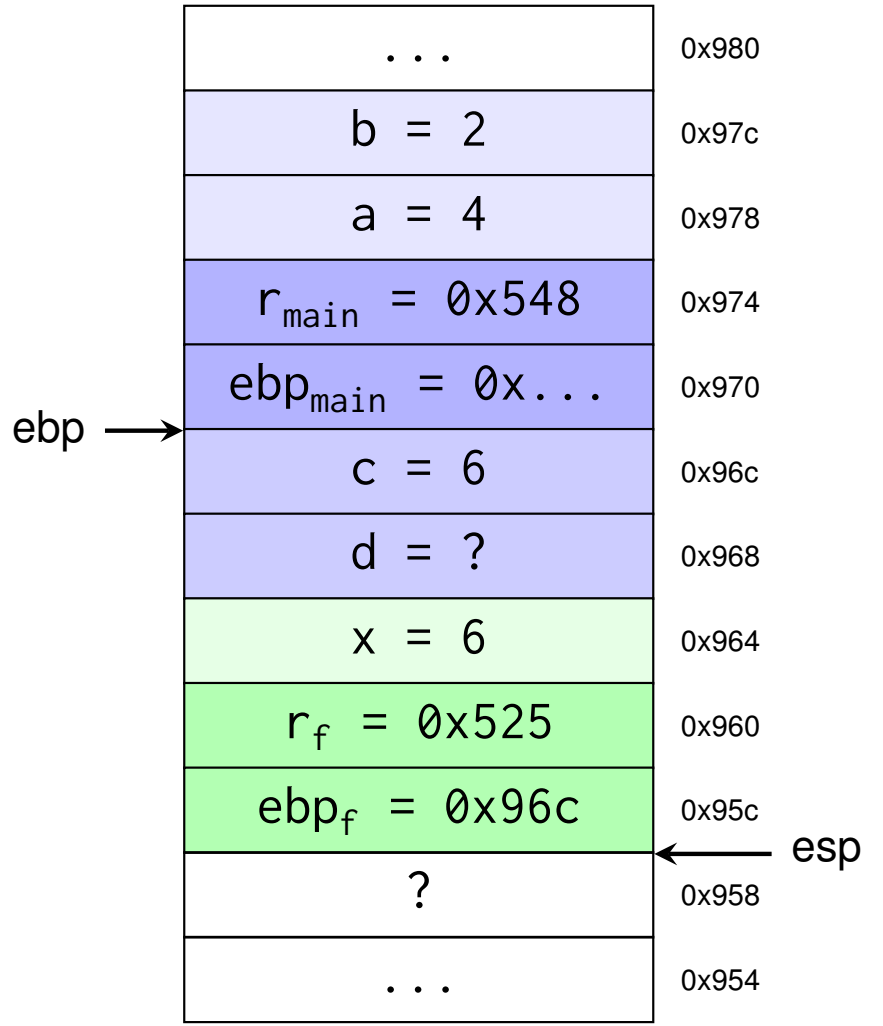
```
4e9:    push ebp
4ea:    mov ebp,esp
4ec:    sub esp,0x4
4ef:    mov eax,DWORD PTR [ebp+0x8]
4f2:    add eax,0x1
4f5:    mov DWORD PTR [ebp-0x4],eax
4f8:    mov eax,DWORD PTR [ebp-0x4]
4fb:    leave
4fc:    ret
```



Beispiel: Programmstapel

```
int g(int x) {
  int y = x + 1;
  return y;
}
```

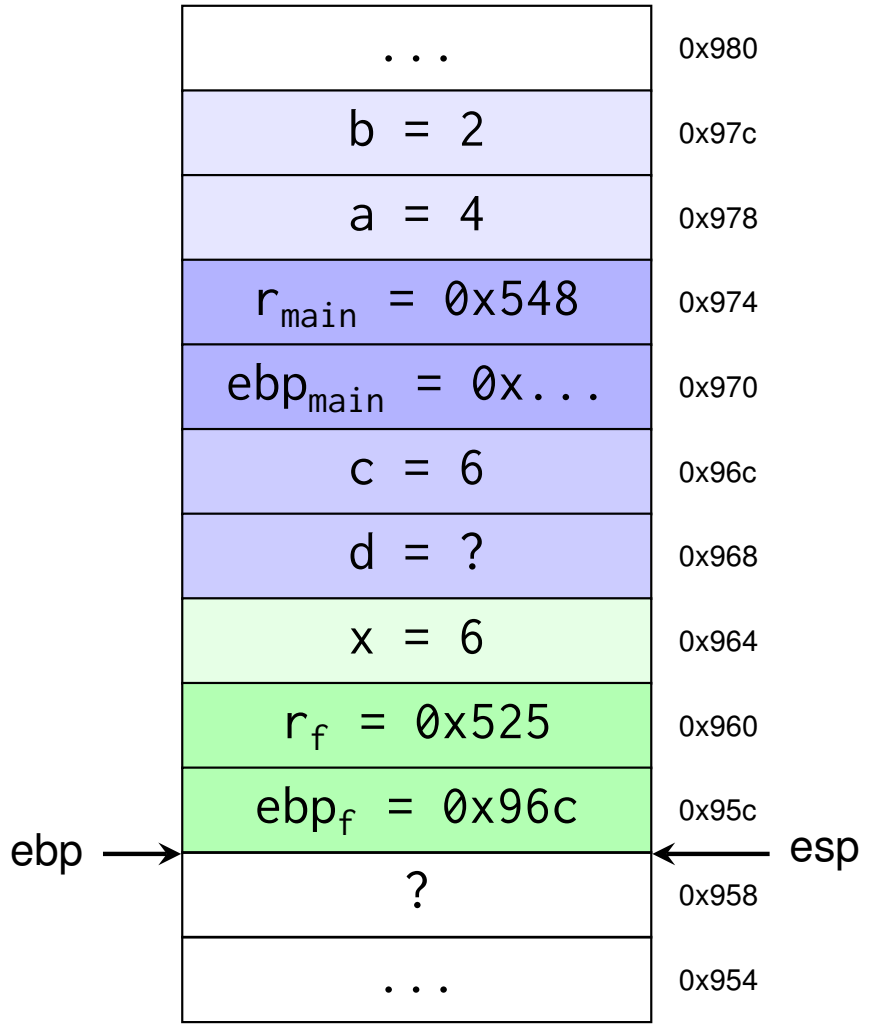
```
4e9:    push ebp
4ea:    mov ebp,esp
4ec:    sub esp,0x4
4ef:    mov eax,DWORD PTR [ebp+0x8]
4f2:    add eax,0x1
4f5:    mov DWORD PTR [ebp-0x4],eax
4f8:    mov eax,DWORD PTR [ebp-0x4]
4fb:    leave
4fc:    ret
```



Beispiel: Programmstapel

```
int g(int x) {
  int y = x + 1;
  return y;
}
```

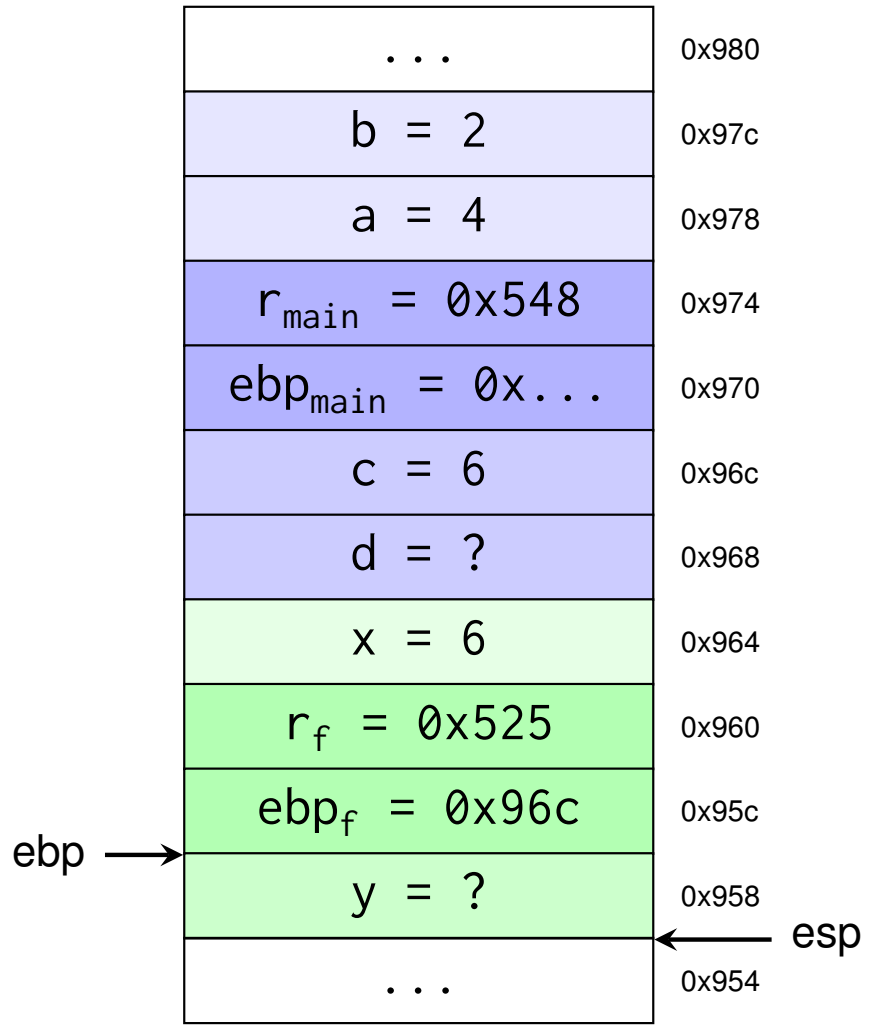
```
4e9:    push ebp
4ea:    mov ebp,esp
4ec:    sub esp,0x4
4ef:    mov eax,DWORD PTR [ebp+0x8]
4f2:    add eax,0x1
4f5:    mov DWORD PTR [ebp-0x4],eax
4f8:    mov eax,DWORD PTR [ebp-0x4]
4fb:    leave
4fc:    ret
```



Beispiel: Programmstapel

```
int g(int x) {
  int y = x + 1;
  return y;
}
```

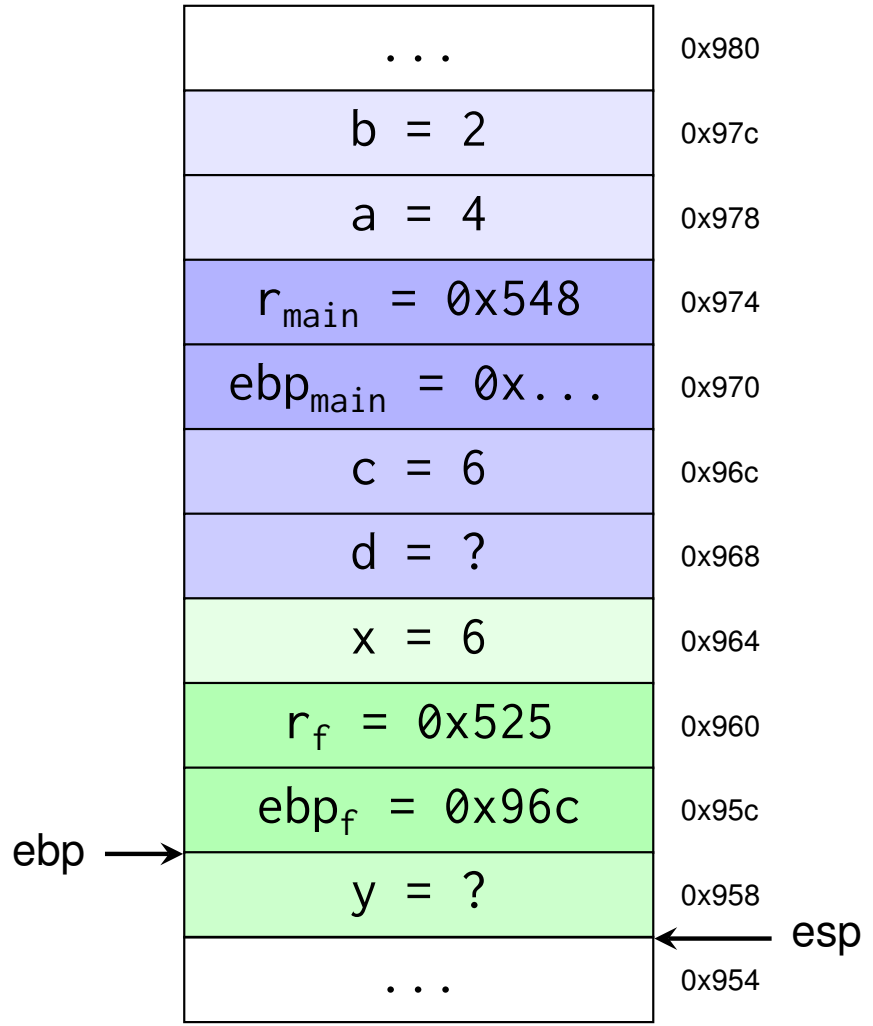
```
4e9:    push ebp
4ea:    mov ebp,esp
4ec:    sub esp,0x4
4ef:    mov eax,DWORD PTR [ebp+0x8]
4f2:    add eax,0x1
4f5:    mov DWORD PTR [ebp-0x4],eax
4f8:    mov eax,DWORD PTR [ebp-0x4]
4fb:    leave
4fc:    ret
```



Beispiel: Programmstapel

```
int g(int x) {
  int y = x + 1;
  return y;
}
```

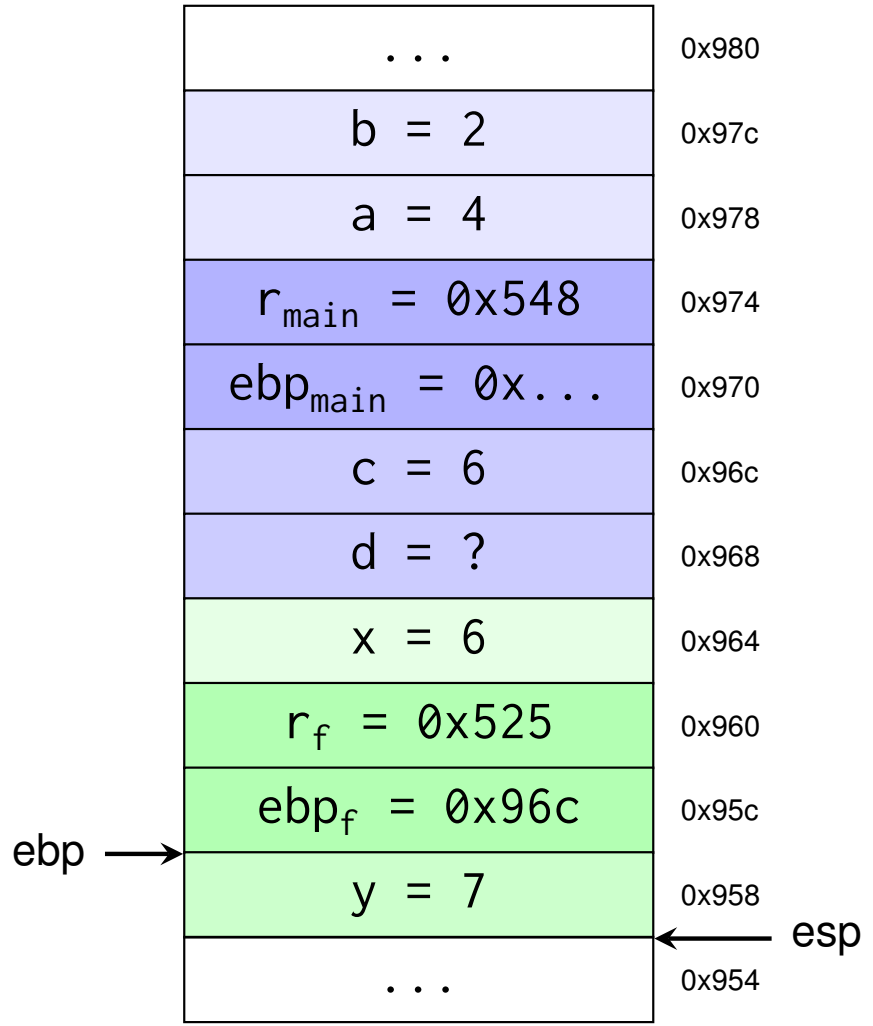
```
4e9:    push ebp
4ea:    mov ebp,esp
4ec:    sub esp,0x4
4ef:    mov eax,DWORD PTR [ebp+0x8]
4f2:    add eax,0x1
4f5:    mov DWORD PTR [ebp-0x4],eax
4f8:    mov eax,DWORD PTR [ebp-0x4]
4fb:    leave
4fc:    ret
```



Beispiel: Programmstapel

```
int g(int x) {
  int y = x + 1;
  return y;
}
```

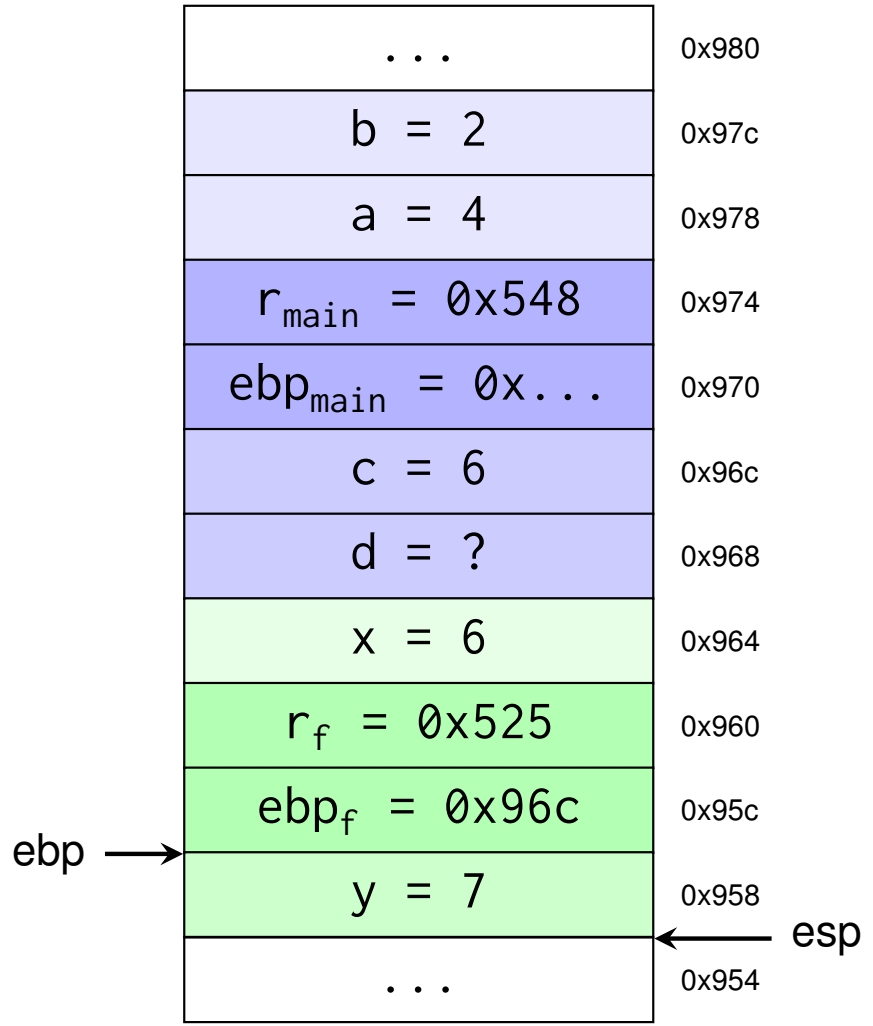
```
4e9:    push ebp
4ea:    mov ebp,esp
4ec:    sub esp,0x4
4ef:    mov eax,DWORD PTR [ebp+0x8]
4f2:    add eax,0x1
4f5:    mov DWORD PTR [ebp-0x4],eax
4f8:    mov eax,DWORD PTR [ebp-0x4]
4fb:    leave
4fc:    ret
```



Beispiel: Programmstapel

```
int g(int x) {
  int y = x + 1;
  return y;
}
```

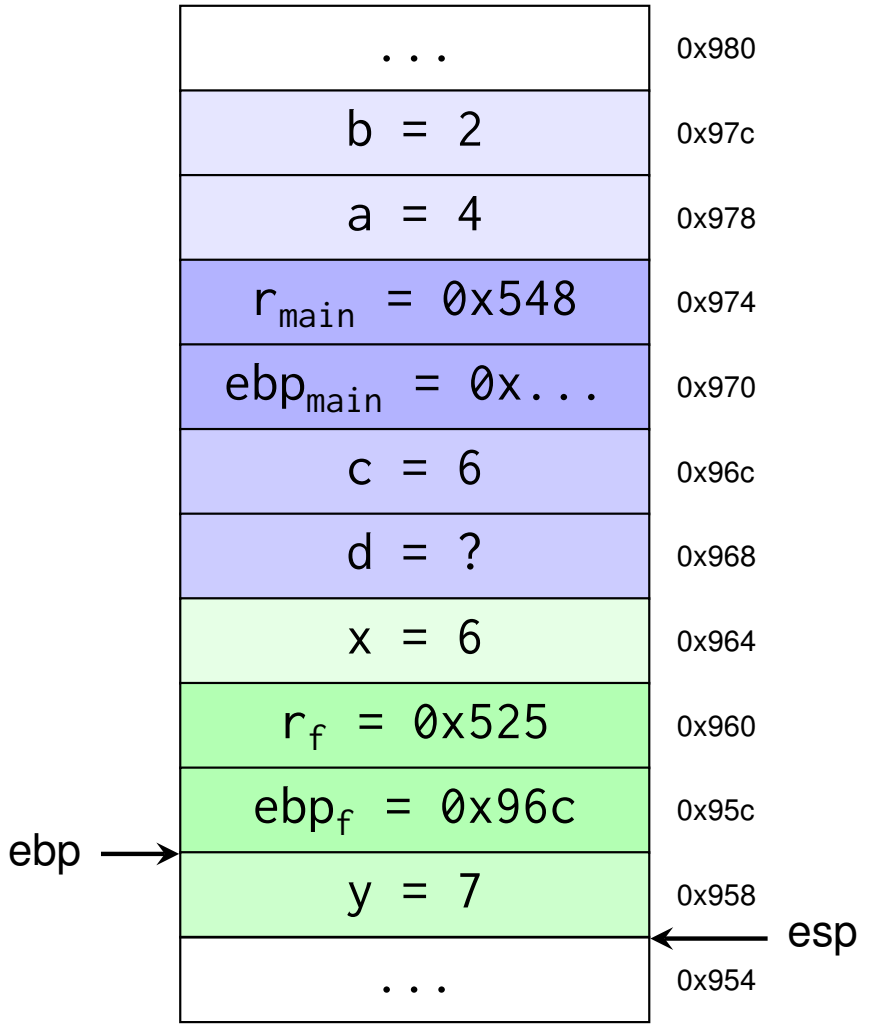
```
4e9:    push ebp
4ea:    mov ebp,esp
4ec:    sub esp,0x4
4ef:    mov eax,DWORD PTR [ebp+0x8]
4f2:    add eax,0x1
4f5:    mov DWORD PTR [ebp-0x4],eax
4f8:    mov eax,DWORD PTR [ebp-0x4]
4fb:    leave
4fc:    ret
```



Beispiel: Programmstapel

```
int g(int x) {
  int y = x + 1;
  return y;
}
```

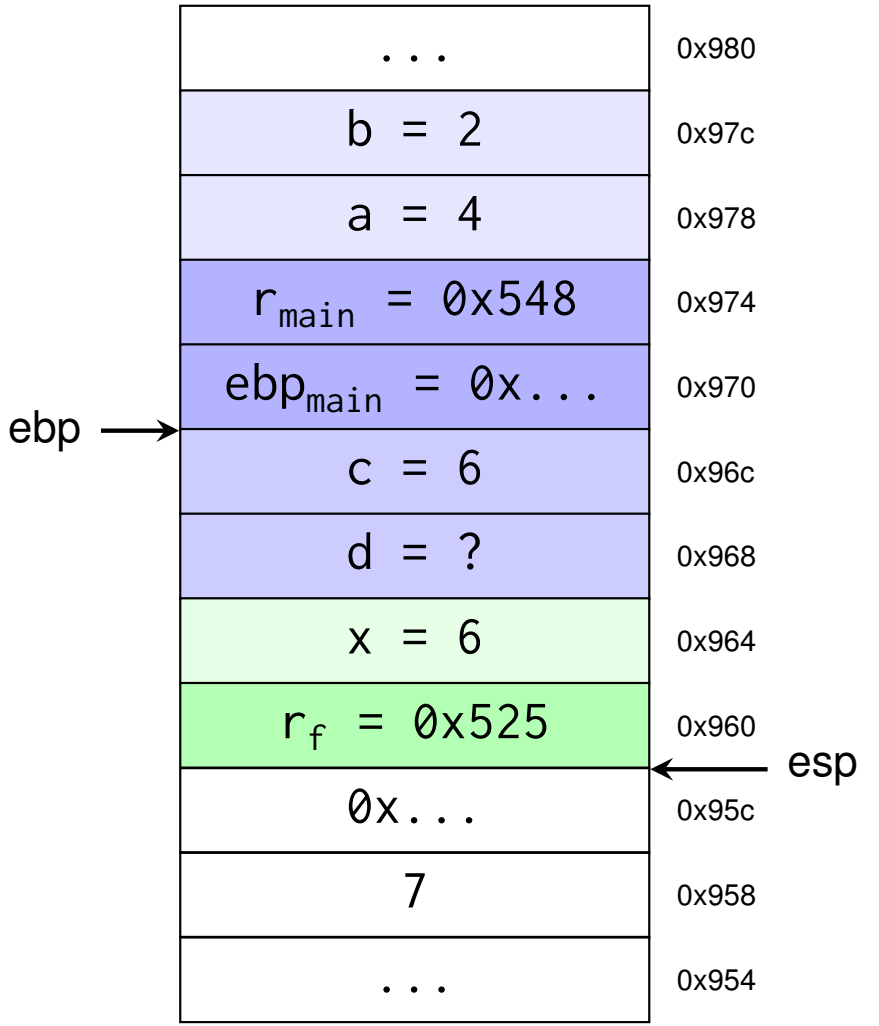
```
4e9:    push ebp
4ea:    mov ebp,esp
4ec:    sub esp,0x4
4ef:    mov eax,DWORD PTR [ebp+0x8]
4f2:    add eax,0x1
4f5:    mov DWORD PTR [ebp-0x4],eax
4f8:    mov eax,DWORD PTR [ebp-0x4]
4fb:    leave
4fc:    ret
```



Beispiel: Programmstapel

```
int g(int x) {
  int y = x + 1;
  return y;
}
```

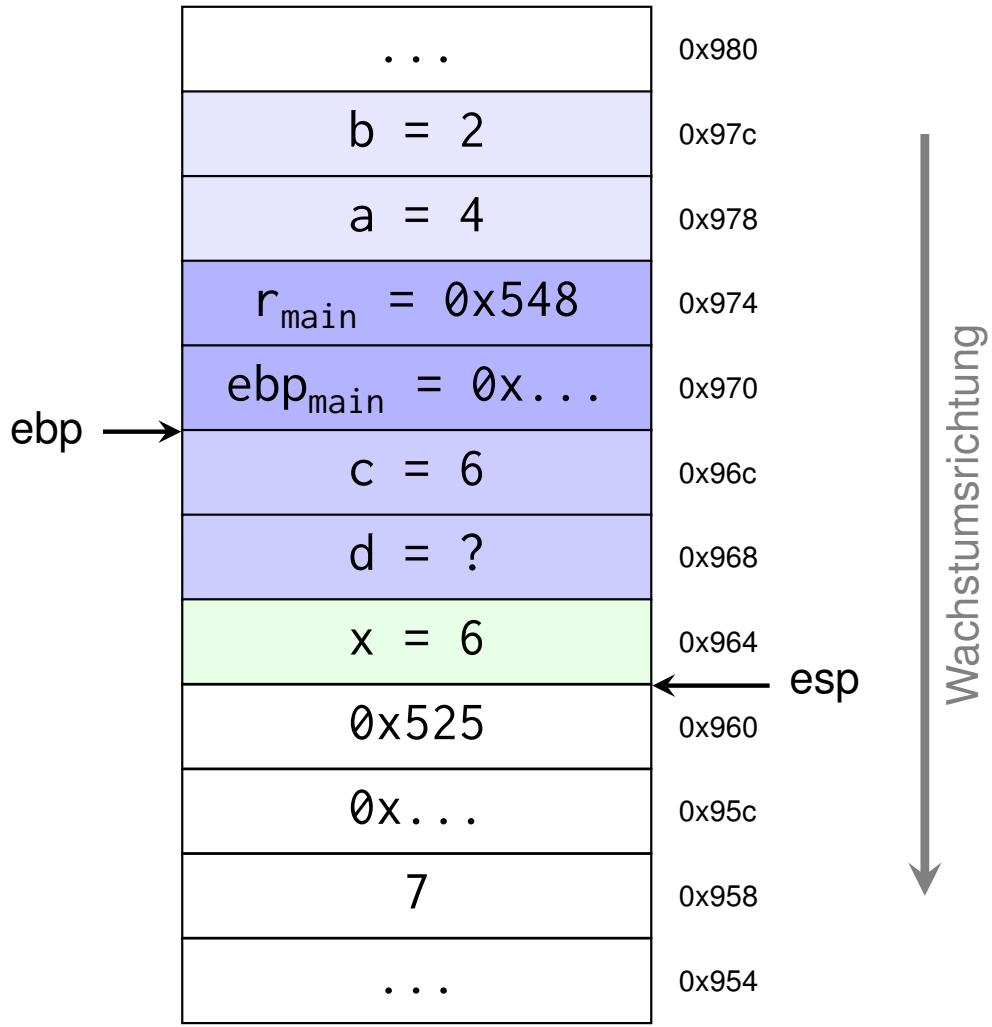
```
4e9:    push ebp
4ea:    mov ebp,esp
4ec:    sub esp,0x4
4ef:    mov eax,DWORD PTR [ebp+0x8]
4f2:    add eax,0x1
4f5:    mov DWORD PTR [ebp-0x4],eax
4f8:    mov eax,DWORD PTR [ebp-0x4]
4fb:    leave
4fc:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

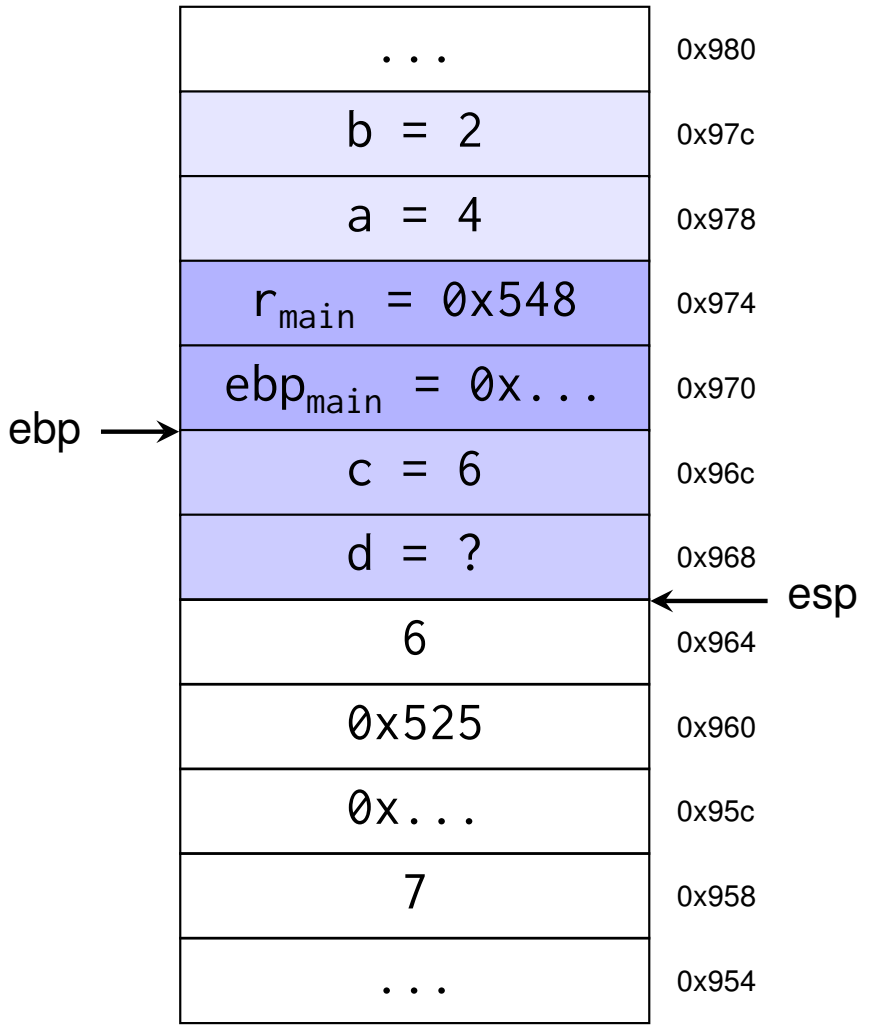
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

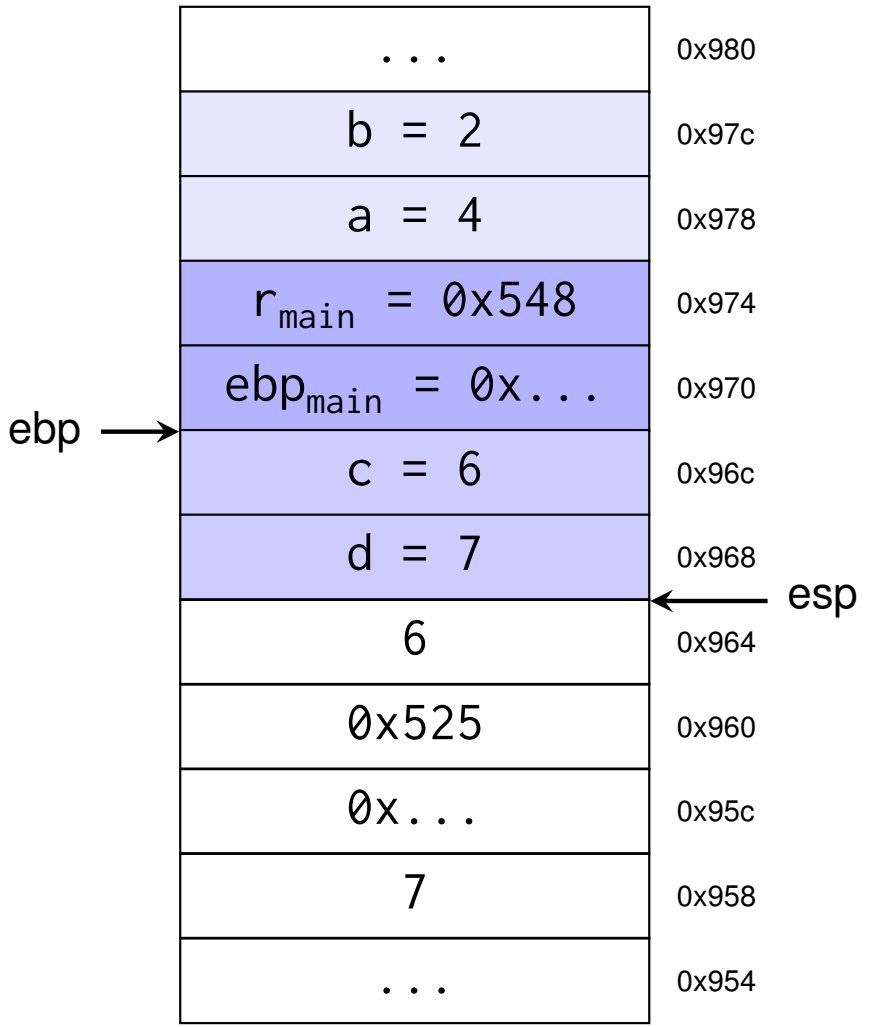
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

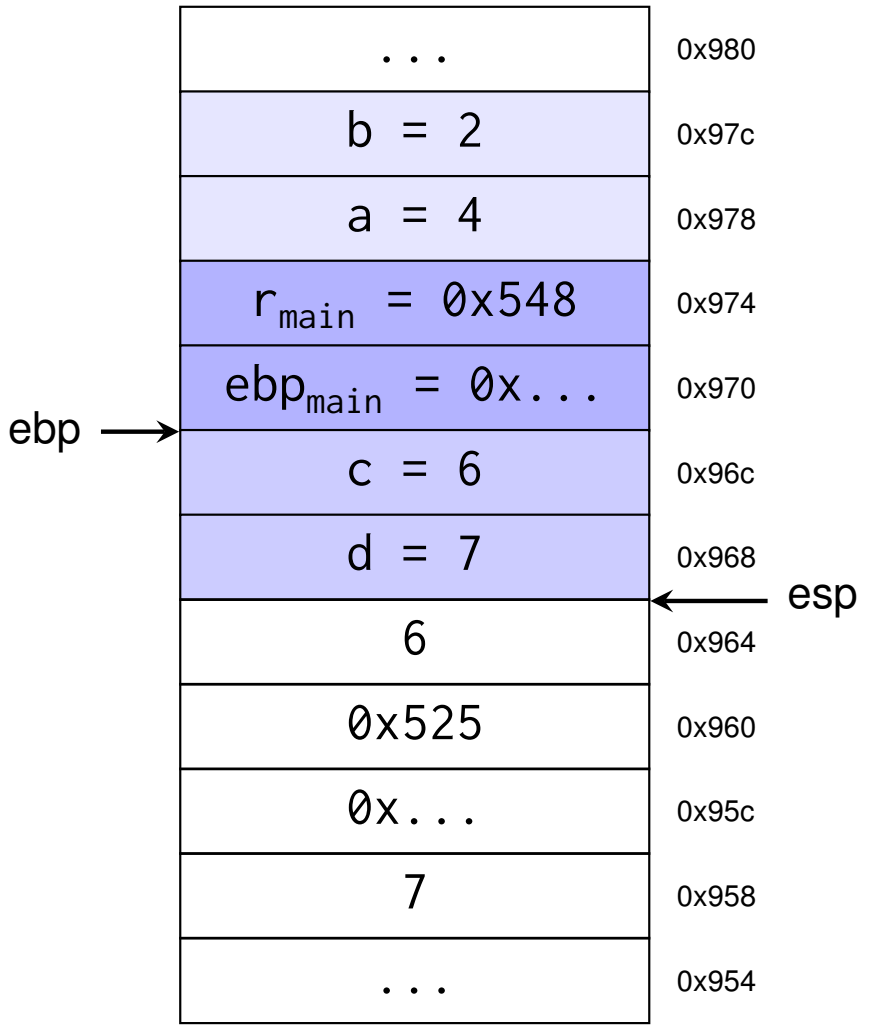
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

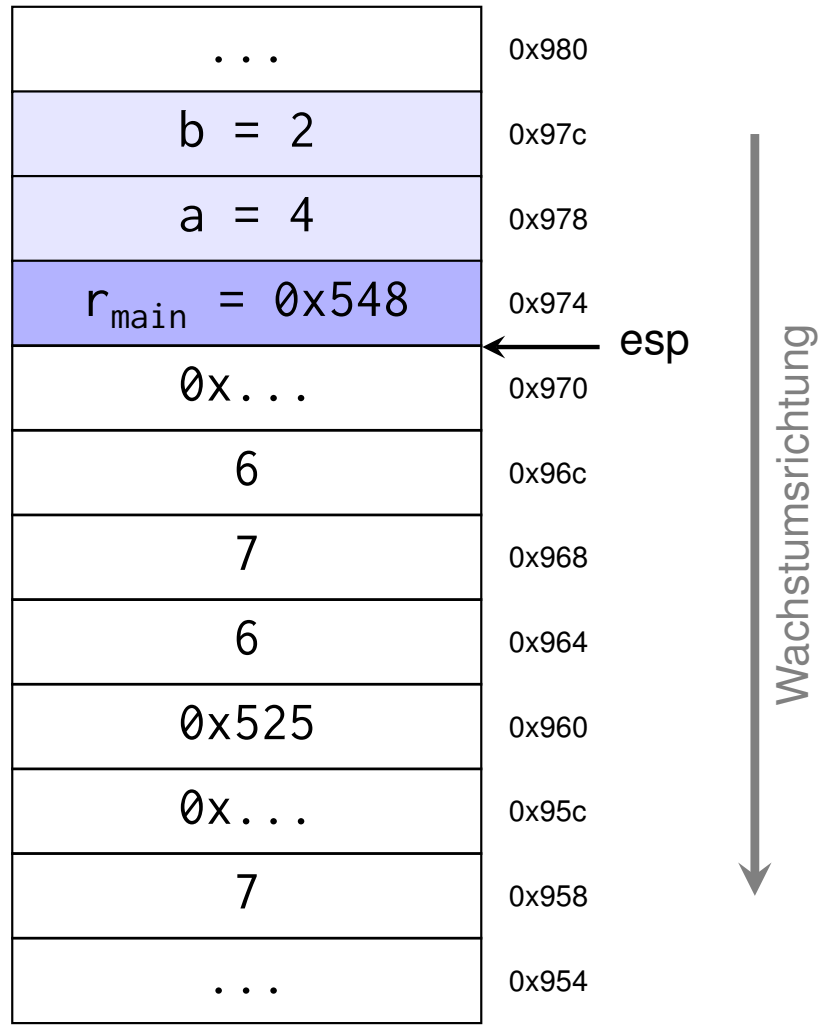
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int f(int a, int b) {
  int c = a + b;
  int d = g(c);
  return d;
}
```

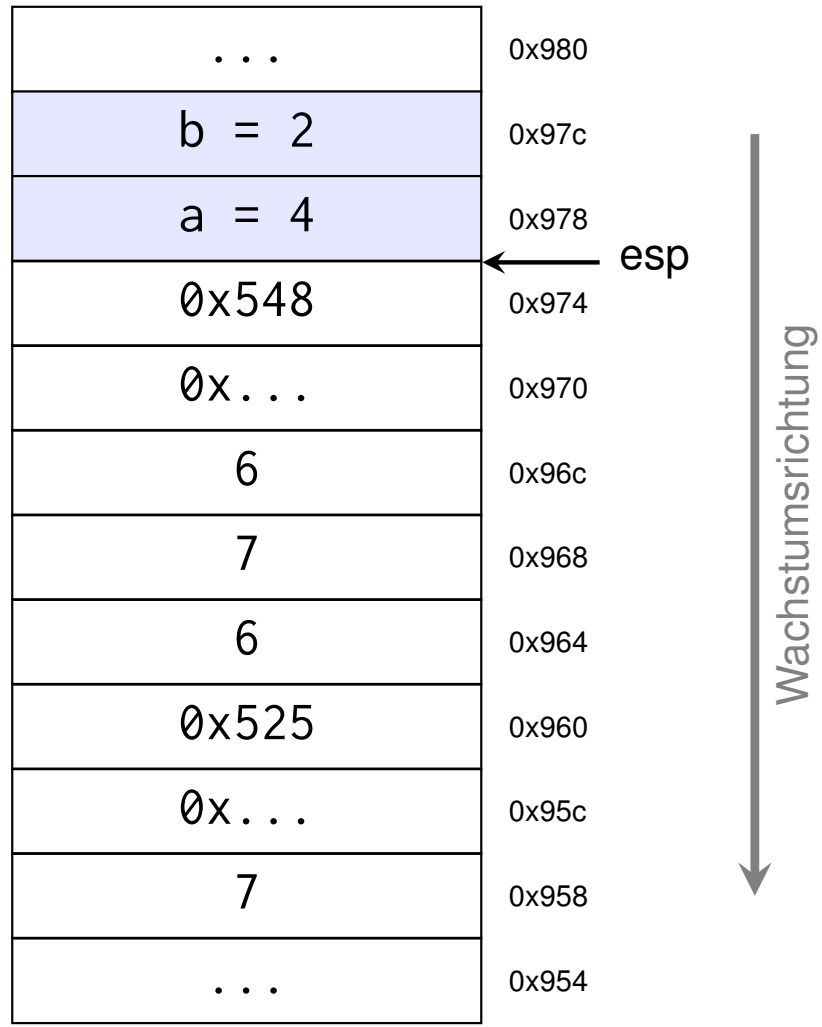
```
4fd:    push ebp
4fe:    mov ebp,esp
500:    lea ...
507:    or ...
50b:    sub esp,0x8
512:    mov edx,DWORD PTR [ebp+0x8]
515:    mov eax,DWORD PTR [ebp+0xc]
518:    add eax,edx
51a:    mov DWORD PTR [ebp-0x8],eax
51d:    push DWORD PTR [ebp-0x8]
520:    call 4e9 <g>
525:    add esp,0x4
528:    mov DWORD PTR [ebp-0x4],eax
52b:    mov eax,DWORD PTR [ebp-0x4]
52e:    leave
52f:    ret
```



Beispiel: Programmstapel

```
int main(void) {
  return f(4, 2);
}
```

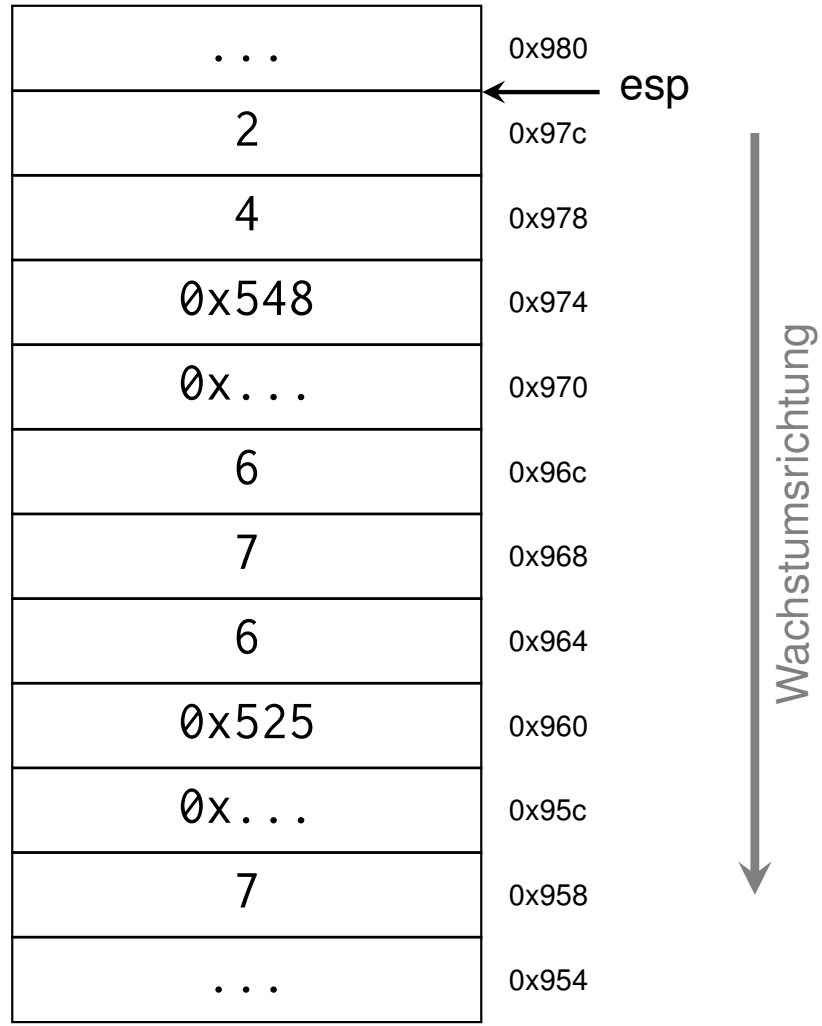
```
52a:    push ebp
52b:    mov ebp,esp
52d:    lea ...
534:    or ...
538:    lea ...
53f:    push 0x2
541:    push 0x4
543:    call 4fd <f>
548:    add esp,0x8
54b:    leave
54c:    ret
```



Beispiel: Programmstapel

```
int main(void) {
  return f(4, 2);
}
```

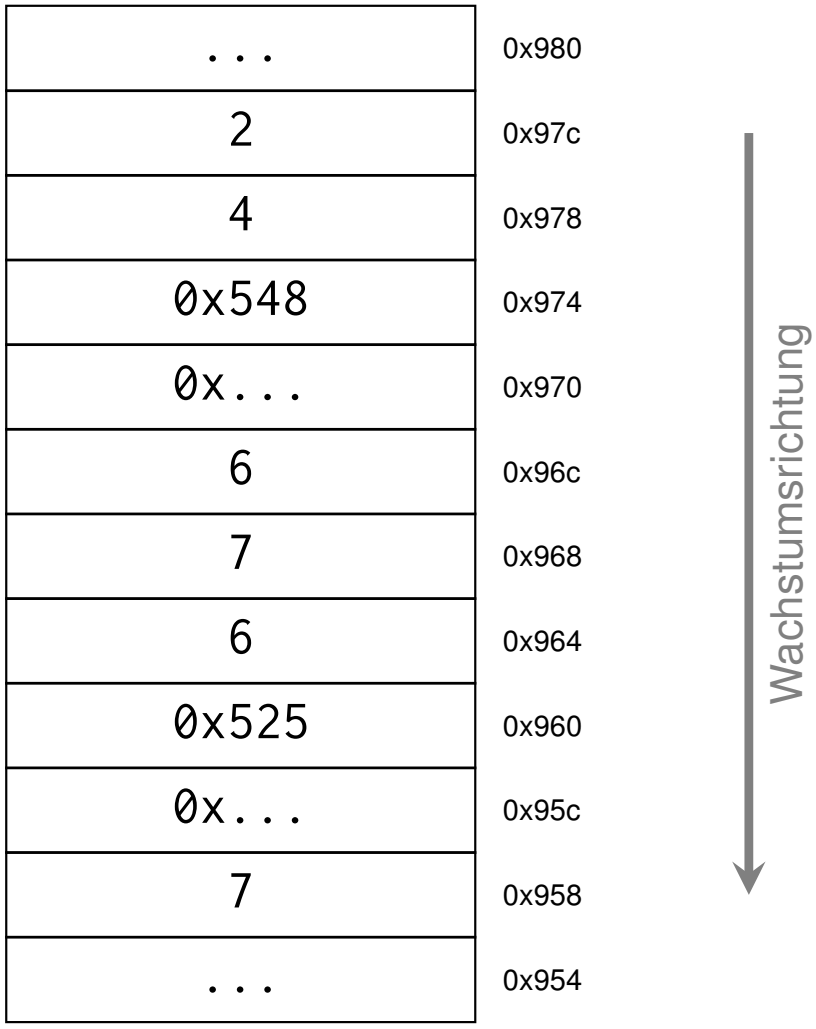
```
52a:    push ebp
52b:    mov ebp,esp
52d:    lea ...
534:    or ...
538:    lea ...
53f:    push 0x2
541:    push 0x4
543:    call 4fd <f>
548:    add esp,0x8
54b:    leave
54c:    ret
```



Beispiel: Programmstapel

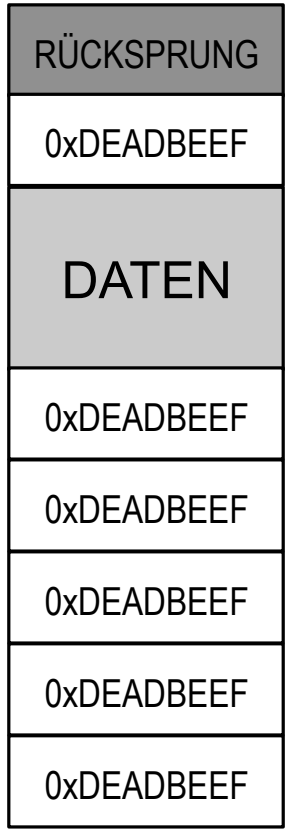
```
int main(void) {
  return f(4, 2);
}
```

```
52a:    push ebp
52b:    mov ebp,esp
52d:    lea ...
534:    or ...
538:    lea ...
53f:    push 0x2
541:    push 0x4
543:    call 4fd <f>
548:    add esp,0x8
54b:    leave
54c:    ret
```



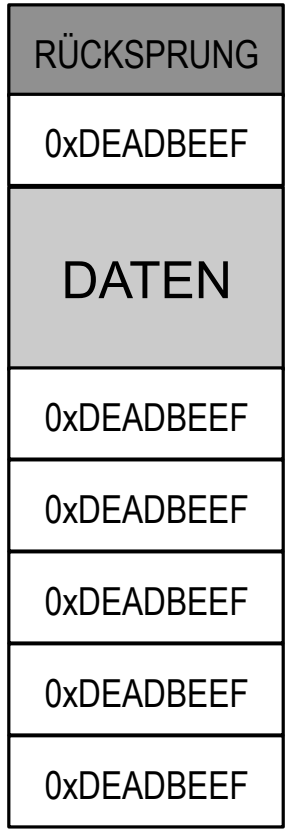
Dynamische Analyse des Stapelspeicherbedarfs

- Messung zur Laufzeit: Wasserstandsmessung
- Grundidee: Einfügen von **Stack Canaries**
- Explizite Verwaltung des Stapelspeichers notwendig
- pthread-Bibliothek ermöglicht Verwaltung
- Mögliche Canaries
  - Lesbare Bitmuster: `0xDEADBEEF`
  - Unwahrscheinliche Bitmuster: `0b101010101010...`
  - Kleinere Bitmuster $\rightsquigarrow$ größere Auflösung



Dynamische Analyse des Stapelspeicherbedarfs

- Messung zur Laufzeit: Wasserstandsmessung
- Grundidee: Einfügen von **Stack Canaries**
- Explizite Verwaltung des Stapelspeichers notwendig
- pthread-Bibliothek ermöglicht Verwaltung
- Mögliche Canaries
  - Lesbare Bitmuster: `0xDEADBEEF`
  - Unwahrscheinliche Bitmuster: `0b101010101010...`
  - Kleinere Bitmuster $\rightsquigarrow$ größere Auflösung
- ⚠ Keine allgemeingültigen Aussagen
  - Liefert nur den konkreten Bedarf der Messungen
  - Vorsichtige Aussagen über Worst-Case-Verhalten
- Einsatz zur dynamischen Fehlererkennung



pthread-Bibliothek

1. (Globalen) Stack anlegen:

```
1   static unsigned int g_data[DATA_SIZE];
```

2. Thread anlegen & starten:

```
1   pthread_t thread;
2   pthread_attr_t attr;
3   pthread_attr_init(&attr);
4   pthread_attr_setstack(&attr, &g_stack, STACK_SIZE);
5   // worker function: void *run(void *param)
6   int status = pthread_create(&thread, &attr, run, NULL);
7   if (status != 0) { ... // handle error
```

3. Auf Thread warten:

```
1   pthread_join(thread, &ret);
```

pthread Stack