Ein Programm mit `fork`

Emilio Pielsticker

Programm mit fork

```
void hit() { printf("%d, ", x++); x += 3; }
```

```
int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b \r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

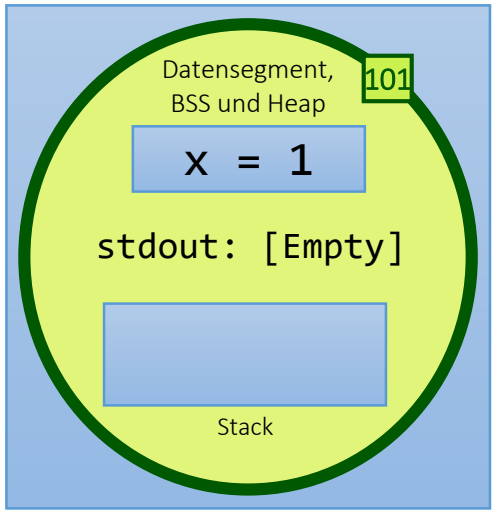
Programm mit fork

```
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC

→ Im nächsten Schritt:

Speicher



Terminalkonsole

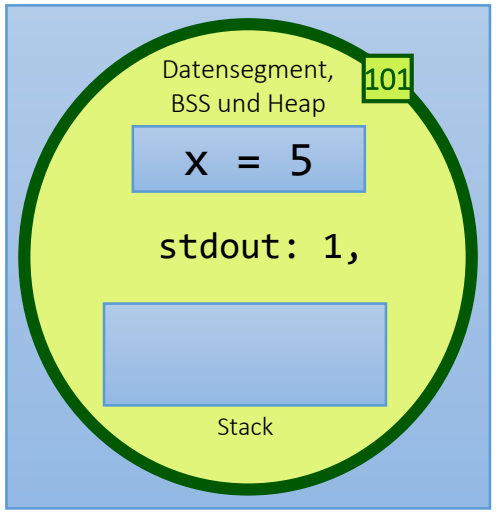
Programm mit fork

```
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

→ Im nächsten Schritt:

Speicher



Terminalkonsole

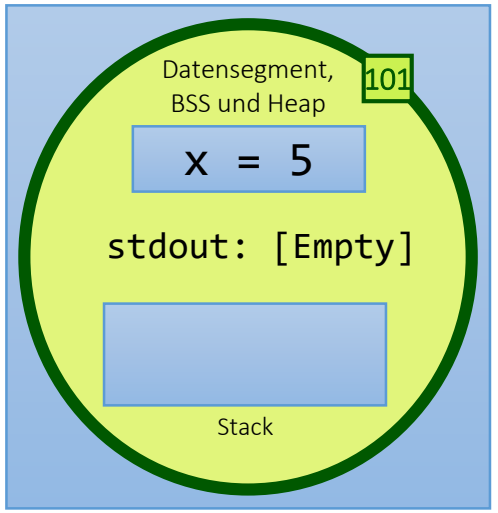
Programm mit fork

```
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

→ Im nächsten Schritt: `pid_t pid = fork();`

Speicher



Terminalkonsole

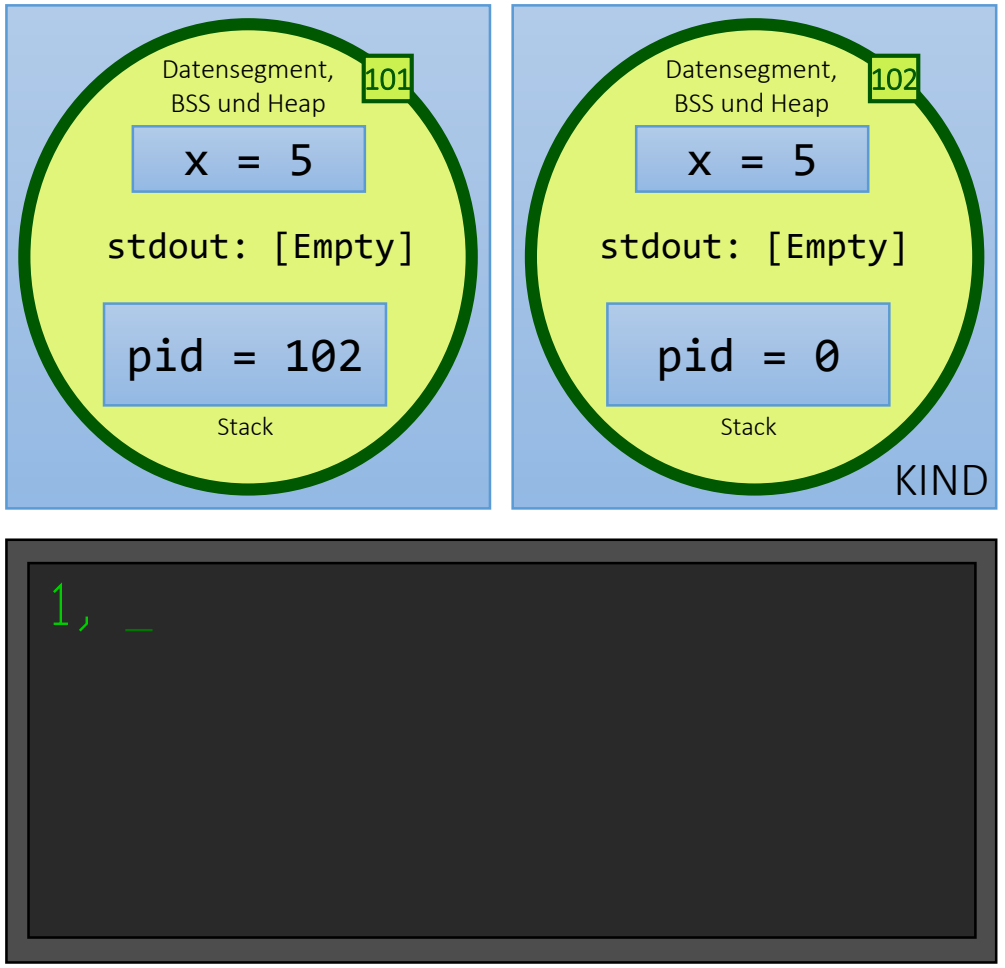
Programm mit fork

```c
void hit() { printf("%d, ", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b \r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

PC (101)

PC (102)



Programm mit fork

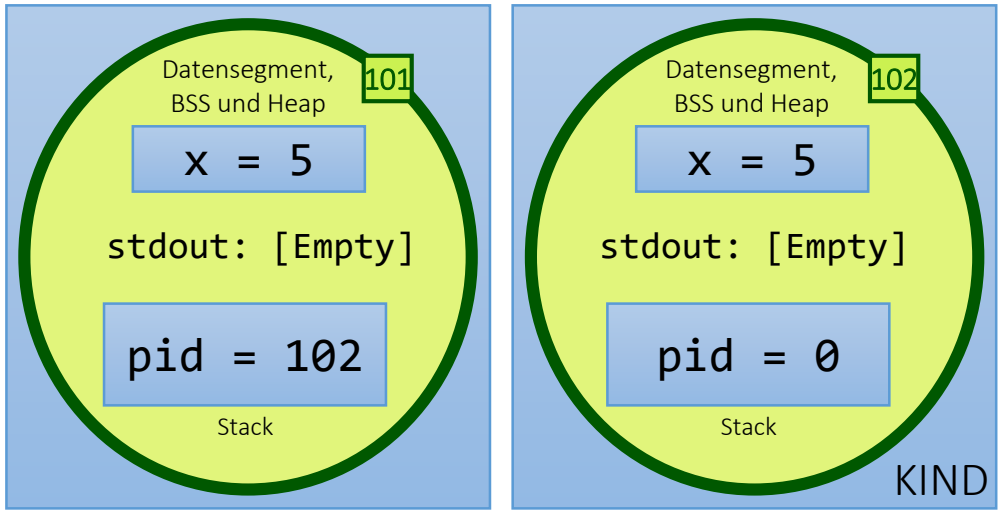
```
void hit() { printf("%d, ", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b \r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

Hier warten!

PC (102)

Speicher



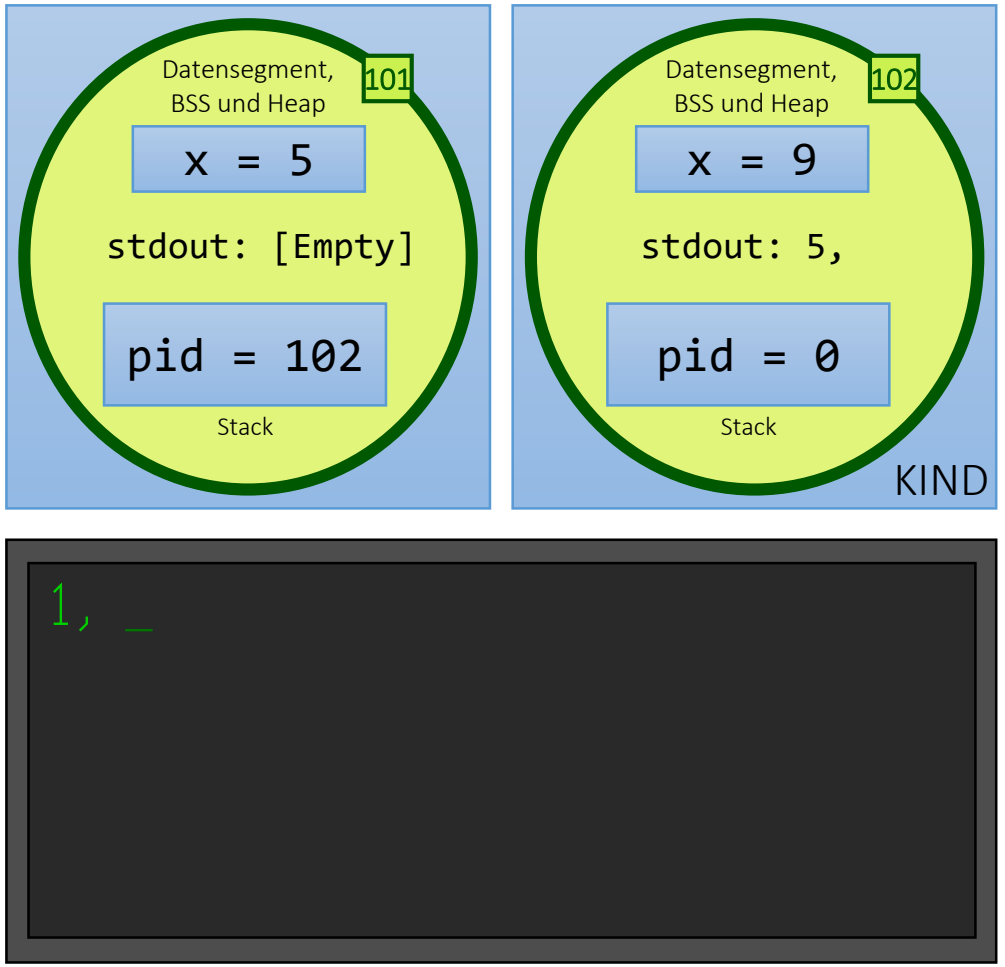
Terminalkonsole

Programm mit fork

```
void hit() { printf("%d, ", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b \r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

PC (102)

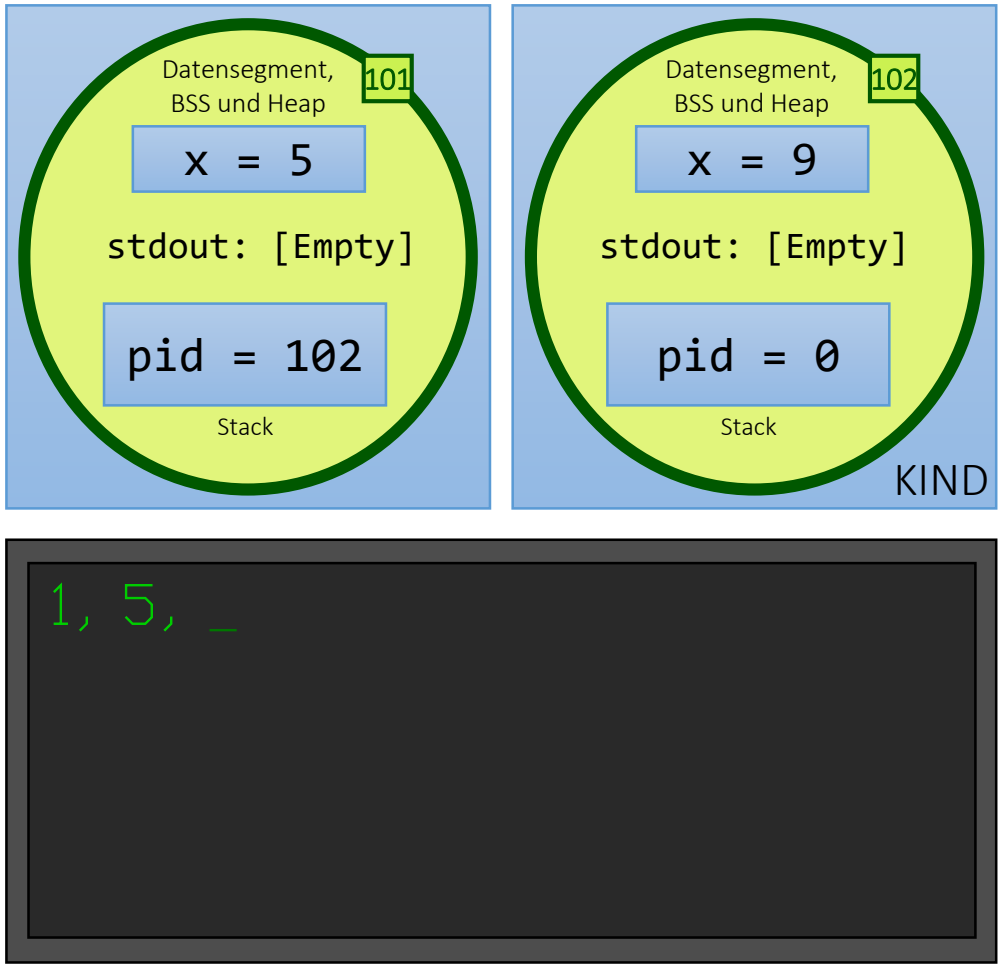


Programm mit fork

```c
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

PC (102)



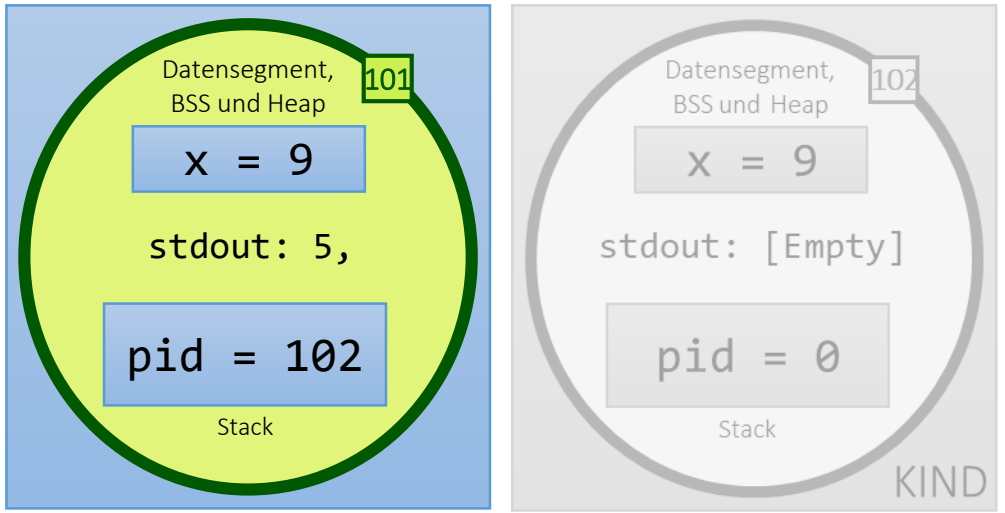
Programm mit fork

```
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

PC (102)

Speicher



Terminalkonsole

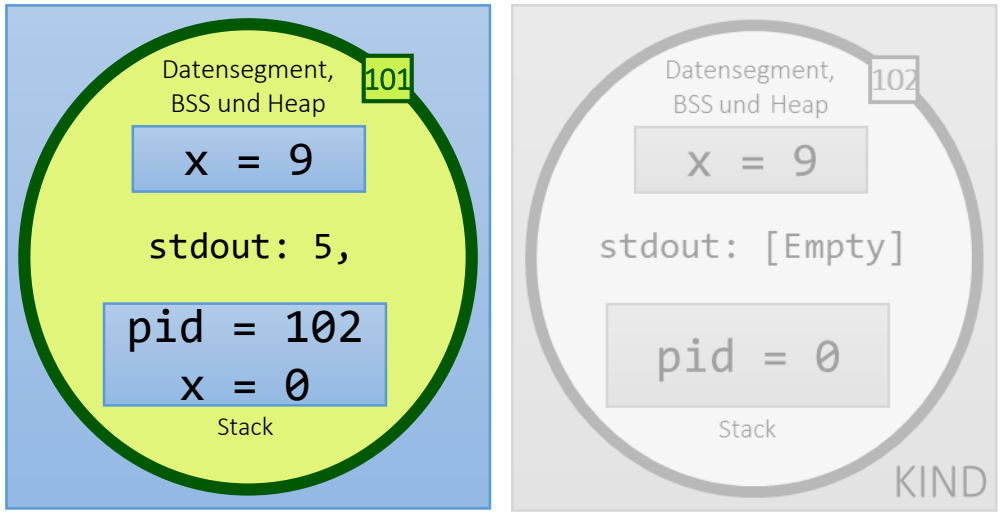
Programm mit fork

```
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

PC (102)

Speicher



Terminalkonsole

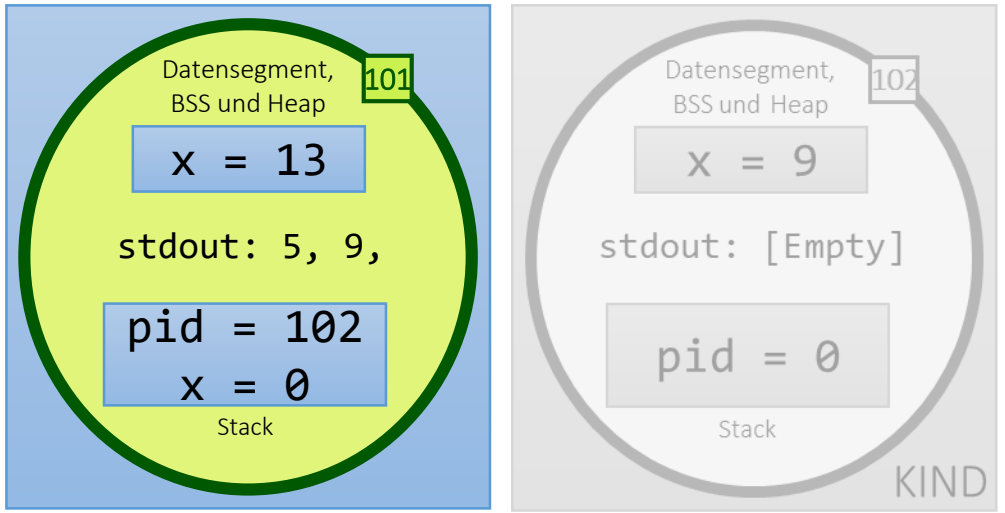
Programm mit fork

```c
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

PC (102)

Speicher

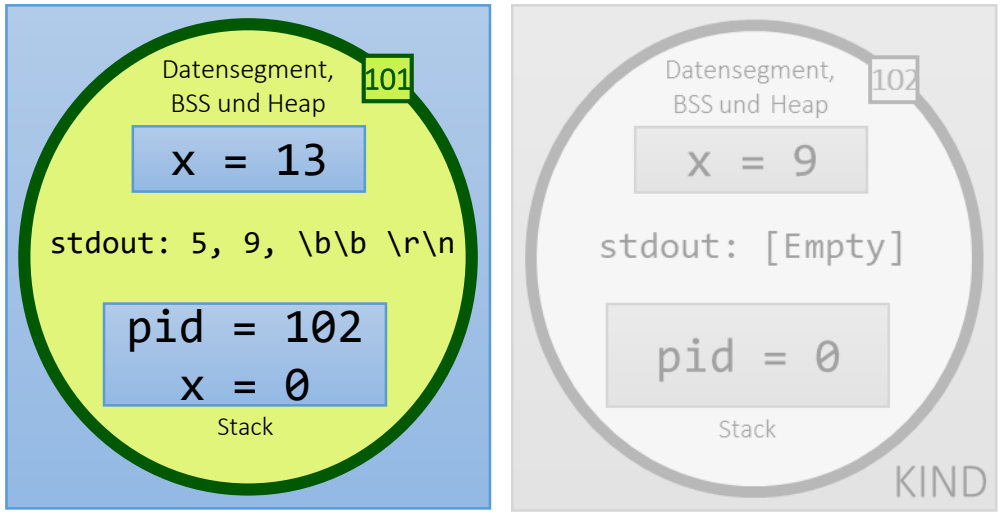


Terminalkonsole

Programm mit fork

```
void hit() { printf("%d, ", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
PC (101) printf("\b\b \r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
PC (102) return 0;
}
```

Speicher

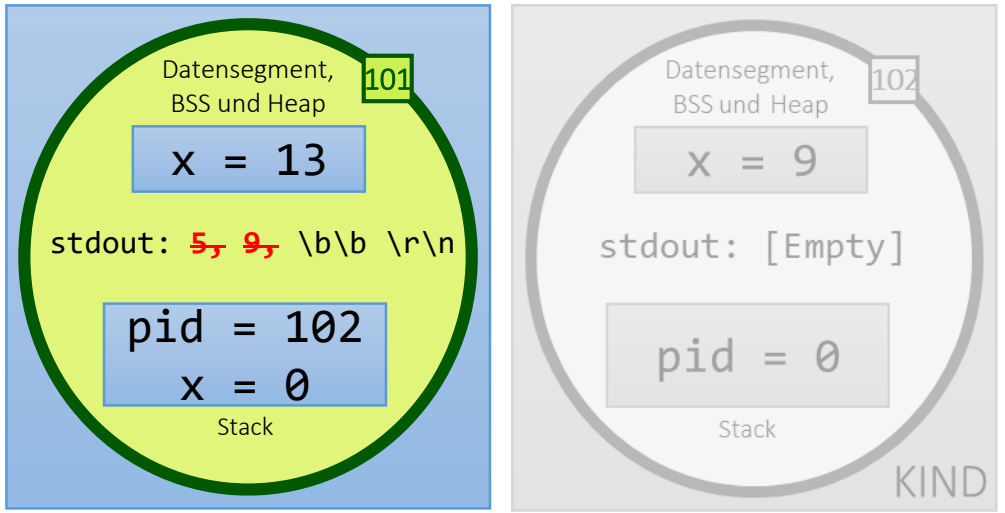


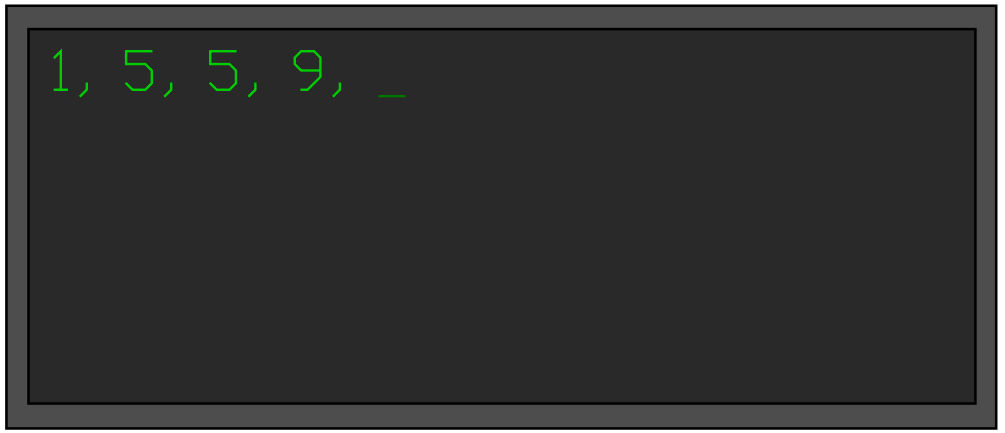
Terminalkonsole

Programm mit fork

```
void hit() { printf("%d, ", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
PC (101) printf("\b\b \r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
PC (102) return 0;
}
```

Speicher



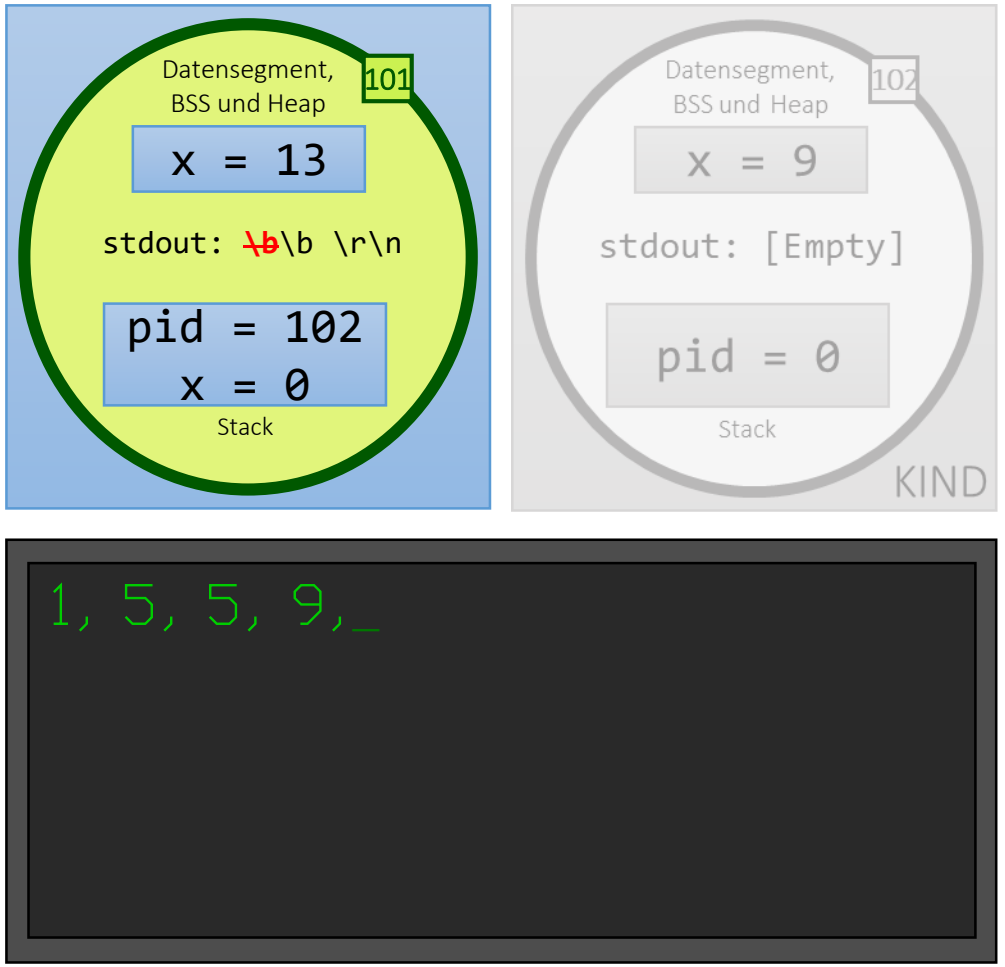


Terminalkonsole

Programm mit fork

```
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
PC (101) printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
PC (102) return 0;
}
```

Speicher

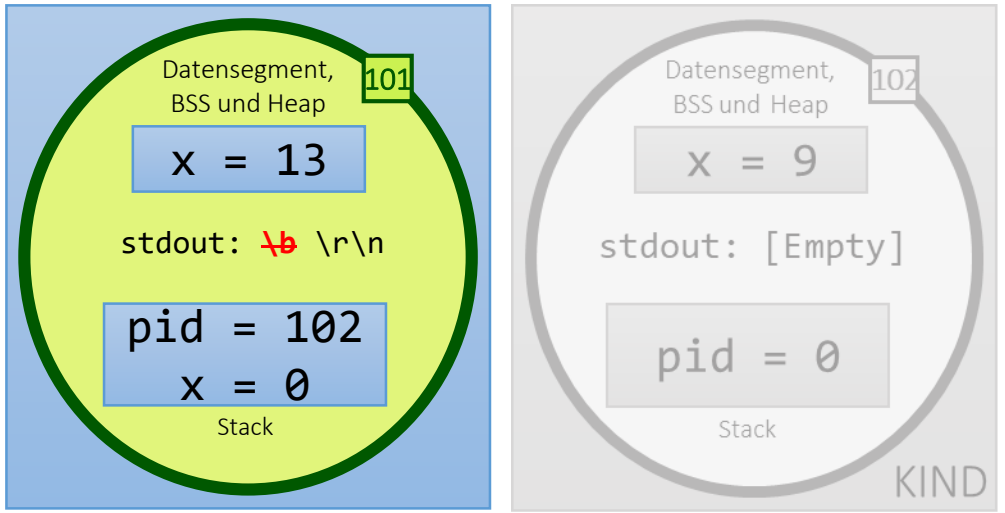


Terminalkonsole

Programm mit fork

```
void hit() { printf("%d, ", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
PC (101) printf("\b\b \r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
PC (102) return 0;
}
```

Speicher

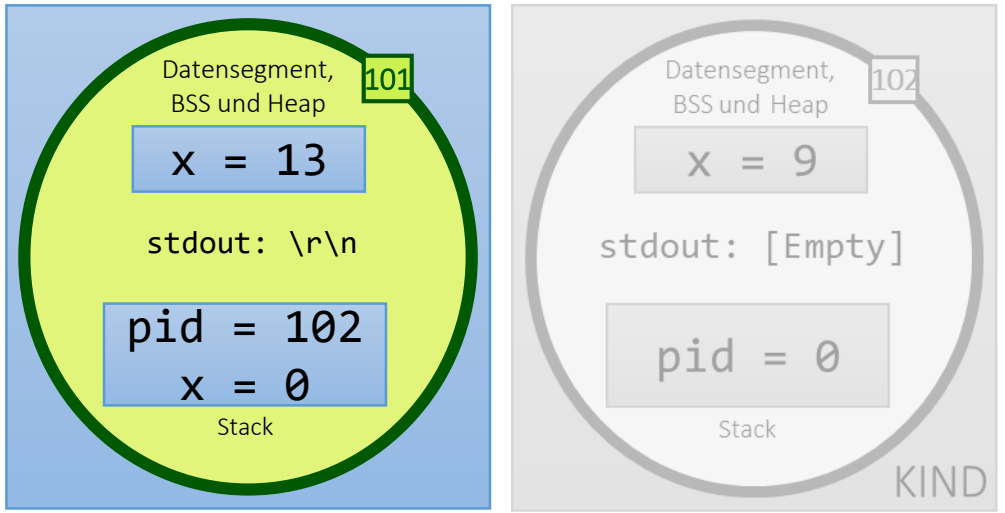


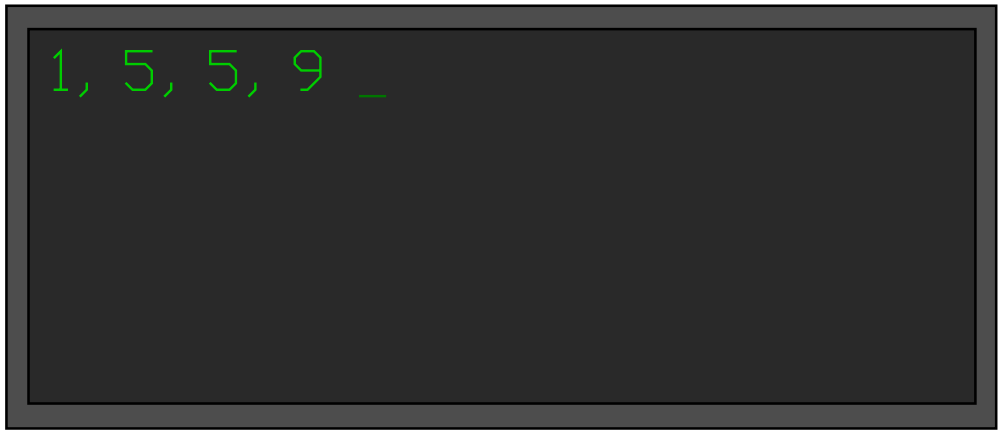
Terminalkonsole

Programm mit fork

```
void hit() { printf("%d, ", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
PC (101) printf("\b\b \r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
PC (102) return 0;
}
```

Speicher



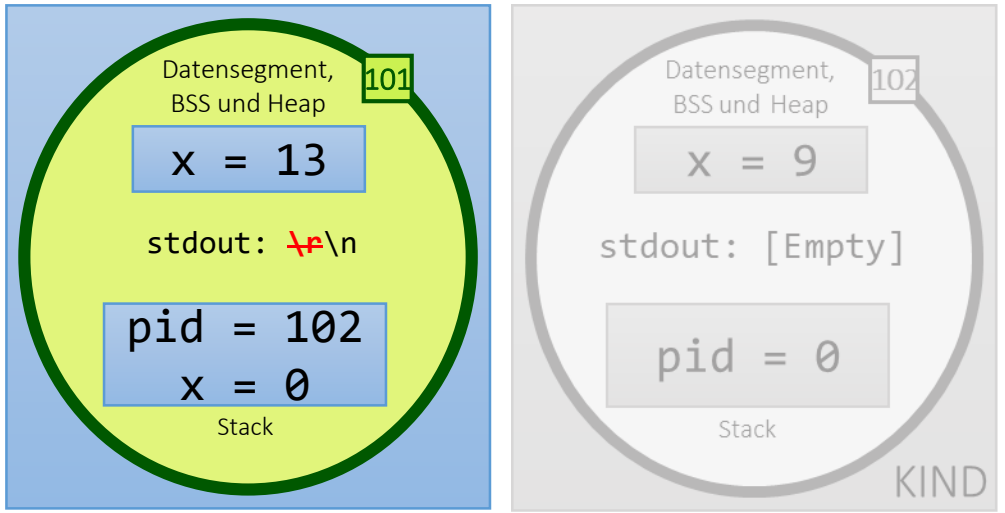


Terminalkonsole

Programm mit fork

```
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
PC (101) printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
PC (102) return 0;
}
```

Speicher

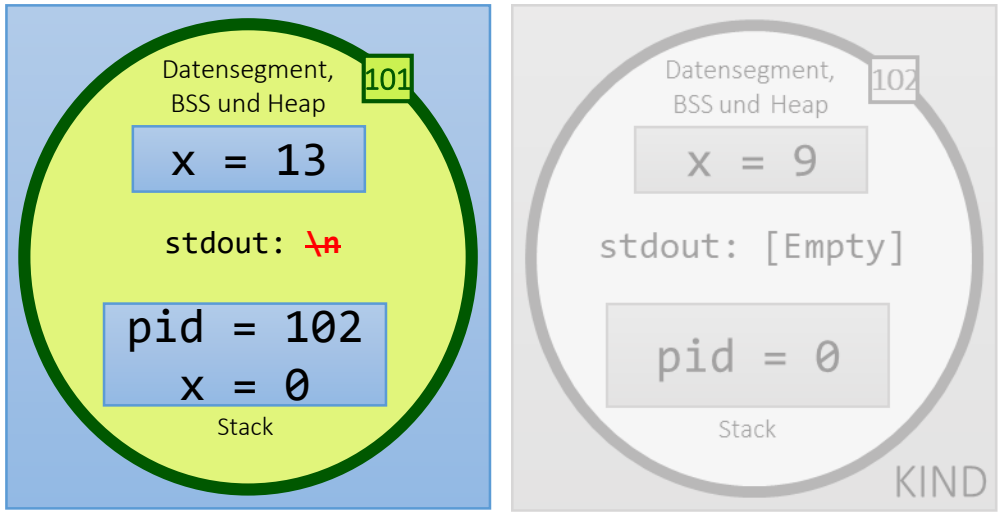


Terminalkonsole

Programm mit fork

```
void hit() { printf("%d, ", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
PC (101) printf("\b\b \r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
PC (102) return 0;
}
```

Speicher



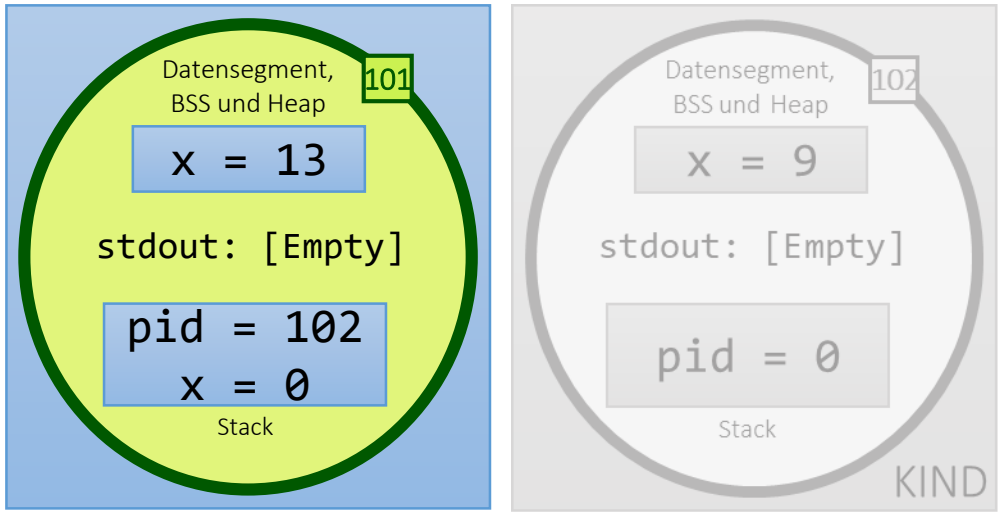
Terminalkonsole

Programm mit fork

```
void hit() { printf("%d,␣", x++); x += 3; }

int x = 1;
int main() {
    hit();
    fflush(NULL);
    pid_t pid = fork();
    if (pid > 0) {
        wait(NULL);
        hit();
        int x = 0;
        hit();
        printf("\b\b␣\r\n");
    } else if (pid == 0) {
        hit();
    } else {
        hit();
        perror("Fehler!\n");
        exit(-1);
    }
    return 0;
}
```

PC (101)

Speicher



Terminalkonsole