Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

Programmieren in C

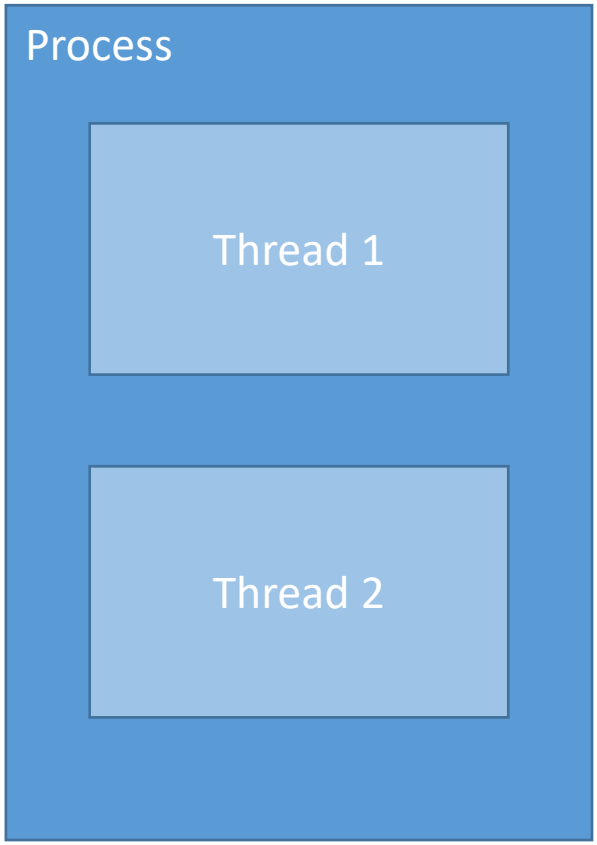
Dynamic Memory Management

Dr. Adrian Koller, Büro E307, adrian.koller@hslu.ch

Processes & Thread

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Prozesse sind aktiv ausgeführte Programme
- Jeder Prozess enthält mindestens 1 Thread, aber auch mehrere Threads
- Threads können einfach auf gemeinsamen Speicher zugreifen
  → Für Prozesse kompliziert



Stack vs. Heap Memory (vs. Static)

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Beide sind in RAM → flüchtig.
  Werden nur bei Exekution bereitgestellt.

- Jedes Programm (process) hat mindestens 1 Stack (1 pro Thread…)
  - Funktionen und deren Variablen
  → genau so viel wie notwendig, zusammenhängend!

- Jeder process hat 1 heap, gemeinsam für alle Threads
  «grosser Haufen», frei verfügbar.
  → bis zum Rechner-Limit

- Static: globale Variablen innerhalb eines Prozesses

Stack vs. Heap Beispiel

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

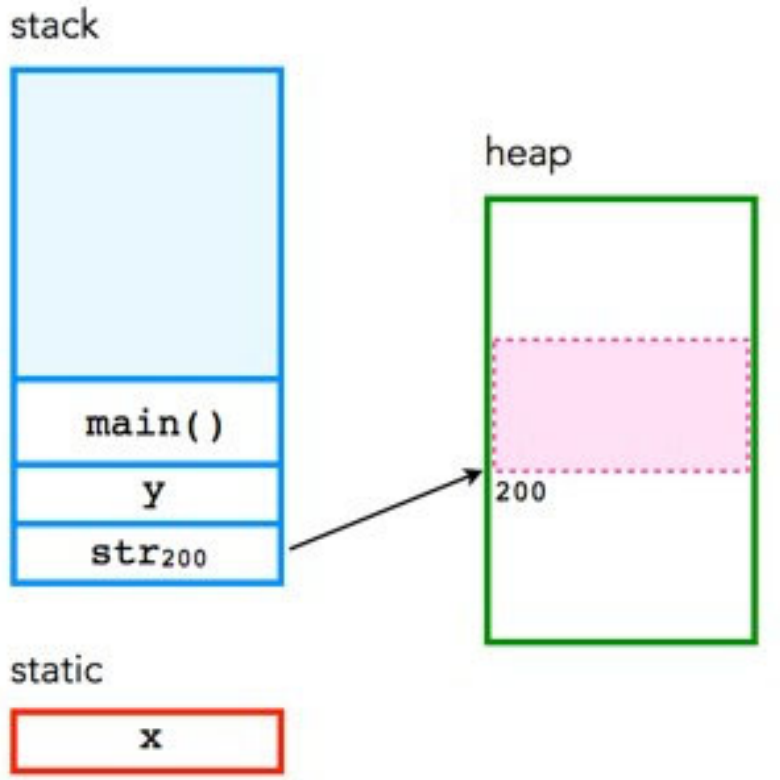
```c
#include <stdio.h>
#include <stdlib.h>

int x;

int main(void)
{
    int y;
    char *str;

    y = 4;
    printf("stack memory: %d\n", y);

    str = malloc(100*sizeof(char));
    str[0] = 'm';
    printf("heap memory: %c\n", str[0]);
    free(str);
    return 0;
}
```



https://craftofcoding.wordpress.com/2015/12/07/memory-in-c-the-stack-the-heap-and-static/

Mehr zum Stack

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- «Last-in-First-out»:

  --> Startet beim main()
  → Variablen werden in dieser Reihenfolge angelegt

  --> Wird eine Funktion aufgerufen, wird der Stack um die Funktion und deren lokalen Variablen erweitert.

  →Endet die Funktion, wird der für die Funktion benötigte Speicher wieder freigegeben → «Scope»!!

- Stack wird automatisch gemanaged

malloc und free

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- mit malloc («memory allocate») wird dynamisch Speicher reserviert.
  → die Funktion returniert einen Pointer zum bereitgestellten Speicher.
  → falls «NULL», malloc nicht erfolgreich!

- mit «free» wird der Speicher wieder freigegeben