CHALMERS
UNIVERSITY OF TECHNOLOGY

Machine-Oriented Programming

C-Programming

Pedro Moura Trancoso
ppedro@chalmers.se

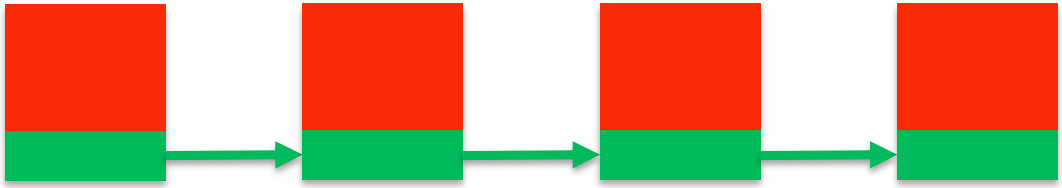
Previous Lecture

- ???
- ???
- ???

Objectives

- More pointers:
  - Linked lists, doubly linked lists, linked lists of linked lists, trees, graphs
  - Example: Text Editor – Simple version and Optimizations
- More structs:
  - Storage, packing, padding
- More strings:
  - Efficient storage for strings of variable length
- Just more!
  - Command line parameters
- List of Don’t s
  - Out-of-bound access
  - Access to local static variables
  - Return address

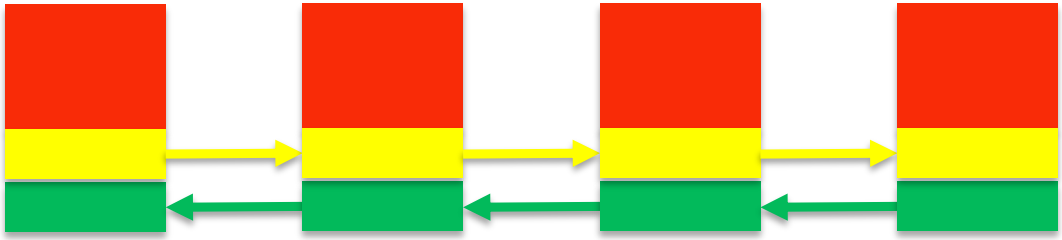
Linked Lists



```
typedef struct node_t {
  char firstName[100];
  char lastName[100];
  int  idnumber;
  struct node_t* next;
} node_t;
```

```
int findInList(int num) {…}
int insertInList(node_t* node) {}
int removeFromList(int num) {}


int findInList(node_t* head, int num) {…}
node_t* insertInList(node_t* head, node_t* node) {}
node_t* removeFromList(node_t* head, int num) {}
```

Doubly Linked Lists

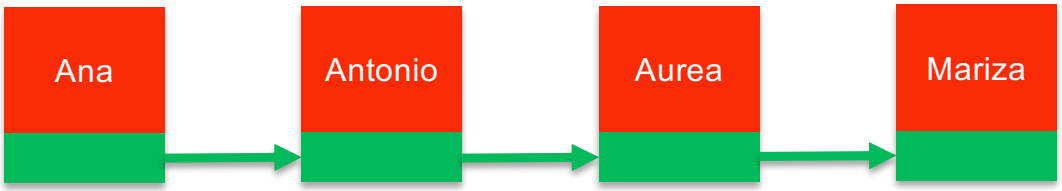


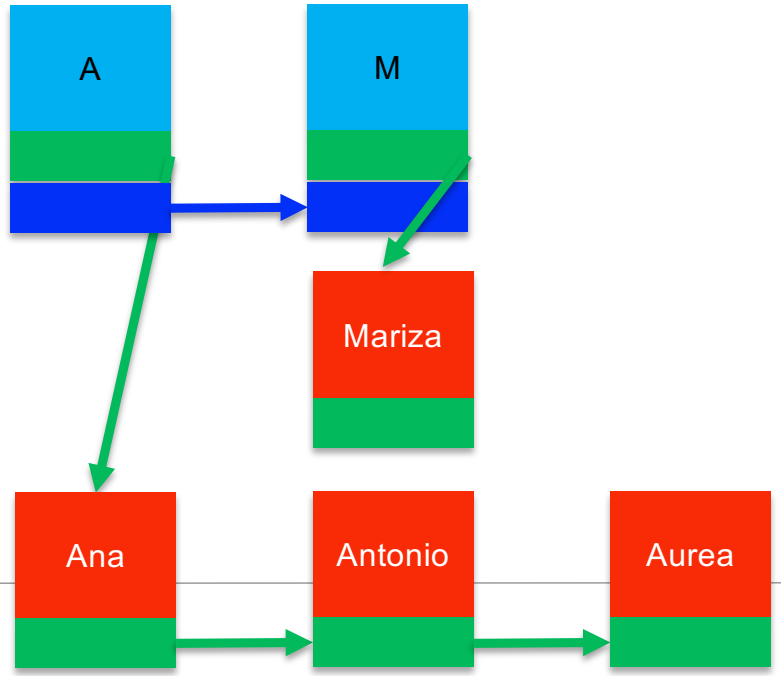
```
typedef struct node_t {
  char firstName[100];
  char lastName[100];
  int  idnumber;
  struct node_t* next;
  struct node_t* prev;
} node_t;
```

```
int findInList(int num) {…}
int insertInList(node_t* node) {}
int removeFromList(int num) {}
```

Linked Lists of Linked LIsts

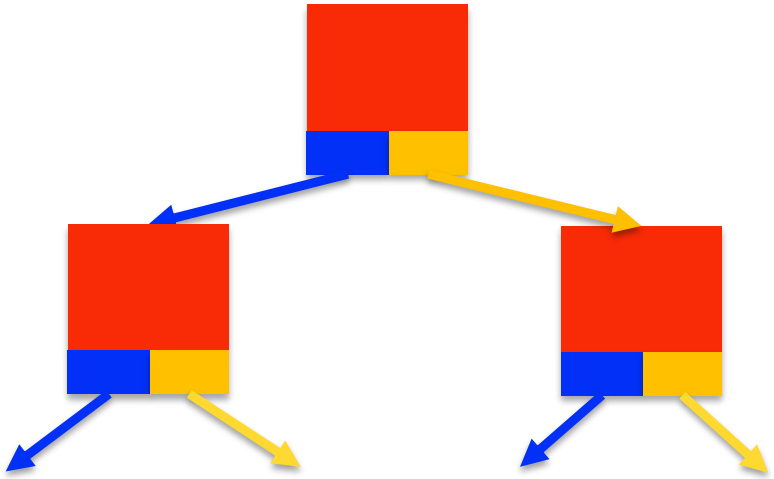
Problem: How to organize a structure for handling different people (e.g. in a Database for the students of MOP!)





Trees and Graphs

More efficient structures



Example: Simple Text Editor

Structs: Packing and Padding

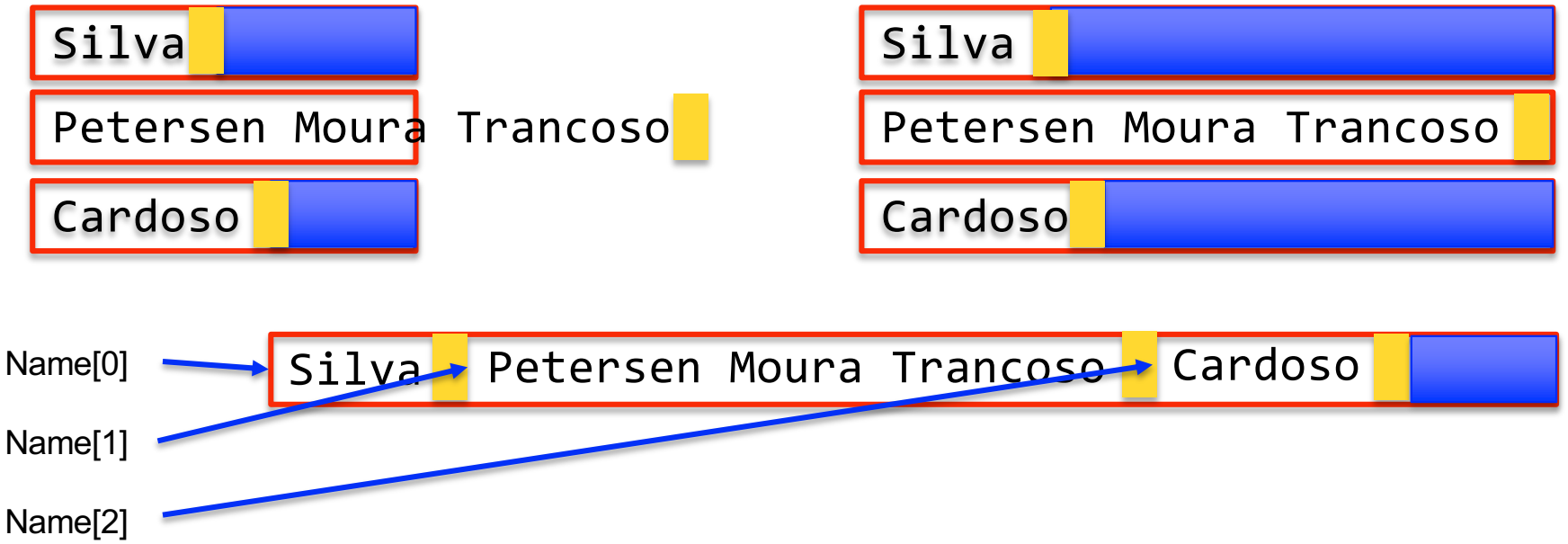
```
#include <stdio.h>

typedef struct info_t {
  char name1[3];
  int  number1;
  char name2[2];
  int  number2;
} info_t;

int main() {
  info_t x[2];

  printf("sizeof(struct)=%d\n", sizeof(info_t));
  printf("sizeof(elements)=%d\n",
sizeof(x[0].name1)+sizeof(x[0].number1)+sizeof(x[0].name2)+sizeof(x[0].number2));
}
```

__attribute__((__packed__))

16 & 13!

Strings: Efficient Storage of Variable Length

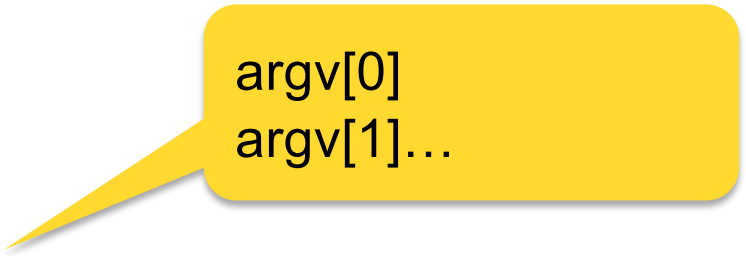
Problem: How to efficiently store name? How long should each string be?



Command Line Parameters

Important to provide inputs to the program

```
#include <stdio.h>

int main(int argc, char *argv[]) {


}
```



DONTs: Out-of-bound Access

```
#include <stdio.h>

int array1[] = { 0,1,2,3,4,5,6,7,8,9 };
int array2[] = { 10,11,12,13,14,15,16,17,18,19 };
int array3[] = { 20,21,22,23,24,25,26,27,28,29 };


int main(int argc, char *argv[]) {
   for(int i=0; i<30; i++)
      printf( "i=%d array1[i]=%d\n", i, array1[i]);

}
```

```
i=0 array1[i]=0
i=1 array1[i]=1
i=2 array1[i]=2
i=3 array1[i]=3
i=4 array1[i]=4
i=5 array1[i]=5
i=6 array1[i]=6
i=7 array1[i]=7
i=8 array1[i]=8
i=9 array1[i]=9
i=10 array1[i]=0
i=11 array1[i]=0
i=12 array1[i]=10
i=13 array1[i]=11
i=14 array1[i]=12
i=15 array1[i]=13
i=16 array1[i]=14
i=17 array1[i]=15
i=18 array1[i]=16
i=19 array1[i]=17
i=20 array1[i]=18
i=21 array1[i]=19
i=22 array1[i]=0
i=23 array1[i]=0
i=24 array1[i]=20
i=25 array1[i]=21
i=26 array1[i]=22
i=27 array1[i]=23
i=28 array1[i]=24
i=29 array1[i]=25
```

DONTs: Access to static local variables

```
#include <stdio.h>

void foo() {
  static int a = 0;
  a++;
  printf("In foo: a=%d\n", a);
}

int main(int argc, char *argv[]) {
   foo();
   foo();
   foo();
   foo();
}
```

```
#include <stdio.h>

int* foo() {
  static int a = 0;
  a++;
  printf("In foo: a=%d\n", a);
  return &a;
}

int main(int argc, char *argv[]) {
   int* x;
   x = foo();
   *x = 0;
   x = foo();
   *x = 0;
   x = foo();
   *x = 0;
   x = foo();
}
```

DONTs: Stack change return address

```
#include <stdio.h>

void foo() {
  unsigned int a = 1;
  unsigned int b = 2;
  unsigned int *x = &b;
  printf("In foo: *x=%u\n", *x); x++;
  printf("In foo: *x=%u\n", *x); x++;
  printf("In foo: *x=%u\n", *x);
  *x+=4;
}

int main(int argc, char *argv[]) {
   int x = 0;
   foo();
   x+=10;
   x+=10;
   printf("x = %d\n", x);
}
```

DONTs: Don’t write code any more!!!

iPad 06:24 72%

New A.I. application can write its own code - Futurity

4 hours ago

Computer scientists have created a deep-learning, software-coding application that can help human programmers navigate the growing multitude of often-undocumented application programming interfaces, or APIs.

Designing applications that can program computers is a long-sought grail of the branch of computer science called artificial intelligence (AI). The new application, called Bayou, came out of an initiative aimed at extracting knowledge from online source code repositories like GitHub. Users can try it out at askbayou.com.

> “The days when a programmer could write code from scratch are long gone.”

bayou http://askbayou.com

Department of Computer Science,
Rice University, USA.

What would you like to see in a MOP C course?