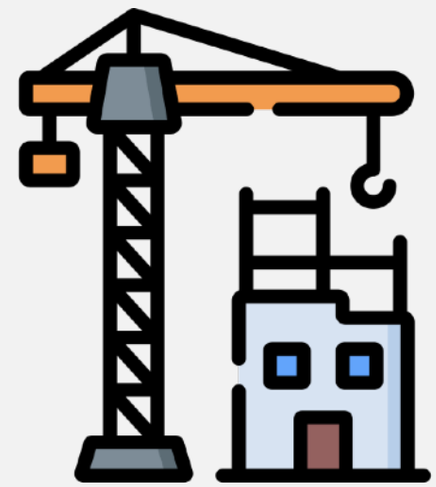
COP 3223H: Introduction to C Programming

Fall 2023

University of Central Florida

Dr. Kevin Moran

Week 13- Class II: Dynamic Arrays & Structs



Administrivia

- LPA 2 due today.
- SPA5 and LPA 3 will be released on Monday.
- Mid-Semester Feedback Survey posted,
  - Will count as Quiz 3 - due Monday
- Office Hours today if you need it for LPA2.
- No class on November 22nd - Happy Thanksgiving! 🦃

Today's Agenda

1. Demo of Dynamic Arrays
2. Intro to Dynamic Structs

Review

Dynamic Array Example

```
int size;

printf ("How many elements would you like: "); scanf(“%d", &size);

int *array = (int *) malloc(size * sizeof(int));

for(int x = 0; x < size; ++x){

    printf("Enter a value: ");
    scanf(“%d", &array[x]) ;
}

for(int x = 0; x < size; ++x){

    printf(“array[%d] = %d\n”, x, array[x]) ;
    free (array) ;
    array = NULL;
}
```

2-D Dynamic Array Example

```
int row, col;

printf("Enter the number of rows and columns you would like. Please separate with a
space.\n");
printf("Enter here: ");

scanf("%d%d", &row, &col);

int *arr = (int *)malloc(row * col * sizeof(int));

int i,j;
for (i = 0; i < row; i++)
    for(j = 0; j < col; j++)
        *(arr + i*col +j)= i + j;

for (i = 0; i < row; i++){
    for(j=0; j < col; j++){
        printf("&d ", *(arr + i*col + j));
    }
    printf("\n") ;
}
  free(arr) ;
  arr = NULL;
```

2-D Dynamic Array Example

```
int r = 3, c = 4, i, j, count;

int** arr = (int**) malloc(r * sizeof(int*));

for (i = 0; i < 1; i++)
    arr[i] = (int*)malloc(c * sizeof(int));

// Note that arr|i][j] is same as *(*(arr+i)+j)
count = 0;
```

```
for (i = 0; i < 1; i++)
    for(j = 0; j < c; j++)
        arr[i][j] = ++count; // OR *(*(arr+i)+j) = ++count

for (i =0; i < 1; i++)
    for(j = 0; j < c; j++)
        printf("%d ", arr[i][j]);

for(int i=0; i< r; i++)
    free(arr[i]);

free (arr);
```

Demo

Dynamic Structs

Indirect Component Selection Operator

- The indirect component selection operator is the character sequence -> placed between a pointer variable and a component name create a reference that follows the pointer to a structure and selects the component.

- While first one is valid to use, it can be a bit cumbersome to use, which is why C provides the indirect operator.

```
book_t *book_ptr = &mybook;

char title[MAX] = (*book_ptr).title;
char title2[MAX] = book_ptr->title;
```

Dynamic Memory and Structs

- Similar syntax to dealing with primitive data types.

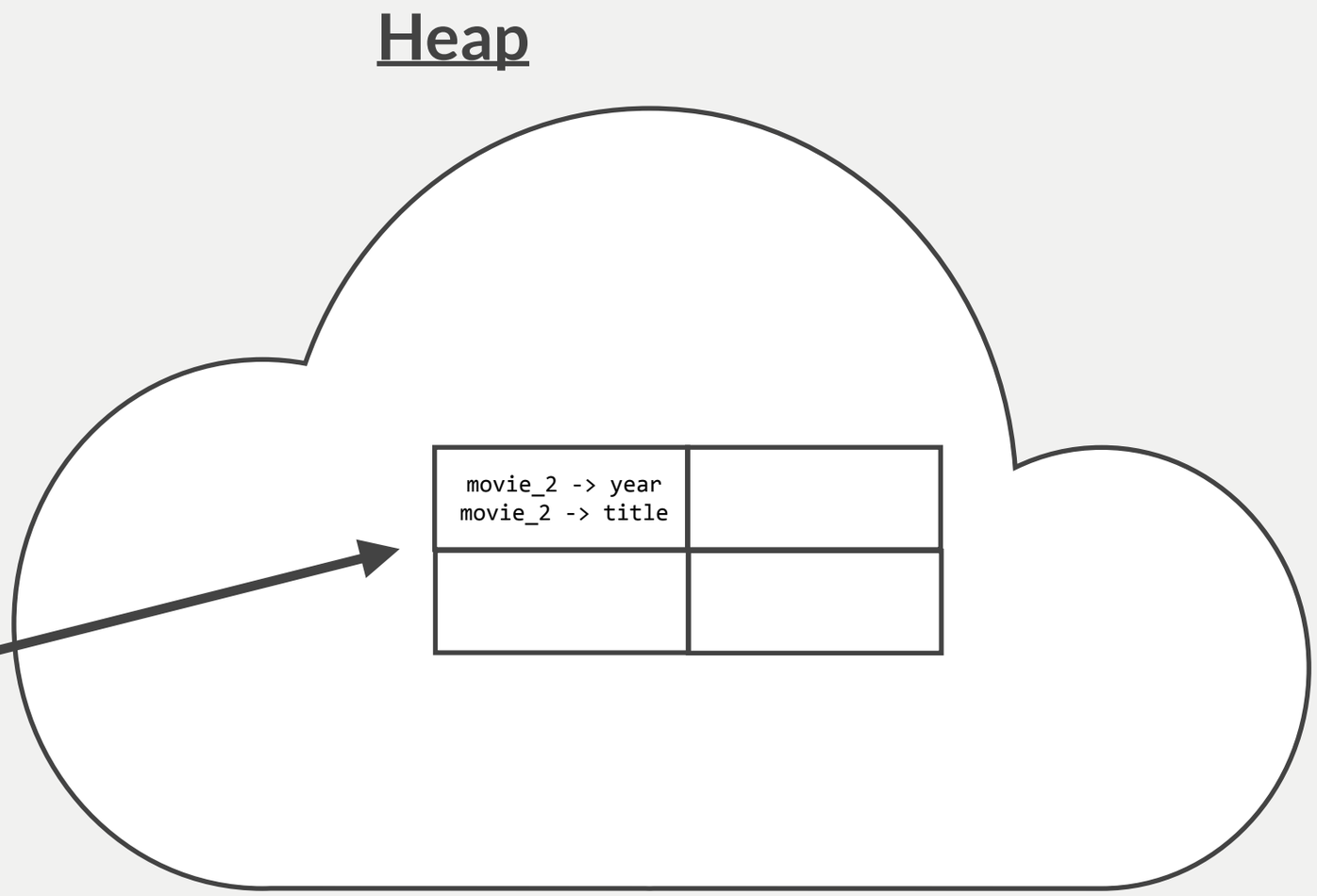
```
typedef struct{
    int year;
    char title[30];
}movie_t;
```

```
movie_t movie1; // declared in stack space
movie_t *movie2 = (movie_t *) malloc(sizeof(movie_t)); // declared in heap space

free(movie2);
```

Visualizing Dynamic Struct Allocation

Stack Space	
AA9	
AA8	
AA7	
AA6	
AA5	
AA4	
AA3	*movie2
AA2	
AA1	
AA0	



Populating the Components of a Dynamic Struct

- Similar syntax to dealing with primitive data types.

```
typedef struct{
    int year;
    char title[30];
}movie_t;
```

```
movie_t *movie2 = (movie_t *) malloc(sizeof(movie_t));
strcpy(movie2->title, "Avatar");
movie2->year = 2022;

printf("%s\n", movie2->title) ;
printf("%d\n", movie2->year);

free (movie2) ;
```

Dynamic Struct Components

```
typedef struct{
    int year;
    char * title;
    char * author;
}book_t;
```

Acknowledgements

Slides adapted from Dr. Andrew Steinberg's COP 3223H course