

UPR: Structured Data Types (struct)

Jan Gaura

<https://github.com/geordi/upr-course>

Outline

- **Administrative**
- struct
- Final notes

What did I do?

Administrative

On SUN's years:

*Kicked butt, had fun, didn't cheat,
loved our customers, changed computing forever.*

Scott McNealy, SUN Microsystems

Administrative

On SUN's years:

*Kicked butt, had fun, didn't cheat,
loved our customers, changed computing forever.*

Scott McNealy, SUN Microsystems

Q & A Session

- Last week, we made a Q & A session for you

Q & A Session

- Last week, we made a Q & A session for you
 - three (3) people came

Q & A Session #2

- So...

Q & A Session #2

- So...
- ... we'll hold another one NEXT week again

Q & A Session #2

- So...
- ... we'll hold another one NEXT week again
- Tuesday 22. 10. 2019 in EB213

GitHub - The Course's Page

geordi / upr-course

Watch

5

Star

2

Fork

0

<> Code

Issues0

Pull requests0

Projects0

Wiki

Security

Insights

Settings

UPR course at VSB

Edit

Manage topics

51 commits

2 branches

0 releases

2 contributors

Branch: masterNew pull request

Create new fileUpload filesFind fileClone or download


Kobzol	Add geordi to Slack list	Latest commit 54e8fc8 2 days ago
assets	First VSC run animation	27 days ago
exercise_handouts/ex_01	Update ex_01.md	24 days ago
faq	Update README.md	5 days ago
lectures	Add basic hello world and guessing game code.	last month
tasks	Add geordi to Slack list	2 days ago
.gitignore	Initial commit	3 months ago
README.md	Add practice programming tasks	8 days ago
exercises.md	Update exercises.md	12 days ago
readings.md	Readings and Exercises updated	13 days ago




README.md

Introduction to Programming (Úvod do programování)

You can find all information regarding the Introduction to programming (UPR) course taught at the [Department of Computer Science, VŠB - TU Ostrava](#).

GitHub - Easy to Check What's Going On


 [geordi](#) / [upr-course](#)



 5  2  0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


Branch: **master** ▾



Commits on Oct 8, 2019

Add geordi to Slack list
 Kobzol committed 2 days ago


Verified  54e8fc8 



Commits on Oct 5, 2019


Update README.md
 geordi committed 5 days ago



Verified  4407706 


Commits on Oct 2, 2019



Add build essential to VSCode FAQ
 Kobzol committed 8 days ago


Verified  1cc5fa3 



Add notice about installing CMake
 Kobzol committed 8 days ago

Verified  220523a 

Add practice programming tasks
 Kobzol committed 8 days ago

 1734275 

Add VS Code useful shortcuts
 Kobzol committed 8 days ago

 d8c2b00 

Outline

- Administrative
- **struct**
- Final notes

Structured Data Types - A Motivation

- Working with all arrays and lengths of arrays separately...

Structured Data Types - A Motivation

- Working with all arrays and lengths of arrays separately...
- ... can be a mess

Structured Data Types - A Motivation

- Working with all arrays and lengths of arrays separately...
- ... can be a mess

```
01 int stats( int * arr, int arr_len ) {  
02     int max = arr[ 0 ];  
03  
04     for ( int i = 1; i < arr_len, i++ ) {  
05         if ( arr[ i ] > max ) {  
06             max = arr[ i ];  
07         }  
08     }  
09  
10     return max;  
11 }
```


Structured Data Types - A Motivation

- Working with all arrays and lengths of arrays separately...
- ... can be a mess

```
01 int stats( int * arr, int arr_len ) {  
02     int max = arr[ 0 ];  
03  
04     for ( int i = 1; i < arr_len, i++ ) {  
05         if ( arr[ i ] > max ) {  
06             max = arr[ i ];  
07         }  
08     }  
09  
10     return max;  
11 }
```

← May result in Segmentation Fault

Structured Data Types - A Motivation

- Working with all arrays and lengths of arrays separately...
- ... can be a mess

```
01 int stats( int * arr, int arr_len ) {  
02     int max = arr[ 0 ];  
03  
04     for ( int i = 1; i < arr_len, i++ ) {  
05         if ( arr[ i ] > max ) {  
06             max = arr[ i ];  
07         }  
08     }  
09  
10     return max;  
11 }
```

← May result in Segmentation Fault

```
01 void stats( int * arr1, int arr1_len, int * arr2, int arr2_len, int *max1, int *max2 ) {  
02     *max1 = arr1[ 0 ];  
03     *max2 = arr2[ 0 ];  
04  
05     for ( int i = 0; i < arr1_len, i++ ) {  
06         ...  
07  
08     for ( int i = 0; i < arr2_len, i++ ) {  
09         ...  
10  
11 }
```

Structured Data Types - A Motivation

- What if...

Structured Data Types - A Motivation

- What if an array would also hold it's length?

Structured Data Types - A Motivation

- What if an array would also hold it's length?

```
01 int stats( int * arr, int arr_len ) {  
02     int max = arr[ 0 ];  
03  
04     for ( int i = 1; i < arr_len, i++ ) {  
05         if ( arr[ i ] > max ) {  
06             max = arr[ i ];  
07         }  
08     }  
09  
10     return max;  
11 }
```

Structured Data Types - A Motivation

- What if an array would also hold it's length?

```
01 int stats( int * arr, int arr_len ) {  
02     int max = arr[ 0 ];  
03  
04     for ( int i = 1; i < arr_len, i++ ) {  
05         if ( arr[ i ] > max ) {  
06             max = arr[ i ];  
07         }  
08     }  
09  
10     return max;  
11 }
```

```
01 int stats( Array * arr ) {  
02     int max = arr.items[ 0 ];  
03  
04     for ( int i = 1; i < arr.len, i++ ) {  
05         if ( arr.items[ i ] > max ) {  
06             max = arr.items[ i ];  
07         }  
08     }  
09  
10     return max;  
11 }
```

Is there any solution?

Sure :-)

The aswer is: struct

Struct Defintion

- To concentrate things together...

Struct Defintion

- To concentrate things together...
- We use structures (struct) in C language

Struct Defintion

- To concentrate things together...
- We use structures (struct) in C language
- How can we define a struct?

Struct Defintion

- To concentrate things together...
- We use structures (struct) in C language
- How can we define a struct?

```
struct Array {  
    int * items;  
    int len;  
};
```

Struct Defintion

Struct Defintion

```
struct Array {  
    int * items;  
    int len;  
};
```

Struct Defintion

```
struct Array {  
    int * items;  
    int len;  
};
```

```
int main() {  
    struct Array * a = array_new( 10 );  
  
    printf( "Array len= %d\n", a->len );  
  
    return 0;  
}
```


Struct Defintion

```
struct Array {  
    int * items;  
    int len;  
};
```

```
int main() {  
    struct Array * a = array_new( 10 );  
    printf( "Array len= %d\n", a->len );  
    return 0;  
}
```

```
struct Array * array_new( const int len ) {  
    struct Array * a = NULL;  
  
    a = (struct Array *)malloc( sizeof( struct Array ) );  
    // memory check  
  
    a->items = (int *)malloc ( len * sizeof( int ) );  
    // memory check  
    a->len = len;  
  
    return a;  
}
```

Struct Usage Simplification

- Ohhh, if I could only get rid of the struct keyword here and there

Struct Usage Simplification

- Ohhh, if I could only get rid of the struct keyword here and there
- Well... you can

Struct Usage Simplification

- Ohhh, if I could only get rid of the struct keyword here and there
- Well... you can

```
1 struct Array {  
2     int * items;  
3     int len;  
4 };  
5  
6 typedef struct Array Array;
```

Struct Usage Simplification

- Ohhh, if I could only get rid of the struct keyword here and there
- Well... you can

```
1 struct Array {  
2     int * items;  
3     int len;  
4 };  
5  
6 typedef struct Array Array;
```

Struct Usage Simplification

- Ohhh, if I could only get rid of the struct keyword here and there
- Well... you can

```
1 struct Array {  
2     int * items;  
3     int len;  
4 };  
5  
6 typedef struct Array Array;
```

```
int main() {  
    Array * a = array_new( 10 );  
  
    printf( "Array len= %d\n", a->len );  
  
    return 0;  
}
```

Struct Usage Simplification

- Ohhh, if I could only get rid of the struct keyword here and there
- Well... you can

```
1 struct Array {  
2     int * items;  
3     int len;  
4 };  
5  
6 typedef struct Array Array;
```

```
int main() {  
    Array * a = array_new( 10 );  
  
    printf( "Array len= %d\n", a->len );  
  
    return 0;  
}
```

```
Array * array_new( const int len ) {  
    Array * a = NULL;  
  
    a = (Array *)malloc( sizeof( Array ) );  
    // memory check  
  
    a->items = (int *)malloc ( len * sizeof( int ) );  
    // memory check  
    a->len = len;  
  
    return a;  
}
```