



Univerza v Ljubljani

Fakulteta *za računalništvo
in informatiko*

Programski jezik C

Tomaž Dobravec

Določila spremenljivk

Storage classes in C

Storage Specifier	Storage	Initial value	Scope	Life
auto	stack	Garbage	Within block	End of block
extern	Data segment	Zero	global Multiple files	Till end of program
static	Data segment	Zero	Within block	Till end of program
register	CPU Register	Garbage	Within block	End of block

Določilo `volatile`

- Vrednost spremenljivke se lahko spremeni tudi zunaj kode
- Prevajalnik take spremenljivke ne sme optimizirati

```
volatile int    *status_reg;  
volatile char  *recv_reg;  
int get_next_char() {  
    while (*status_reg == 0);  
    return *recv_reg;  
}  
status_reg = ...  
recv_reg   = ...  
int ch = get_next_char();
```

Določilo `const`

- `const ...` konstanta
- Prevajalnik ne dovoli spreminjati vrednosti `const` spremenljivke

```
int          a=10;  
const int b=15;  
a = 11;      // OK  
b = 12;      // ni dovoljeno!
```

```
const int *i  = &a;  
i  = &b;      // OK  
*i = 13;      // ni dovoljeno!
```

Večdimenzionalne tabele

Prostor za **večdimenzionalno tabelo** je določen statično pred začetkom izvajanja programa, podatki v njej so zapisani v “row-major order”

```
int anArray[3][5] ={  
    { 1,  2,  3,  4,  5 },    // 1. vrstica  
    { 6,  7,  8,  9, 10 },    // 2. vrstica  
    { 11, 12, 13, 14, 15 }    // 3. vrstica  
};
```

[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
[2][0]	[2][1]	[2][2]	[2][3]	[2][4]

Večdimenzionalne tabele

Pri deklaraciji je **le ena** dimenzija lahko lahko “dinamična” (ostale so potrebne za izračun indeksa)

```
void i(int a[][5]) {  
    printf("%d\n", a[1][1]);  
}  
  
int main(int argc, char **args) {  
    int a[][5] = {1,2,3,4,5, 6, 7, 8, 9};  
    i(a);  
}
```

Tabela tabel

Tabelo tabel deklarirano z dvojno zvezdico (kazalec na kazalec)

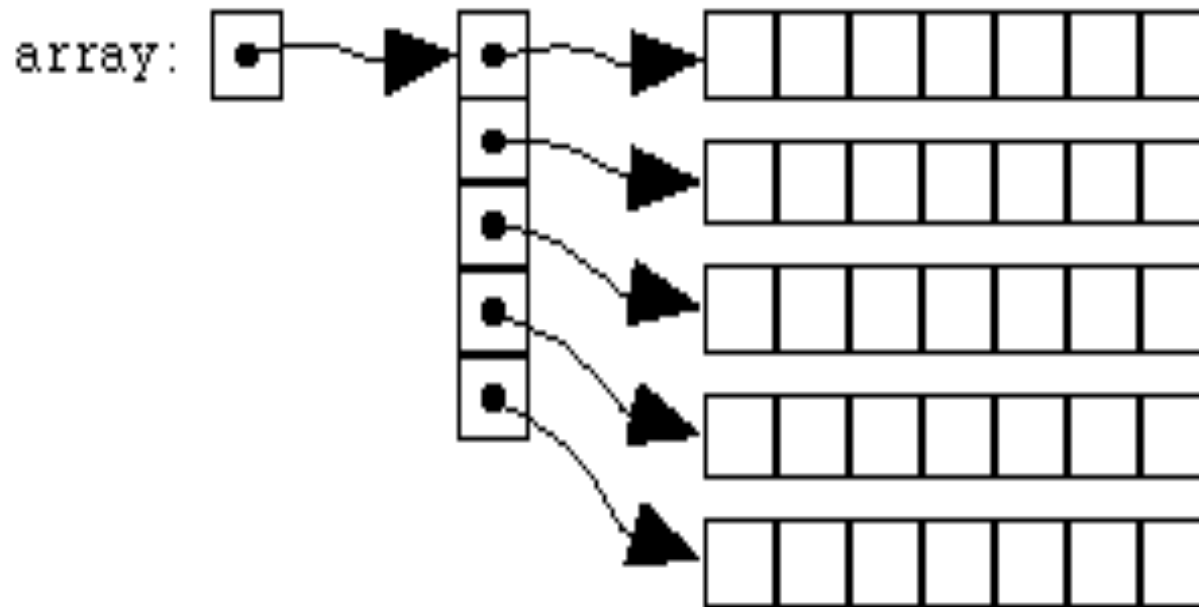
```
int** array; // tabela tabel
```

Rezervacija pomnilnika:

- rezervirati je treba prostor za tabelo kazalcev (`array`)
- rezervirati je treba prostor za vsako podtabelo (`array[i]`)

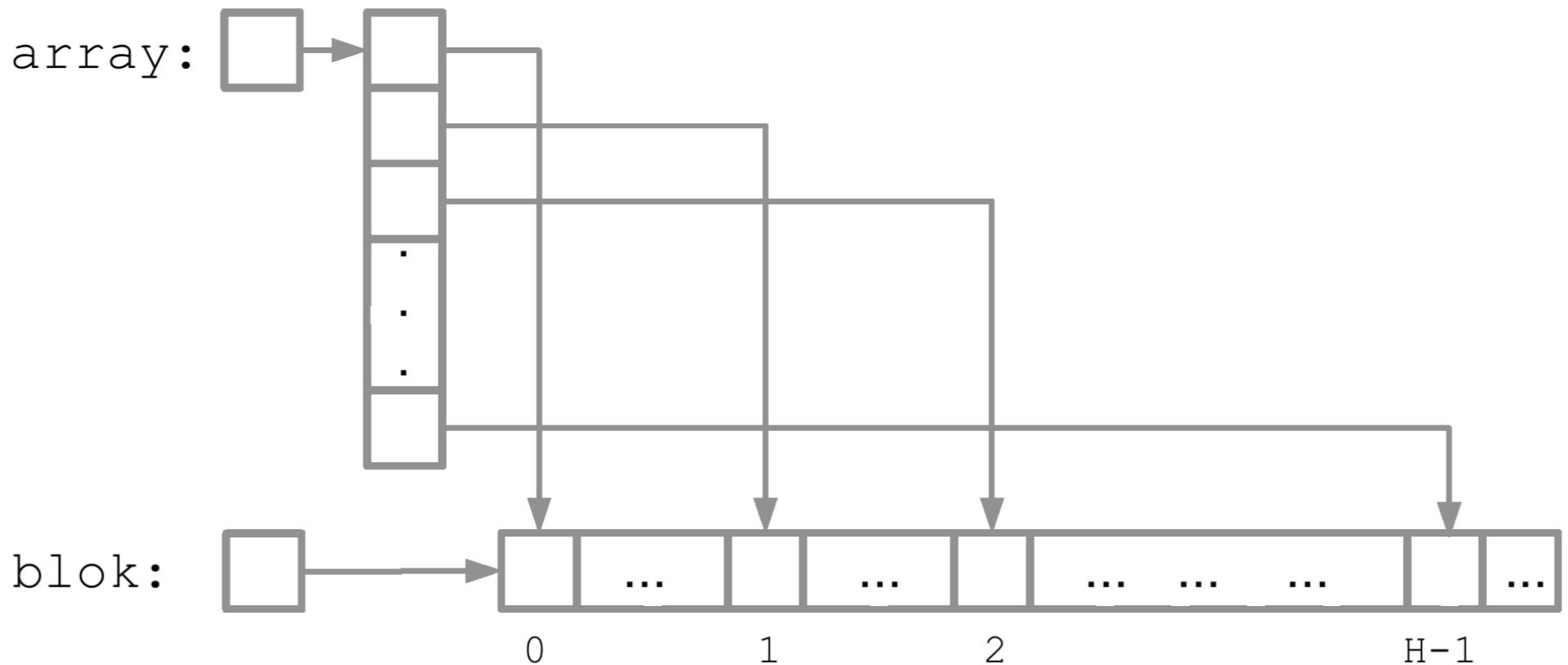
Tabela tabel

```
int** array; // tabela tabel  
array = (int**) malloc(5 * sizeof(int *));  
for(int i=0; i<5; i++)  
    array[i] = (int*) malloc(7 * sizeof(int));
```



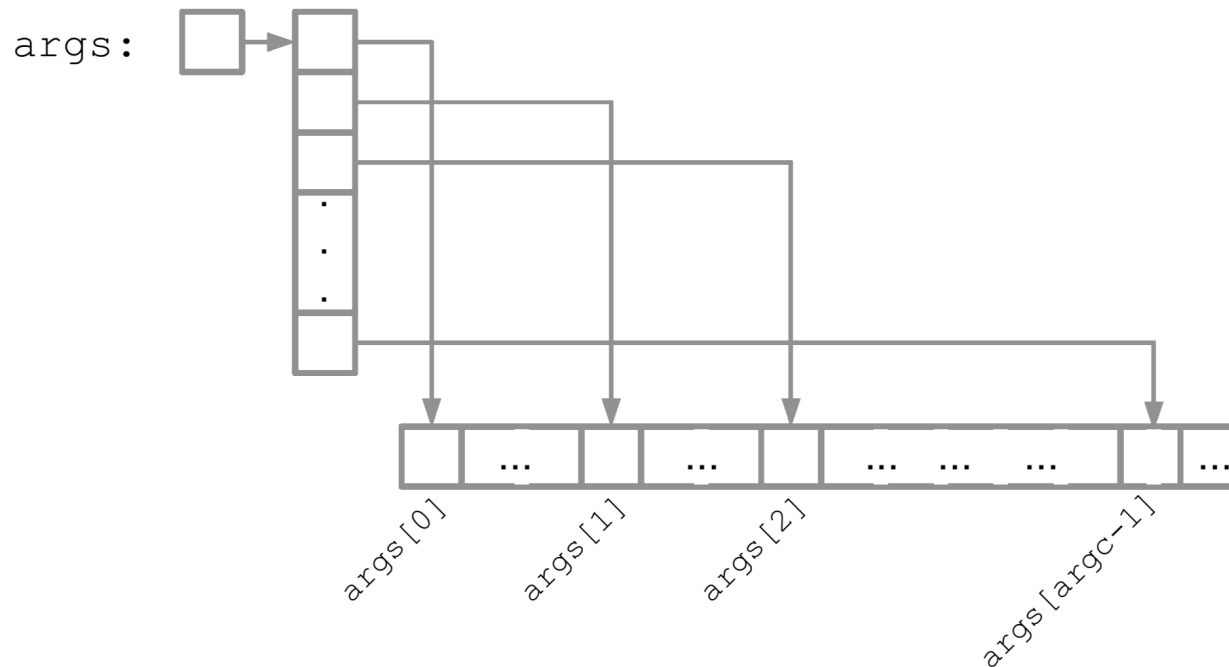
Neprekinjena dvodimenzionalna dinamična tabela

```
char *blok = (char *) malloc(H * W * sizeof(char));  
char **array = (char**) malloc(H * sizeof(char *));  
for(i=0; i<H; i++)  
    array[i] = blok + W * i * sizeof(char);
```



Argumenti programa

```
int main(int argc, char **args) {  
    for(int i=0; i<argc; i++) {  
        printf("%s\n", *args);  
        args++;  
    }  
}
```



Okoljske spremenljivke

- Izpis vseh okoljskih spremenljivk (odvisno od sistema)

```
int main(int argc, char **args, char **env) {  
    while (*env != NULL) {  
        printf("%s\n", *env);  
        env++;  
    }  
}
```

- Izpis vrednosti ene spremenljivke:

```
getenv()
```

Delo z binarnimi datotekami

- uporaba načina “b” pri klicu `fopen`,
- branje “znak-po-znak” s funkcijo `fgetc()` (`fputc()`),
- hitreje: `fread()` (`fwrite()`),
- strukturo preberemo z ukazom `fread()`