

Dinamička dodjela memorije

- **Statička i dinamička dodjela memorije**

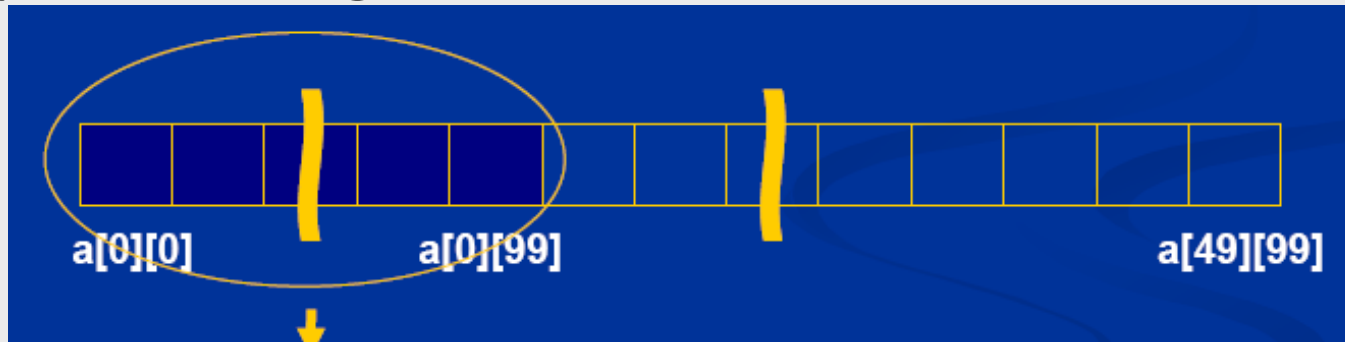
- Naredbe statičke deklaracije - na početku programa (globalne) ili na početku određene funkcije (lokalne).
- Statička deklaracija - pod kontrolom prevodioca, unaprijed rezervisan prostor.
- Naredbe dinamičke deklaracije - bilo gdje u programu, unutar bilo koje funkcije i na bilo kom mjestu.
- Dinamička deklaracija – tokom izvršavanja programa se rezervise prostor.

Dinamička dodjela memorije

- **Statička dodjela memorije**
- **char x; 1 bajt**
- **int y; 4 bajta**
- **int niz_1d[100]; 400 bajtova**
- **double niz_2d[50][100]; 40 000 bajtova**
- **Na početku programa/funkcije zna se tačan broj bajtova koji će biti rezervisan za pojedine promenljive.**

Dinamička dodjela memorije

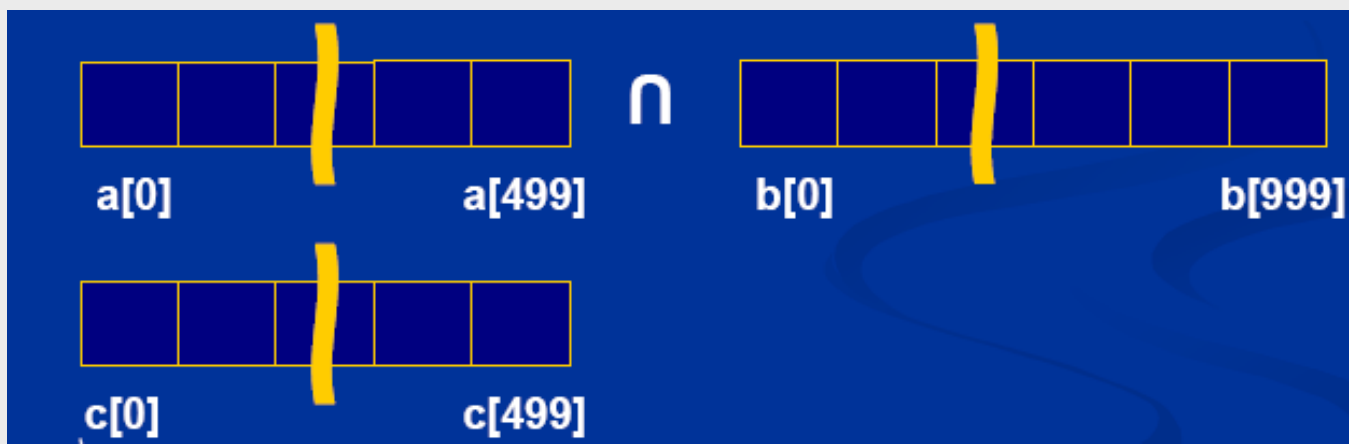
- **Statička dodjela memorije**
- Nije pogodan oblik deklaracije kada se:
- zadaje i koristi mnogo manji broj podataka od predviđenog.



- **Od 5000 promjenljivih ne koriste se sve.**

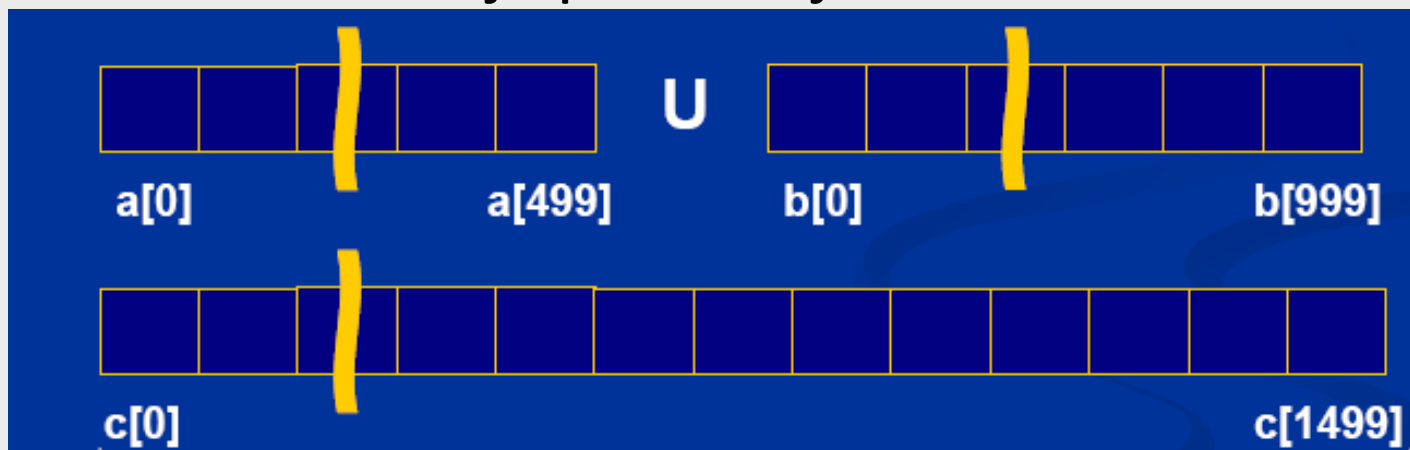
Dinamička dodjela memorije

- **Statička dodjela memorije**
- Nije pogodan oblik deklaracije kada se:
- formira kolekcija promenljivih koja predstavlja presjek od zadatih kolekcija promenljivih.
- **Možda u presjeku nije ni 500 elemenata (1. niz).**



Dinamička dodjela memorije

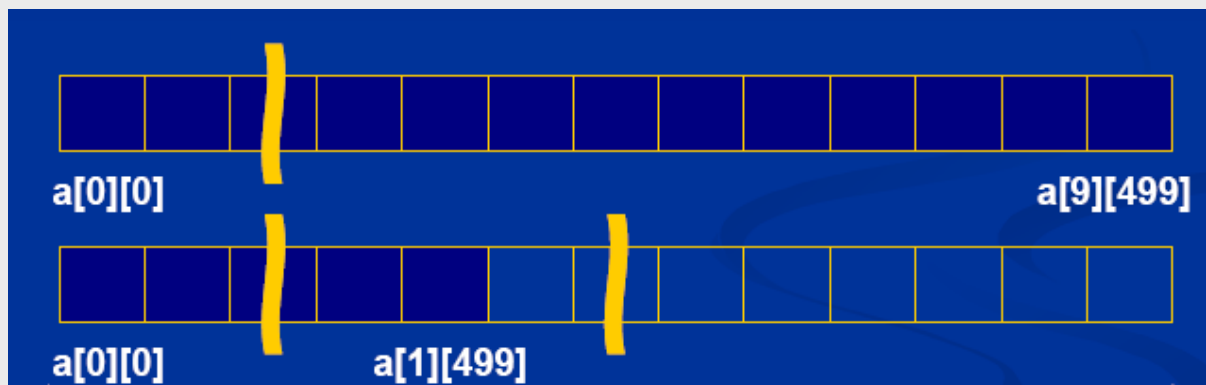
- **Statička dodjela memorije**
- Nije pogodan oblik deklaracije kada se:
- Formira kolekcija promenljivih koja predstavlja uniju od zadatih kolekcija promenljivih.



- **Možda u uniji nije svih 1500 elemenata od oba niza**

Dinamička dodjela memorije

- **Statička dodjela memorije**
- Nije pogodan oblik deklaracije kada se:
- Od određenog mjesta u programu dio zadatih podataka više ne koristi.



- **Od 5000 promenljivih program dalje koristi dio.**

Dinamička dodjela memorije



- **Statička dodjela memorije**
- Nedostaci:
- Na početku programa neophodno je donijeti odluku o veličini prostora potrebnog u memoriji – za sve objekte tog programa.
- Ne može se promijeniti veličina jednom dodeljenog prostora u memoriji.
- Ne može se prostor koji više nije od koristi jednom objektu koristiti za neki drugi objekat programa.

Dinamička dodjela memorije

- **Statička dodjela memorije**
- Koristi se:
- kod pojedinačnih promenljivih osnovnog tipa,
- kada niz bilo kog tipa ima relativno mali broj promenljivih,
- kada je kompletan niz promenljivih potreban u svim djelovima jedne funkcije.

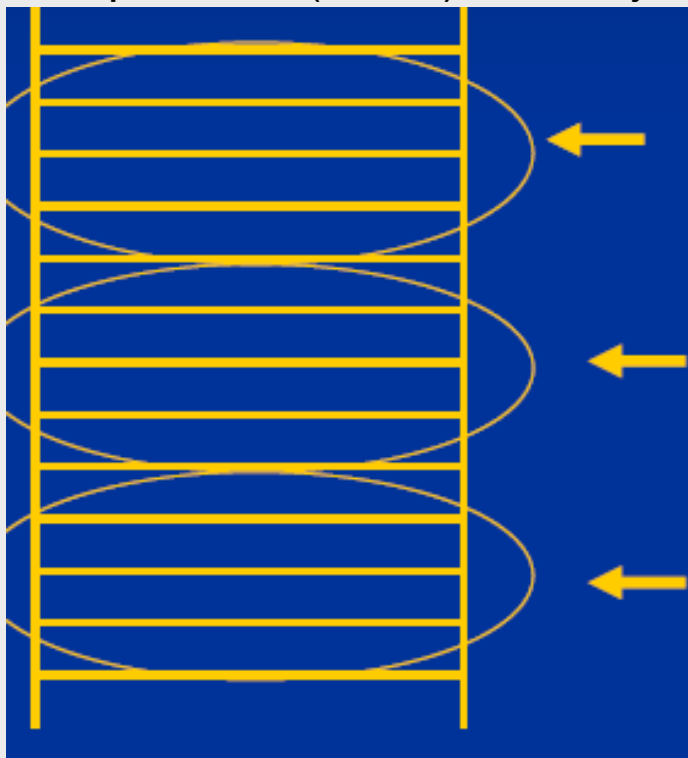
Dinamička dodjela memorije

- **Dinamička dodjela (alokacija) memorije**
- Koristi se:
- za uzimanje dijela memorijskog prostora rezervisanog za podatke, tokom izvršavanja programa.
- za vraćanje iskorišćenog memorijskog prostora, tokom izvršavanja programa da ne bi bilo nepotrebnog velikog trošenja memorije od početka do kraja izvršavanja jednog programa.

Dinamička dodjela memorije

- **Dinamička dodjela (alokacija) memorije i zone u operativnoj memoriji**

operativna(radna) memorija



Permanent storage

za instrukcije programa i za globalne statički deklarirane promenljive

Heap

za dinamičku dodelu memorije tokom izvršavanja programa

Stack

za poziv / povratak iz funkcija i za lokalne statički deklarirane promenljive

Dinamička dodjela memorije



- Dinamička dodjela (alokacija) memorije u “C-u”.
- 4 funkcije iz C biblioteke <stdlib.h>
- malloc()
- calloc()
- realloc()
- free()

Dinamička dodjela memorije

- **malloc()**, dinamička dodjela memorije za jedan blok od traženog broja bajtova (za n bajtova).
- **calloc()**, dinamička dodjela memorije za veći broj blokova i inicijalizacija svakog bajta u svakom bloku na 0 (za mxn bajtova, sa vrednošću 0).
- **realloc()**, izmena veličine prethodno dinamički dodijeljene memorije (za n2 bajtova).
- **free()**, Oslobođanje prethodno dinamički dodijeljene memorije (0 bajtova).

Dinamička dodjela memorije



- Za funkcije **malloc()**, **calloc()** i **realloc()** važi:
 - Ako je pomoću neke od funkcija nađen i dodijeljen odgovarajući prostor u memoriji – funkcija vraća u pokazivaču adresu dodeljenog prostora.
 - Ako se pomoću neke od funkcija traži veći prostor od raspoloživog u memorijskoj heap zoni funkcija vraća NULL pokazivač.

Dinamička dodjela memorije

- Deklaracija funkcije
- **void *malloc(unsigned int vel);**
- Traži se alokacija bloka u memoriji od **vel** bajtova i dobija:
 - **NULL** ako nema dovoljno memorije ili
 - **adresa bloka** (generički pokazivač void *)

Dinamička dodjela memorije

- Primjer

```
int *ptr;
```

```
ptr = malloc( 100 * sizeof(int));
```

- Uspješno izvršena naredba rezerviše memorijski blok za 100 promenljivih tipa int i adresu ovog bloka dodeljuje pokazivaču ptr, **ali ne inicijalizuje** ovaj memorijski blok.

Dinamička dodjela memorije

- Deklaracija funkcije
- **void *calloc(unsigned int n, unsigned int vel);**
- Traži se alokacija **n** blokova u memoriji, svaki od **vel** bajtova i dobija se od ove funkcije:
 - **NULL** ako nema dovoljno memorije ili
 - **adresa 1. bloka** (generički pokazivač void *), pri čemu je svaki bajt u svakom bloku **inicijalizovan na vrijednost 0.**

Dinamička dodjela memorije

- **Primjer**

```
int *ptr;
```

```
ptr = calloc( 10, 100 * sizeof(int));
```

- Uspješno izvršena naredba rezerviše i inicijalizuje (na vrijednost 0) 10 memorijskih blokova, svaki od 100 promenljivih tipa int i adresu prvog od ovih blokova dodeljuje pokazivaču ptr.

Dinamička dodjela memorije

- Deklaracija funkcije
- **void *realloc(void *ptr, unsigned int vel);**
- Traži se izmjena veličine dodeljenog prostora u memoriji, počev od adrese **ptr**, na **vel** bajtova, i dobija:
 - **NULL** ako nema dovoljno memorije za izmjenu ili
 - **adresa prostora** izmjenjene veličine (generički pokazivač void *).

Dinamička dodjela memorije

- Primjer 1:

```
int *ptr;
```

```
ptr = malloc( 100 * sizeof(int));
```

```
.....
```

```
ptr = realloc( ptr, 500 * sizeof(int));
```

- Uspješno izvršena naredba mijenja veličinu dodijeljenog prostora od **adrese ptr, sa 100 na 500** promenljivih tipa int i adresu ovog bloka dodeljuje pokazivaču ptr (**ne mijenja se postojeći sadržaj od adrese ptr**).

Dinamička dodjela memorije

- Primjer 2:

```
int *ptr;
```

```
ptr = malloc( 100 * sizeof(int));
```

```
ptr = realloc( ptr, 10 * sizeof(int));
```

- Uspješno izvršena naredba mijenja veličinu dodijeljenog prostora od adrese `ptr`, sa 100 na 10 promenljivih tipa `int` i adresu ovog bloka dodeljuje pokazivaču `ptr` (ne mijenja se sadržaj zadržanih bajtova od adrese `ptr`).

Dinamička dodjela memorije



- Deklaracija funkcije
- `void free(void *ptr);`
- Oslobađa se prostor u memoriji, počev od adrese `ptr`, samo ako je dinamički dodijeljen.

Dinamička dodjela memorije

- Primjer:

```
int *ptr;
```

```
ptr = malloc( 100 * sizeof(int));
```

```
...
```

```
ptr = realloc( ptr, 500 * sizeof(int));
```

```
...
```

```
free(ptr);
```



- Oslobadja memoriju od adrese ptr.

Dinamička dodjela memorije

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main()
{
char *bafer;
    bafer = (char *)malloc( 12 ); // Umjesto char bafer[12];

    if( bafer == NULL ) { printf("\nGreska alokacije"); exit(1); }
    printf( "\n Kreiran bafer velicine 12 bajtova");
    strcpy(bafer, "Jedan tekst");
    printf("\nSadrzaj bafera: %s", bafer);

    bafer =(char *)realloc( bafer, 24 ); // Umjesto char bafer[24];

    if(bafer == NULL) { printf("\nGreska realokacije"); exit(1); }
    printf( "\nVelicina bafera promenjena na 24 bajtova" );
    printf("\nBafer jos uvek sadrzi: %s", bafer);
    strcat(bafer, " Drugi tekst");
    printf("\nBafer sada sadrzi: %s\n", bafer); //Ne bi bilo sa char bafer[12];
    free(bafer);
    getchar();
}
```