
Inicijalizacija

- ❖ Sve nelokalne varijable (objekti i reference) sa statičkim trajanjem skladišta, a to obuhvata statičke podatke članove i varijable definisane u oblasti važenja prostora imena, inicijalizuju se pri pokretanju programa, pre poziva funkcije *main* (mada to nije garantovano i nije uvek slučaj, kao što je ranije objašnjeno)
- ❖ Sve varijable sa životnim vekom vezanim za nit (*thread_local*) inicijalizuju se pri pokretanju niti, pre izvršavanja funkcije niti
- ❖ Obe ove kategorije varijabli inicijalizuju se u dve faze, sledećim redom:
 - Statička inicijalizacija (*static initialization*)
 - Dinamička inicijalizacija (*dynamic initialization*)
- ❖ Statička inicijalizacija vrši se za vreme prevođenja
- ❖ Dinamička inicijalizacija se konceptualno vrši nakon statičke inicijalizacije, a praktično za vreme izvršavanja (pokretanja) programa, mada je u nekim slučajevima moguće, a to je prevodiocu i dozvoljeno, da i nju izvrši za vreme prevođenja (tzv. rana dinamička inicijalizacija, *early dynamic initialization*)

Inicijalizacija

❖ Statička inicijalizacija radi sledeće:

- Ukoliko je dozvoljeno, vrši se *konstantna inicijalizacija* (*constant initialization*); u praksi, ova inicijalizacija obavlja se za vreme prevođenja tako što prevodilac izračunava konstantne izraze koji su inicijalizatori, a izračunate vrednosti upisuju se u prostor alociran za varijable u prevedenim fajlovima; na primer:

```
const int n = 5;  
int x = 2*n+1, *p = &x;  
size_t s = sizeof(X);
```

- U onim slučajevima u kojima se ne vrši konstantna inicijalizacija, vrši se *inicijalizacija nulom* (*zero initialization*); u praksi, za ove varijable ne odvajaju se prostor u prevedenim fajlovima, već program u izvršavanju koristi uslugu operativnog sistema koja alocira segment memorije inicijalizovan nulama; na primer:

```
int n, *p;
```

- ❖ Dinamička inicijalizacija vrši se po redosledu koji određuju specifična pravila jezika; za nelokalne varijable koje ne spadaju u neke posebne kategorije (npr. statički podaci članovi šablonskih klasa ili *inline* varijable pod određenim uslovima), redosled je određen redosledom definisanja u jednoj jedinici prevođenja
- ❖ Ako dinamička inicijalizacija ne menja vrednost nijednog objekta iz prostora imena pre njegove inicijalizacije, i ako bi statička inicijalizacija proizvela isti rezultat kao i dinamička, prevodiocu je dozvoljeno da dinamičku inicijalizaciju obavi kao statičku, zapravo za vreme prevođenja (rana dinamička inicijalizacija, *early dynamic initialization*)
- ❖ Kao što je već objašnjeno, dinamička inicijalizacija ne mora biti završena pre početka izvršavanja funkcije *main*
- ❖ Ako dinamička inicijalizacija baci izuzetak, poziva se funkcija *std::terminate*