Dinamički životni vek

- *Placement new* zapravo omogućava inicijalizaciju objekta pozivom konstruktora za deo memorije koji je već određen za smeštanje objekta; drugačije nije moguće pozvati konstruktor koji inicijalizuje objekat na definisanom mestu u memoriji
- *Placement new* se može koristiti u sistemskim rutinama niskog nivoa, recimo u delovima sistema (npr. operativnog sistema) koji upravljaju alokacijom memorije, keširanjem blokova sa diska, implementaciji fajl sistema i slično
- Na primer, želimo da dealocirane segmente memorije uvezujemo u listu, s tim da strukturu *FreeSeg* koja predstavlja jedan slobodan segment smestimo baš na početak tog slobodnog segmenta (a ne da alociramo poseban deo memorije koji bi se trošio za evidenciju slobodne memorije):

```
class FreeStore {
public:
  inline void free (void* addr, size_t sz);
  …
private:
  class FreeSeg {
  public:
    FreeSeg (FreeSeg* nxt, size_t sz) : next(nxt), size(sz) {}
  private:
    FreeSeg* next;
    size_t size;
  };
  FreeSeg* head;
};
void FreeStore::free (void* addr, size_t sz) {
  head = new (addr) FreeSeg(head,sz);
}
```

Zadatak: implementirati funkciju *void* FreeStore::alloc(size_t size);*

Pitanje: kako postići isto bez korišćenja *placement new?*

Pitanje: šta radi i šta vraća ugrađena sistemska operatorska funkcija koja se poziva za *placement new? void* operator new (std::size_t count, void* ptr);*

Dinamički životni vek

- Izraz *new* vraća vrednost koja je po tipu pokazivač na tip specifikovan pomoću *type-id* i koji ukazuje na objekat napravljen ovim izrazom, odnosno na prvi objekat u nizu, ako je napravljen niz
- Izrazom *new* mogu se praviti objekti (ne i reference), pošto ne postoje pokazivači na reference (pa operator *new* ne može da vrati pokazivač na dinamički kreiranu referencu)
- Ova vraćena vrednost je jedina veza do napravljenog dinamičkog objekta koji je inače anoniman (nema ime); ukoliko se ova veza izgubi, objekat ostaje nedostupan, pa se ne može ni uništiti:

```
int i = new int(2);

int j = *new int(2);

int& r = *new int(2);

delete &r;
```