Динамична памет

Схема на програмната памет

Програмен стек

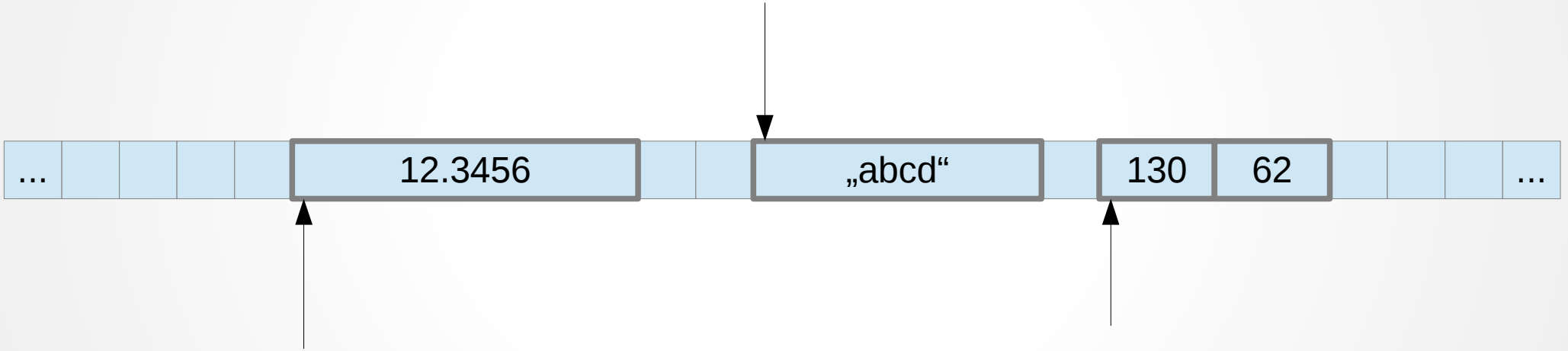
Свойства на програмния стек

- паметта се заделя в момента на дефиниция
- всеки заделен блок памет носи името на променливата
- паметта се освобождава при изход от блока (или функцията), в който е дефинирана променливата
- последно заделената памет се освобождава първа
- разработчикът няма контрол над управлението на паметта
  - паметта не може да се освободи по-рано (преди края на блока)
  - паметта не може да се запази за по-дълго (след края на блока)
- количеството заделена памет до голяма степен е определено по време на компилация
  - при какви случаи заделената памет може да варира по време на изпълнение?

Област за динамична памет (heap)

- Динамичната памет може да бъде заделена и освободена по всяко време на изпълнение на програмата
- Областта за динамична памет е набор от свободни блокове памет
- Програмата може да заяви блок с произволна големина
- Операционната система се грижи за управлението на динамичната памет
  - поддържа „карта“ кои клетки са свободни и кои заети
  - контролира коя част от паметта от коя програма се използва (защитен режим)
  - позволява използването на външни носители (виртуална памет)

Схема на динамичната памет



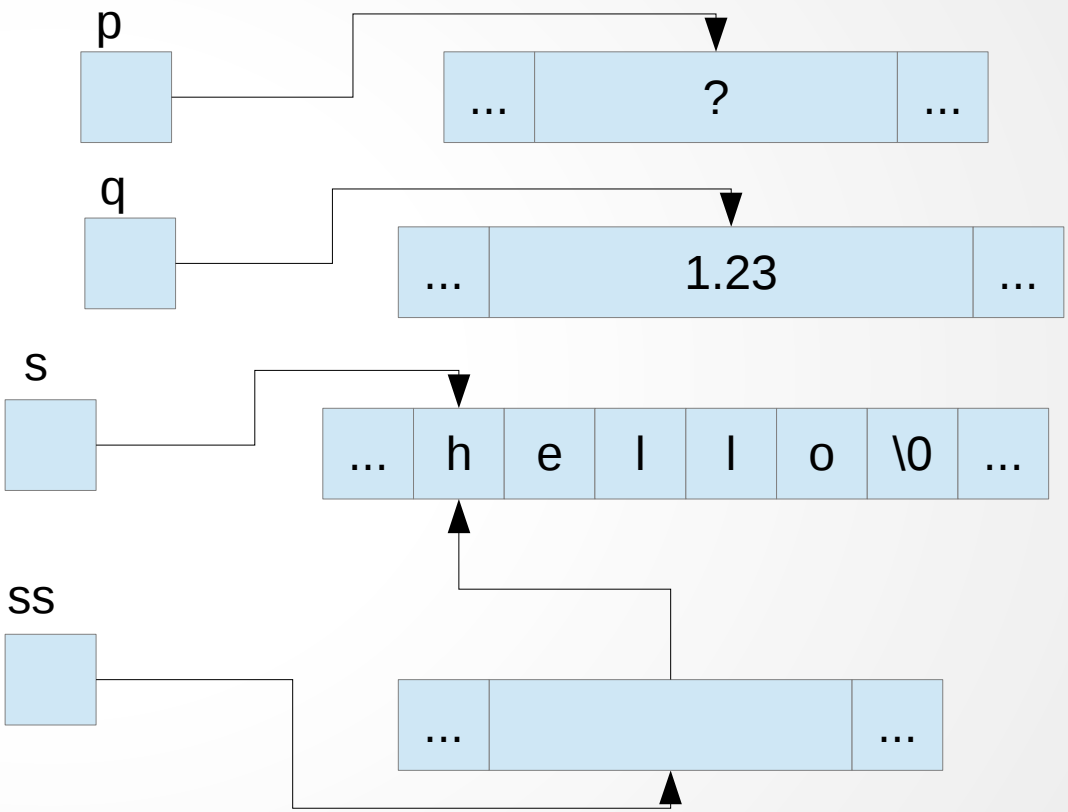
Заделяне на динамична памет

- Заделянето на динамична памет става с операциите new и new[]
- new <тип>
  заделя блок от памет за една променлива от <тип>
- new <тип>[<брой>]
  заделя блок от памет за масив от <брой> елемента от <тип>
- new <тип>(<инициализация>)

  заделя блок от памет за една променлива от <тип> и я инициализира със зададените един или повече параметри
- връща указател <тип>* към новозаделения блок
  - връща NULL, ако заявката не може да бъде изпълнена

Примери за заделяне

- int* p = new int;
- float* q = new float(1.23);
- char* s = new char[6];
  strcpy(s, „hello“);
- char** ss = new char*(s);



Освобождаване на памет

- Динамична памет се освобождава с операциите delete и delete[]
- delete <указател>
  освобождава блок от памет с начало, сочено от <указател>
- delete[[<брой>]] <указател>
  освобождава блок от памет, съдържащ масив от <брой> обекти, първият от които е сочен от <указател>
- указването на <брой> не е задължително, понеже операционната система „знае“ колко е голям заделения блок

Ограничения при освобождаването на памет

- На delete може да се подаде само указател, върнат от new
- Не е позволено освобождаването на памет в програмния стек или областта за програмен код
  int x; int* p = &x; ... ~~delete p;~~
  double (*op)(double) = sin; ... ~~delete op;~~
- Не е позволено частично освобождаване на памет
  int* a = new int[10]; ... ~~delete (a+2);~~
- Не е позволено използването на памет след като е освободена
- Не е позволено повторното освобождаване на една и съща памет
  int* p = new int[5], *q = p; ... delete p; ~~q[1] = 5; delete q;~~

Задачи

- Да се напише програма, която въвежда няколко положителни дробни числа в динамичната памет и намира средното им аритметично
- Да се напише програма, която създава матрица от числа в динамичната памет и я транспонира

Особености на динамичната памет

- Разработчикът има контрол над заделянето на памет
- Разработчикът носи отговорност за правилната работа с динамичната памет
- Заделената динамична памет остава непокътната до освобождаването ѝ с delete или до завършване на програмата
- След приключване на програмата, цялата заделена от нея памет се освобождава от операционната система
- Честото заделяне и освобождаване на малки блокове памет води до **фрагментация**

- Динамично заделените блокове памет не се свързват с имена

Грешки при работа с динамична памет

- Работа с указател към незаделена или освободена памет
- Освобождаване на непозволена памет
- „Загубване“ на указател към заделена памет
  - изтичане на памет (memory leak)
- Неосвобождаване на неизползвана памет