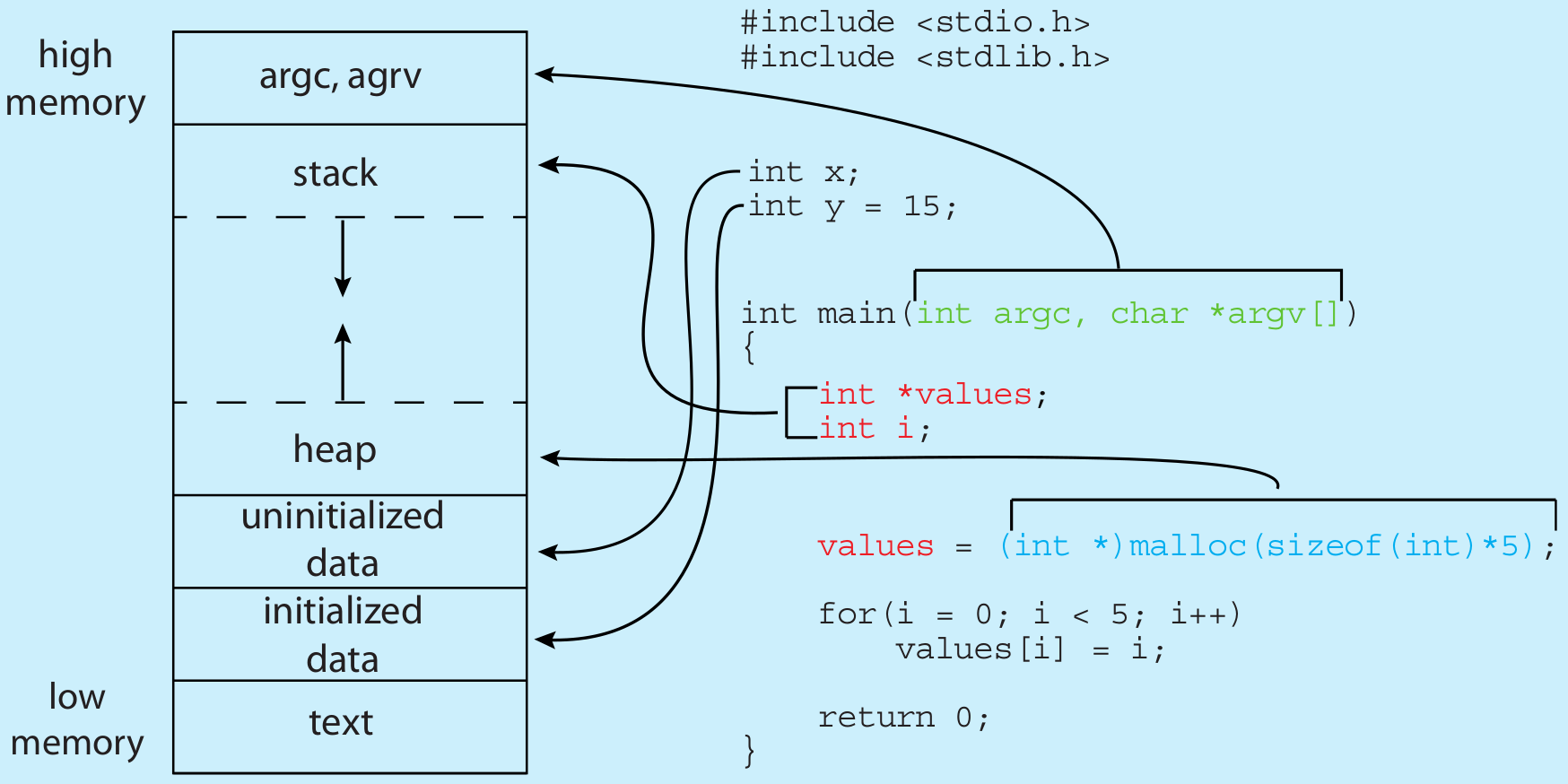
**Systemy operacyjne**

**Laboratorium 04**

**Pamięć**

**Proszę zapoznać się z:**

* Polecenia: ltrace i strace, vmstat
* Polecenie pagemon, np.: sudo pagemon -p `pidof xclock` (dla procesu xclock)
* Polecenie smem, np.: sudo smem --piename -c “pss”
* Tworzenie pliku/partycji wymiany: dd/fdisk, mkswap, swapon
* Czyszczenie pamięci:
  + # sync; echo 1 > /proc/sys/vm/drop\_caches
  + 1, 2 lub 3
  + Oraz swap:
  + swapoff -a && swapon -a



**Zadanie 1.** Stos i sterta: **stackheap.cc**

Napisać program, który działa wg schematu:

* Z funkcji głównej wywoływane są kolejno dwie funkcje:
  + void statyczna();
  + void dynamiczna();
  + Realizujące odpowiednio statyczny i dynamiczny przydział pamięci.
* W funkcji statyczna() proszę utworzyć znacznych rozmiarów zmienną lokalną, automatyczną, np.: double tablica[10^6] i zatrzymać działanie programu, np. funkcją blokującą wczytywanie znaku z klawiatury.
* W funkcji dynamiczna() proszę utworzyć znacznych rozmiarów zmienną lokalną, dynamiczną, np.: double \*tablica = new double[10^6] i zatrzymać działanie programu, np. funkcją blokującą wczytywanie znaku z klawiatury. Przed zakończeniem działania funkcji proszę pamiętać o zwolnieniu zajmowanej pamięci.
* W obu funkcjach proszę zaraz po utworzeniu tablic wypełnić je liczbami losowymi.
* Między wywołaniami funkcji statyczna() i dynamiczna() proszę wstrzymać działanie programu.
* Proszę w trakcie działania programu, w miejscach jego zatrzymania zaobserwować zajętość pamięci przez program (np. htop, /proc/[id]/smaps, itp.).
* Zsumuj wartości RSS w pliku /proc/[id]/smaps i odnieś do wartości RSS z htop.

**Zadanie 2.** Rozrost zajętości pamięci: **incmemusage.cc**

Warsztat: (a) lista wskaźnikowa lub (b) tablica tablic.

(a) Materiał do poznania:

Lista wskaźnikowa / lista odsyłaczowa:

[AiSD-skrypt.pdf](https://drive.google.com/file/d/1sbtSO_BA_Ed6woifkCYo4T2NpCL7sn3m/edit) - strona 37, odniesienie do:

Thomas Cormen, „Algorytmy i Struktury Danych”, rozdział 11, str. 240 – 244.

(b) Zadeklarować tablicę (np. 1000 komórek) wskaźników do tablic 2-wymiarowych (np. 1000x1000) typu double.

Napisać program, który przy pomocy (a) lub (b) rozrasta się do pewnej, zadanej wielkości.

Program tworzy (a) lub (b) i oczekuje na naciśnięcie klawisza, aby zakończyć działanie.

Proszę prześledzić działanie programu za pomocą: free, htop, vmstat oraz zawartość /proc

Język programowania: dowolny, zalecany czysty ANSI C.

Proszę dodać przed każdym utworzeniem nowej komórki (nowego elementu w (a) lub nowej tablicy w (b)) przerwę i zwrócić uwagę jak w czasie rozrasta się zajętość pamięci.

**Zadanie 3.** Kontrola zajętości pamięci

Zapoznać się z rozwiązaniem ‘ulimit’ i w taki sposób zestawić ograniczenie, aby przerwać działanie programu z zadania 2 w zadanym momencie.

Czy ulimit dotyczy konta, czy pojedynczego procesu ?

**Zadanie 4.** Śledzenie wykonania programu

Zapoznać się z wynikiem działania programu strace, np.:

$ strace date

Proszę zinterpretować wynik działania.

Lub napisać prosty program typu “Hello world”.

**Zadanie 5\*.** Odkrywanie położenia stron pamięci wirtualnej dla danego procesu

Podstawowe informacje omówione zostały na wykładach pt. “Pamięć główna” oraz “Pamięć wirtualna”.

Program, po uruchomieniu staje się procesem. Odwołania do pamięci realizowane są poprzez numer strony oraz offset. Jednostka zarządzania pamięcią (ang. Memory Management Unit) wykonuje tłumaczenie adresu na ramkę oraz offset w pamięci głównej.

W przypadku braku wolnej przestrzeni w pamięci głównej następuje przenoszenie ramek do pamięci wymiany. Zadaniem jest wykazanie, że tak się faktycznie dzieje oraz wykazanie, jaka % (procentowo) wyrażona liczba ramek (lub jaka % ilość pamięci) z całego procesu znajduje się w pamięci głównej, a jaka w pamięci wymiany.

Elementy wymagane do realizacji zadania można przygotować w następujący sposób:

* sprawdzić istnienie przestrzeni wymiany:
  + $ free
  + razem użyte wolne dzielone buf/cache dostępne
  + Pamięć: 7942508 4409368 201752 538728 3331388 2688124
  + **Wymiana: 2097148 38912 2058236 <- potwierdza pamięć wymiany**

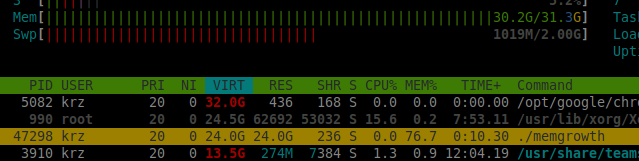
lub:

* + $ htop
  + 
* jeśli przestrzeń wymiany nie istnieje, czyli wynik jest taki:
  + $ free
  + Wymiana: 0 0 0

lub:

* + 
* to przygotować ją w postaci pliku (nie musi być to partycja):
  + $ dd if=/dev/zero of=swapfile bs=512 count=1024000 <- 500 MiB
  + $ chmod 0600 swapfile
  + $ mkswap swapfile
  + $ sudo chown root.root swapfile
  + $ sudo swapon swapfile
  + efekt:
  + 
* przygotować program (np. z Zadania 2.) wypełniający pamięć operacyjną - może to być program wywoływany z parametrem określającym wielkość tablicy do przydziału dynamicznego, ruchamiany wielokrotnie, albo program, który w kontrolowany sposób (np. Kolejnymi naciśnięciami klawisza) powiększa zajętość pamięci, ponad wolną przestrzeń w pamięci głównej.
* na przykładzie tak przygotowanego programu przedstawić rozwiązanie zadania.

Przykład efektu, kiedy system operacyjny zaczyna korzystać z pamięci wymiany:



**Pytanie dodatkowe:** zakładając, że zajętość pamięci główej spadła tak, że korzystanie z pamięci wymiany w danym momencie nie jest konieczne, jak zmusić system operacyjny, do przeniesienia zawartości z pamięci wymiany do pamięci głównej ?