Operacijski sistemi

Upravljanje s pomnilnikom

Jurij Mihelič, FRI, Uni LJ

Vsebina

- Naslovni prostor
- Pomnilniški API
- Pomnilniške storitve OS

Naslovni prostor

- Pomnilnik
  - fizični pomnilnik
    - je pomnilnik, ki je dejansko fizično povezan s procesorjem
    - proces hrani podatke v fizičnem pomnilniku
  - naslovni prostor
    - pomnilnik kot ga vidi proces
    - OS abstrahira fizični pomnilnik
      - zaradi lažje uporabe

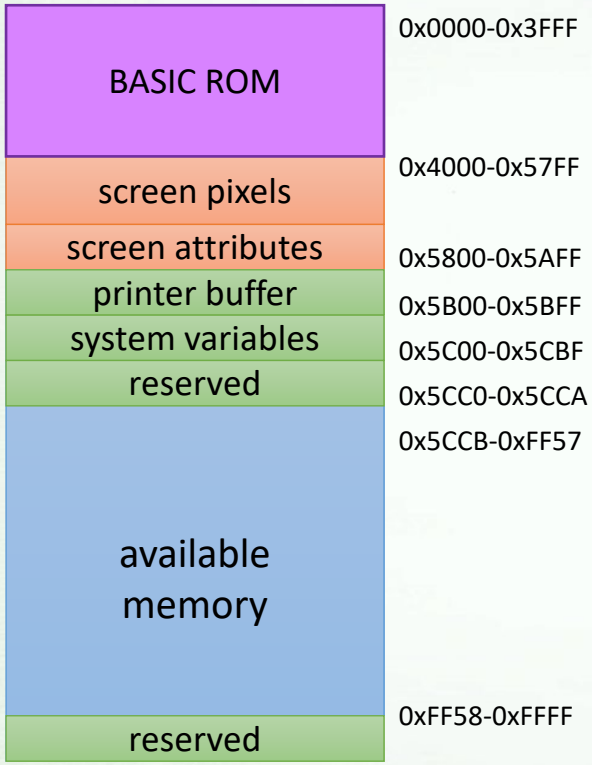
Naslovni prostor

- Zgodnji preprosti sistemi
  - brez posebnega mehanizma
  - logični naslov = fizični naslov
  - delitev pomnilnika
    - del rezerviran za OS
    - ostalo za proces
  - OS kot knjižnica

Naslovni prostor

- Zgodnji preprosti sistemi

ZX Spectrum 48 K

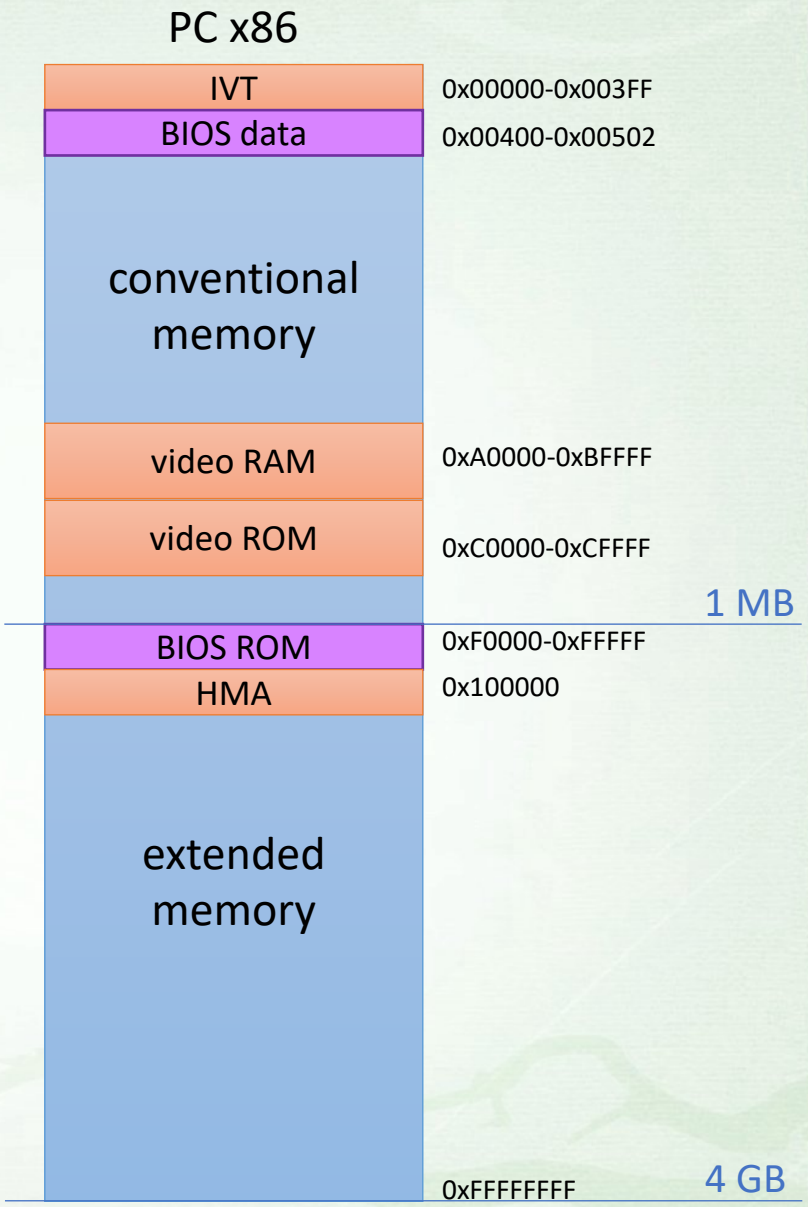



```
 310 LET A=USR 18288
 320 IF INKEY$="" THEN GOTO 320
 330 POKE 17901,INT (RND*128)
 335 IF PEEK 21623<>178 THEN POK
E ((PEEK 16396+256*PEEK 16397)+6
6),0
 340 POKE 16522,CODE INKEY$
 350 LET A=USR 18224
 360 IF PEEK 16519>=128 THEN GOT
O 5000
 370 FOR N=0 TO 5
 380 NEXT N
 390 IF PEEK (PEEK 16514+17920)<
>45 THEN GOTO 330
 400 FOR N=0 TO 30
 405 POKE 17901,INT (128*RND)
 410 LET A=USR 18224
 420 FOR M=0 TO 3
 425 NEXT M
 430 NEXT N
 440 CLS
 450 GOTO 2520

 370 FOR N=0 TO 0
```

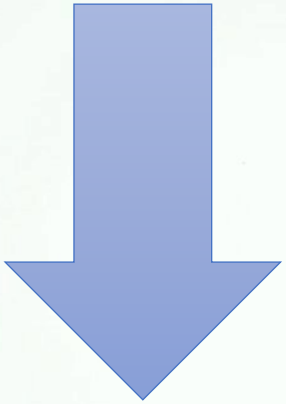
Naslovni prostor

- Zgodnji preprosti sistemi

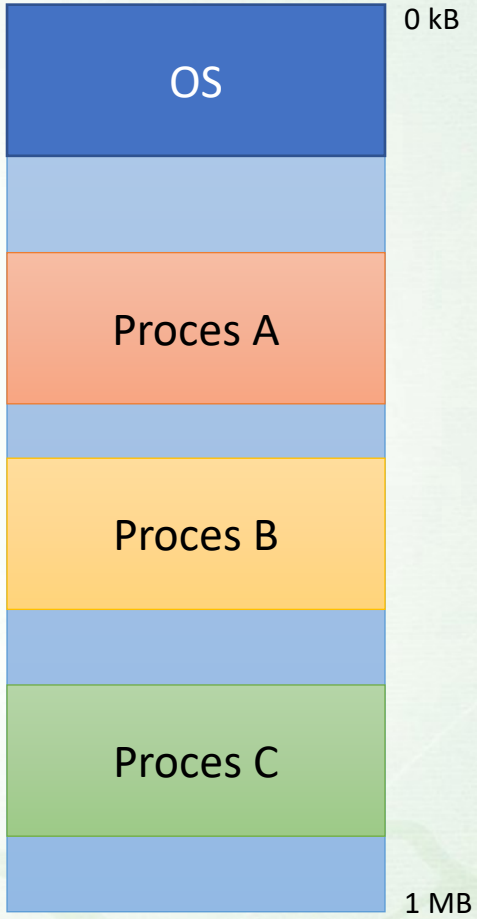


Naslovni prostor

- Večprogramiranje
  - več procesov v pomnilniku



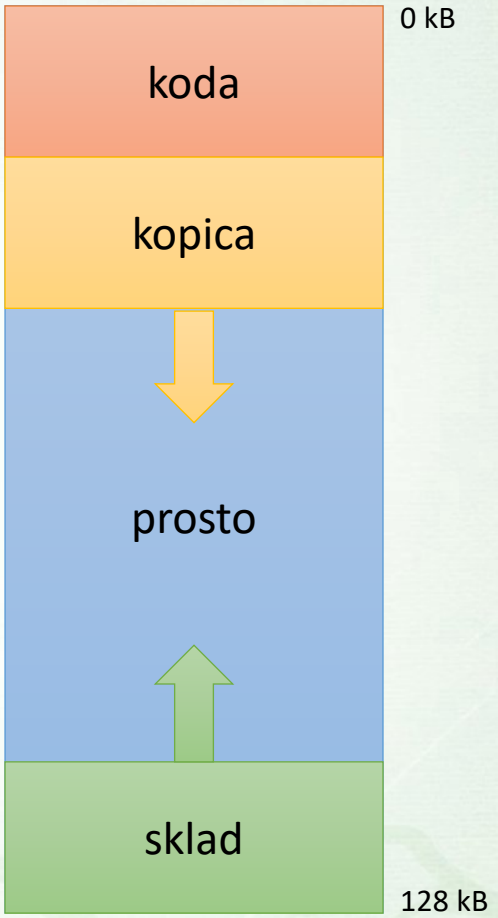
  - deljenje pomnilnika
  - preklapljanje procesov
  - zaščita procesov



Izgled fizičnega pomnilnika

Naslovni prostor

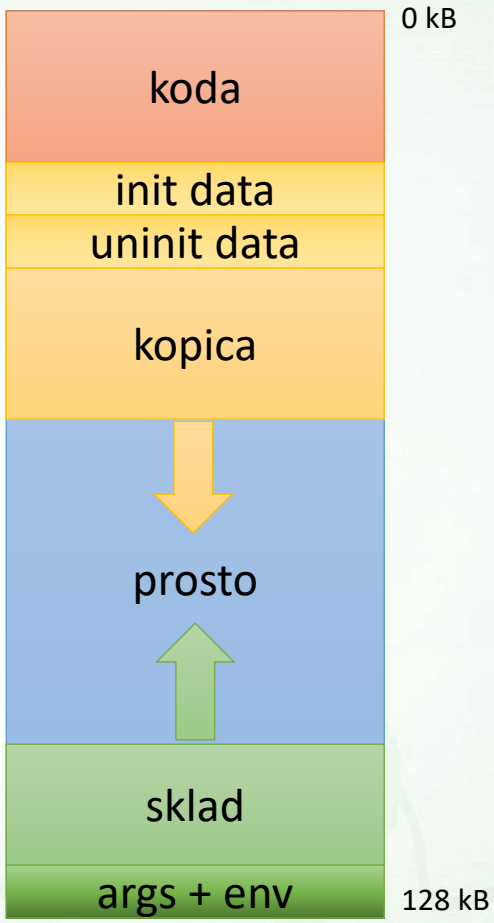
- Naslovni prostor
  - pomnilnik kot ga vidi proces
  - vidi samo svoj kos
  - sestava
    - koda
    - kopica
    - sklad
    - drugo
  - **preslikava**
    - navidezni/logični naslov v fizični naslov



Naslovni prostor

Naslovni prostor

- Naslovni prostor programov v C



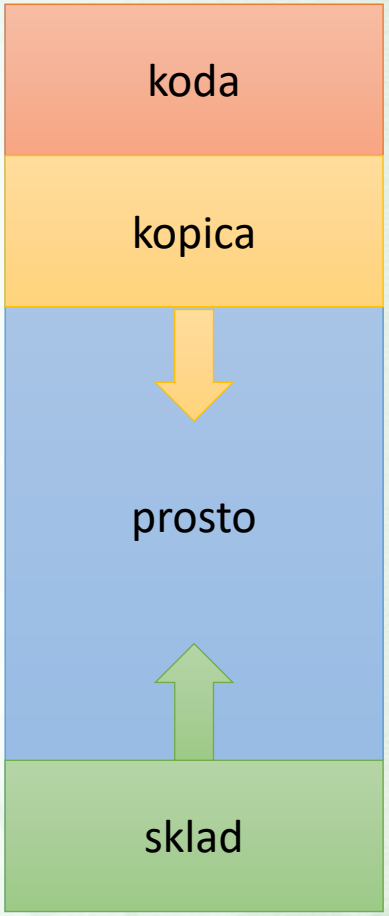
- koda – strojna koda (text)
- init data – inicializirani podatki (data)
- uninit data – neinicializirani podatki (bss)
- kopica – dinamična alokacija (heap)
- sklad – sklad (stack)
- args – argumenti ukazne vrstice
- env – okoljske spremenljivke

Naslovni prostor

- Določanje naslovov
  - čas prevajanja (compile time)
    - zbirnik ali prevajalnik določi naslove simbolov oz. spremenljivk
  - čas povezovanja (link time)
    - zlaganje več prevedenih modulov skupaj
  - čas nalaganja (load time)
    - prenaslovljiva koda (relocatable code)
    - nalaganje na poljubno lokacijo v pomnilnik
  - čas izvajanja (run time)
    - premikanje in zamenjava procesov (zunanji pomnilnik)

Naslovni prostor

- Zaščita procesa
  - izolacija naslovnega prostora
    - OS skrbi za ustreznost preslikave VA → PA
  - ščitenje delov naslovnega prostora
    - preverjanje naslovov
      - preslikava je za proste dele neveljavna (invalid)
    - dovoljenja za operacije
      - branje, pisanje, izvajanje
    - past zaščite



Pomnilniški API – libc

- Alokacija pomnilnika
  - **sklad**
    - implicitna alokacija in dealokacija preko prevajalnika
    - avtomatski pomnilnik
    - lokalne spremenljivke, argumenti funkcij, izvajalni podatki
  - **kopica**
    - eksplicitna alokacija pomnilnika
    - dealokacija
      - eksplicitna dealokacija ali
      - smetiščenje (garbage collection)

Pomnilniški API – libc

- Standardna knjižnica v C
  - alokacija pomnilnika
    - `void* malloc(size_t size)`
    - glej tudi: `calloc(...)`, `realloc(...)`
  - dealokacija pomnilnika
    - `free(void* ptr)`
  - različni alokatorji
    - GNU malloc, tcmalloc, jemalloc, Hoard, Lockless, lockfree, dlmalloc, ...
    - diplomska naloga: M. Zavrtanik, FRI-UL
      http://eprints.fri.uni-lj.si/3464/,
      https://ipsitransactions.org/journals/papers/tar/2017jan/p10.pdf

Pomnilniški API – libc

- Klasične programerske napake
  - alokacija premalo pomnilnika
    - malloc(strlen(s) + 1)
  - pomnilnik ni inicializiran
    - C: ne inicializira, java: inicializira na 0
  - prezgodnja sprostitev pomnilnika
    - *dangling pointer problem: free(p); *p += 1;*
  - večkratna sprostitev pomnilnika
    - *double free problem: free(p); free(p);*
  - napačna sprostitev pomnilnika
    - free(malloc(42)+1);
  - puščanje pomnilnika (memory leak)
    - pozabimo sprostiti pomnilnik

Motiti se je človeško

neumno pa je napake ponavljati.

Pomnilniške storitve OS

- **Kaj** naredi `exit()` glede pomnilnika?
  - sprosti naslovni prostor procesa
  - puščanje pomnilnika
    - gledano izven procesa (v okviru sistema) ni možno
    - možno le znotraj procesa
  - programi
    - puščanje pomnilnika ni huda težava
    - koda lahko pogrne pri recenziji kode (code review) ☺
  - dolgotrajni programi
    - puščanje pomnilnika lahko povzroči sesutje procesa
    - npr. strežniki, servisi, demoni, zahtevna obdelava

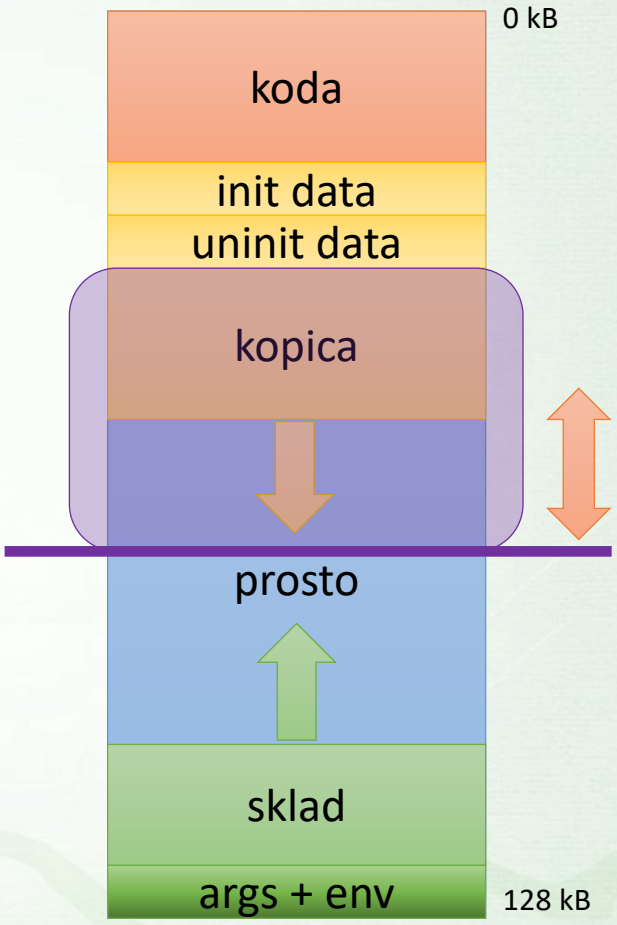
Pomnilniške storitve OS

- Velikost podatkovnega segmenta
  - oz. velikost kopice
  - sistemski klic `brk(...)`

Nastavljanje velikosti
- `void* brk(void* addr)`

Spreminjanje velikosti
- `void* sbrk(int incr)`

Trenutna velikost
- `brk(0)`



Pomnilniške storitve OS

- Alokacija in preslikava pomnilnika

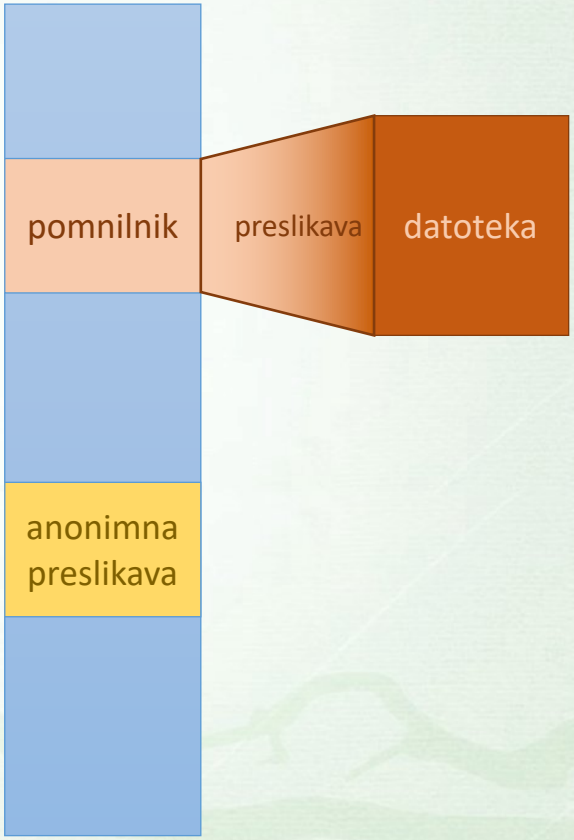
Alokacija in preslikava

- `void* mmap(void* addr, size_t len, int prot, int flags, int fd, off_t offset)`

Sproščanje pridobljenega pomnilnika

- `void* munmap(void* addr, size_t length)`

```
char *p = mmap(
    NULL,
    4096,
    PROT_READ | PROT_WRITE,
    MAP_PRIVATE | MAP_ANONYMOUS,
    -1,
    0);
...
munmap(p, 4096);
```



Pomnilniške storitve OS

- Randomizacija naslovnega prostora
  - ASLR – addres space layout randomization
  - KASLR – kernel ASLR
  - napad
    - skok v okvarjeno kodo
      na nek znan fiksen naslov
  - rešitev
    - naključna razporeditev
      ključnih delov naslovnega prostora
    - napadalec težje najde pravi
      ciljni naslov skoka
  - OS + prevajalnik oz. povezovalnik