

Vežbe 6

- Teme koje ćemo preći:
 - Memorijska organizacija procesa
 - Pravljenje novog procesa `fork()/exec()`
 - Organizacija procesa:
 - Odnos roditelj/dete (init proces i hijerarhija)
 - Specijalni tipovi procesa: siroče, zombi, demon
 - Jednostavna razmena podataka između roditeljskog procesa i dete procesa (PIPE)

Memorijska organizacija procesa

- Slika je pozajmljena iz TLPI (Figure 6-1)

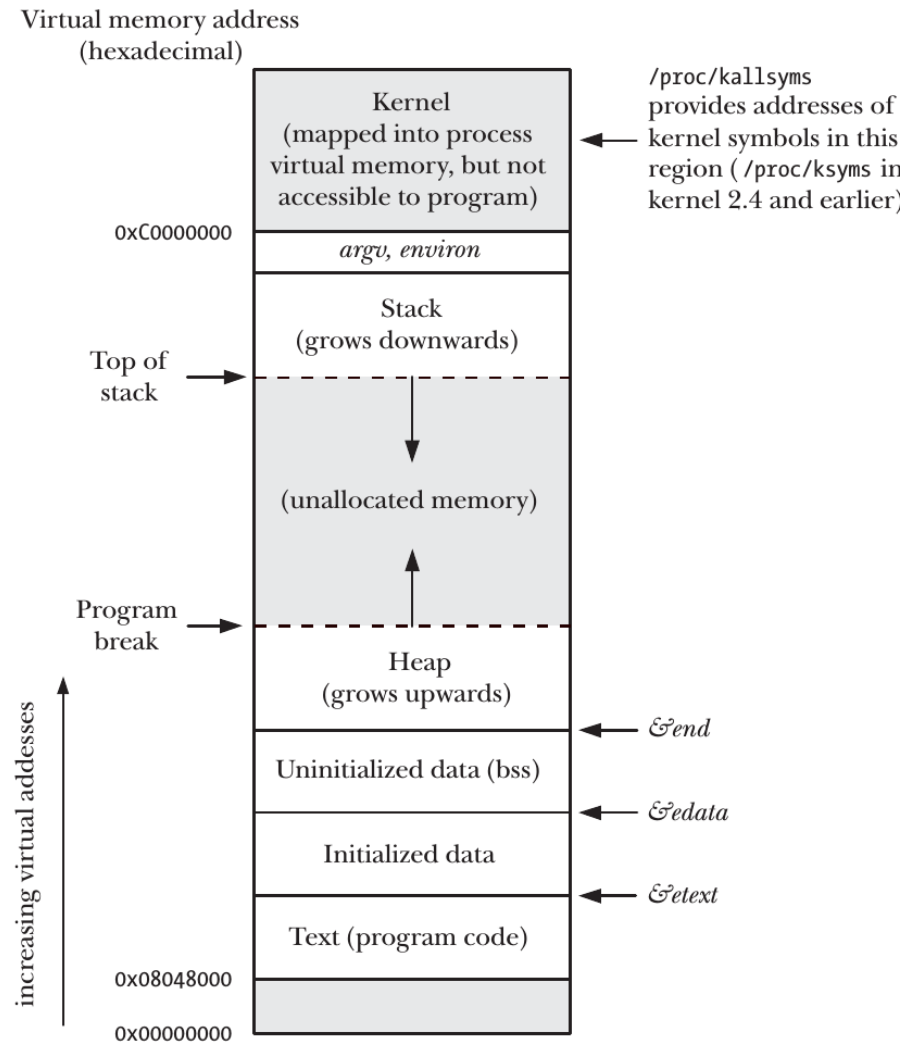


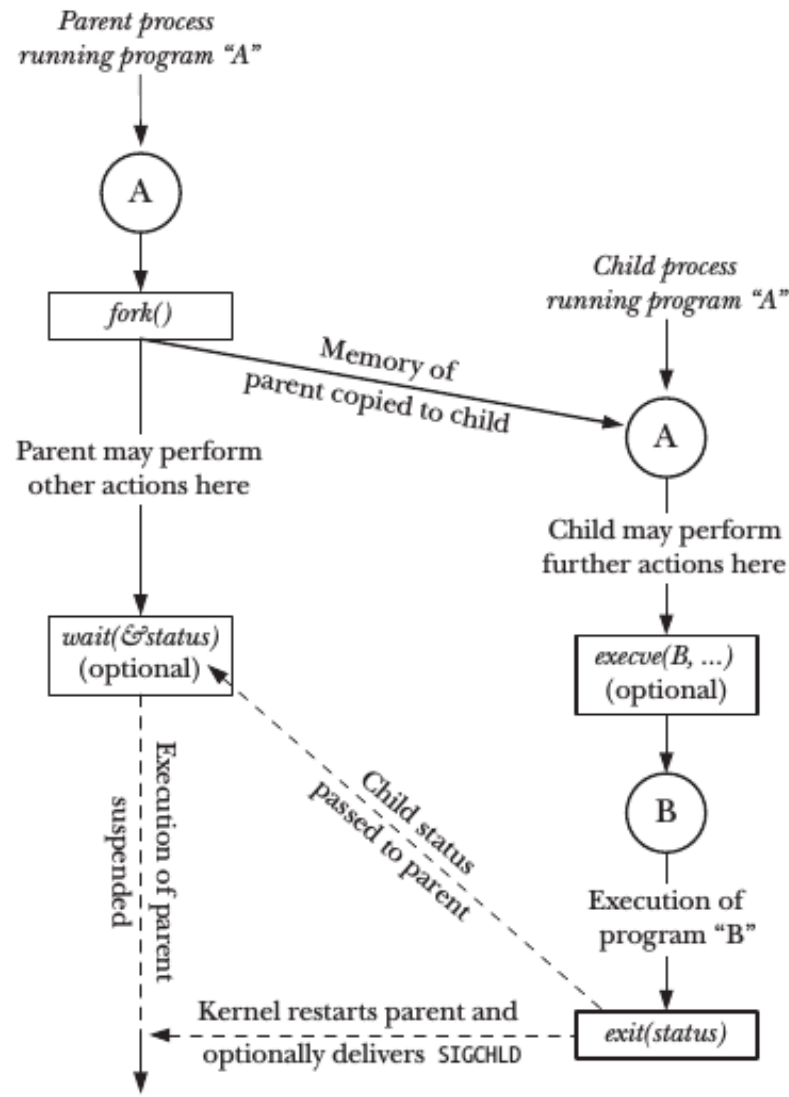
Figure 6-1: Typical memory layout of a process on Linux/x86-32

...

- Za više detalja koji uključuju gde se koje promenljive čuvaju i zašto, šta se dešava sa adresama programa itd.:
 - TLPI 6.3-6.6

Pravljenje novog procesa

- Slika je pozajmljena iz TLPI (Figure 24-1)



...

- Detaljnije pročitati:
 - TLPI 24.2 do 24.2.1, *fork()* (ne uključujući 24.21.)
 - TLPI 27.2 do 27.2.1 *exec()* familija (ne uključujući 27.2.1)

Organizacija procesa

- Prvi proces koji se pokrece na sistemu naziva se *init* proces i on ima *pid* 1 (neke od implementacija su *systemd* i *upstart*)
- Svaki proces osim *init* procesa ima roditeljski proces
- Dakle, svi procesi na sistemu su organizovani kao stablo (**nije binarno**)
- Ispis drvolike strukture procesa se može dobiti komandom *ps tree* (man 1 *ps tree*)

...

- Kada proces prekinе sa izvršavanjem kernel operativnog sistema čuva informacije o izvršavanju koje roditeljski proces može dohvatiti nekim od *wait()* sistemskih poziva (man 2 wait)
- Kernel čuva ove informacije jer nas najčešće zanima šta se desilo sa procesom koji smo mi pokrenuli
- Šta se dešava kada niko ne sačeka na proces (niko ne pozove *wait()* sistemski poziv)?

Zombi proces

- Ako roditeljski proces ne sačeka na dete, informacije o izvršavanju deteta se čuvaju u memoriji i dete se pretvara u *zombi* proces
- Previše zombi procesa može usporiti pa i oboriti rad sistema
- Ako roditeljski proces završi sa izvršavanjem, a ne sačeka na dete, *init* postaje roditeljski proces deteta i nakon završavanja čeka na njega (izbegava se zombi situacija)

Siroče i demon

- Na prethodnom slajdu je opisana situacija kada se roditeljski proces završio pre deteta → takvo dete se naziva *siroče* (eng. orphan)
- Nekada imamo procese za koje nam je bitno da rade u pozadini i ne želimo da čekamo na njih. Ovakvi procesi se nazivaju *demoni* (eng. Daemon)
 - Primer ovakvog procesa je Veb server:
 - Ovaj program radi u pozadini, prima zahteve preko mreže i isporučuje, opet preko mreže, tražene sadržaje. Jasno je da je prirodno da ne prestaje sa radom, nego da čeka na nove zahteve i da ih redom obrađuje.

Jednostavna komunikacija između deteta i roditelja - PIPE

- Roditeljski i dete proces često moraju da razmenjuju informacije – zgodno je koristiti PIPE
- PIPE možete zamisliti kao JEDNOSMERNU ulicu ograničenog kapaciteta (može stati X automobila). Smer ulice može biti i na jednu i na drugu stranu, međutim kada se odlučimo nema menjanja smerova.
- Automobili odgovaraju bajtovima koji se razmenjuju, smer odgovara toku komunikacije
 - dete → roditelj ili
 - roditelj → dete
- Detaljnije TLPI 44.1 do 44.3 (bez 44.3)