



Naloga 2

Napišite program v programskem jeziku C, ki ga uporabljamo na naslednji način:

- `./Naloga2 akcija parametri`

Prvi argument je torej *akcija* in predstavlja opravilo, ki naj se izvede. Argumenti od drugega naprej pa so *parametri* akcije. Program naj podpira spodaj naštetе akcije. Ob uspešnem izvajanju naj bo izhodni status enak 0, razen če akcija ne zahteva drugače.

Več akcij izpisuje različne informacije o sistemu oz. o procesih, zato je pri njih prvi parameter akcije pot do imenika (`proc` sistem), ki vsebuje informacije o jedru (privzeta vrednost parametra je `/proc`). V spodnjih primerih mu podamo pot do imenika `proc-demo`, katerega najdete v arhivski datoteki `proc-demo.tgz`. Testni `proc` sistemi ne bodo vseboval vseh podatkov iz pravega `proc` sistema, le nekatere pomembnejše datoteke, ki jih potrebujete za izvedbo naloge.

Pri implementaciji smete uporabljati ovojne funkcije sistemskih klicev, kot so npr. `fork`, `wait`, `exec`, `pause`, `kill`, `signal`, `sleep`, `getpid`, `getppid`, `opendir`, `readdir`, `closedir`, itd. Od "višje nivojskih" funkcij so dovoljene funkcije za delo z nizi, npr. `strlen`, `strcmp`, `strcasecmp`, `atoi`, itd., za delo s pomnilnikom, npr. `malloc`, `free`, `memcpy`, itd., za branje in pisanje (izpis), npr. `fopen`, `fclose`, `fscanf`, `scanf`, `printf`, `fprintf`, `sprintf` itd. Izvajanje ukazov, npr. `ps`, `uname`, iz okolja **ni** dovoljeno - vse informacije o sistemu preberite iz sistema `proc`. Prav tako **ni** dovoljeno uporabiti funkcij, ki združujejo več operacij, npr. `scandir`.

V primeru nejasnosti se posvetujte z najbližjim asistentom.

Akcija `sys`

Izpiše osnovne informacije o sistemu:

- različica Linux jedra,
- različica prevajalnika `gcc` za prevajanje jedra,
- prva swap particija,
- število jedrnih modulov.

Izpis naj bo formatiran kot je razvidno iz primera. Primer:

```
$ ./Naloga2 sys proc-demo
Linux: 3.10.0-327.36.2.el7.x86_64
gcc: 4.8.5
Swap: /dev/dm-1
Modules: 59
```

Akcija `pids`

Naraščajoče urejeno izpiše številke vse procesov (PID), vsak PID naj bo v svoji vrstici. Primer izpisa:

```
$ ./Naloga2 pids proc-demo
1
10
11
12
14
15
16
20
```

Akcija names

Izpiše PIDE in imena vseh procesov nepadajoče urejeno po imenih: vsako ime v svoji vrstici, v primeru enakih imen se upošteva urejenost po PIDu. Primer izpisa:

```
$ ./Naloga2 names proc-demo
1 bash
10 bash
20 copyproc.sh
14 dash
15 dash
16 dash
11 sleep
12 sleep
```

Akcija ps

Izpiše naslednje informacije o procesih: PID, PPID, stanje in ime procesa. Izpis naj bo urejen po PIDih. Primer izpisa:

```
$ ./Naloga2 ps proc-demo
PID  PPID STANJE IME
  1    0      S bash
 10    1      S bash
 11   10      S sleep
 12   10      S sleep
 14    1      S dash
 15   14      T dash
 16   14      T dash
 20   14      S copyproc.sh
```

Za formatiranje izpisa lahko uporabite naslednji format:

- `printf("%5s %5s %6s %s\n", "PID", "PPID", "STANJE", "IME");`

Če je podan še dodatni argument - število, le-ta predstavlja PID. V tem primeru izpišite zgolj procese, katerih **prednik** je proces s podanim PID (vključno z danim procesom). Primer izpisa:

```
$ ./Naloga2 ps proc-demo 10
PID  PPID STANJE IME
 10    1      S bash
 11   10      S sleep
 12   10      S sleep
```

Akcija psex

Ta akcija je nadgradnja akcije ps. Dodatno izpišite še število niti in število odprtih datotek procesa. Za formatiranje izpisa uporabite naslednji format:

- `printf("%5s %5s %6s %6s %6s %s\n", "PID", "PPID", "STANJE", "#NITI", "#DAT.", "IME");`

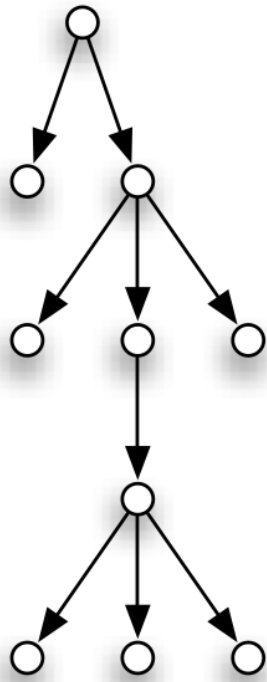
Prav tako naj akcija podpira dodatni argument - PID prednika. Primer izpisa:

```
$ ./Naloga2 psex proc-demo 14
PID  PPID STANJE  #NITI  #DAT. IME
 14    1      S      1      4 dash
 15   14      T      1      4 dash
 16   14      T      1      4 dash
 20   14      S      1      4 copyproc.sh
```

Akcija forktree

Ustvarjanje podane hierarhije procesov in njen izpis s klicem ukaza `pstree`. Na standardnem vhodu bo podano drevo kot zaporedje stopenj vozlišč po posameznih nivojih drevesa. Začetni proces naj sporži ustvarjanje podanega drevesa s klici `fork`. Po stvaritvi drevesa naj začetni proces s kombinacijo klicev `fork` in `exec` zažene ukaz `pstree` tako, da se bo izpisala ustvarjena hierarhija procesov vključno z začetnim procesom (koren tega drevesa je torej začetni proces). Pri uporabi ukaza `pstree` vklopite stikalo `-c`. V vašem programu ne izvajajte nobenega izpisa, saj bo edini izpis na standardni izhod opravil `pstree`.

Kodiranje drevesa. Ponazorimo kodiranje drevesa z zaporedjem stopenj vozlišč s primerom.



Zgornje drevo predstavimo s kodo 2 0 3 0 1 0 3. Koren ima stopnjo 2, njegov levi otrok stopnjo 0, desni otrok pa stopnjo 3. Na naslednjem nivoju ima prvo vozlišče (vedno gledano od leve proti desni) stopnjo 0, nato stopnjo 1 in zadnje stopnjo 0. Na predzadnjem nivoju je samo eno vozlišče, ki ima stopnjo 3. Tu se vhodno zaporedje zaključi, kar pomeni, da imajo vsa vozlišča, katerih stopnje nismo podali, stopnjo 0.

Primer z razlago.

Izpis po zagonu ukaza `./Naloga2 forktree <<< "1 5 0 3"`:

```

$ ./Naloga2 forktree <<< "1 5 0 3"
Naloga2-+-Naloga2-+-Naloga2
  |           | -Naloga2-+-Naloga2
  |           |           | -Naloga2
  |           |           ` -Naloga2
  |           | -Naloga2
  |           | -Naloga2
  |           ` -Naloga2
  ` -pstree
  
```

Začetni proces `Naloga2` in `pstree` sta vedno prisotna. Začetni proces je koren drevesa kodiranega z "1 5 0 3". Torej ima koren 1 otroka. Ta ima nato 5 otrok - vnuki. Prvi vnuk nima otrok, drugi vnuk pa ima 3. Vsi sledeči procesi nimajo otrok.

Prevajanje

Vaš program se bo prevedel s prevajalnikom `gcc` verzije 5.4. Torej lahko uporabljate razširitve jezika, kot da bi bilo vklopljeno stikalo `-std=gnu11`.

Če uporabljate sistemski klic `wait()`, ne pozabite vključiti zaglavja `wait.h`, sicer se znajo dogajati čudne stvari.

Status oddaje naloge

Status oddaje naloge	Pri tej nalogi vam ni treba oddati ničesar.
	Pri tej nalogi ne morete oddati prispevkov.
Stanje ocen	Neocenjeno
Rok za oddajo	nedelja, 13. maj 2018, 23:55
Preostali čas	Rok za oddajo naloge je potekel
Zadnja sprememba	-
Komentar oddaje	► Komentari (0)

Odziv

Ocena	9,00 / 9,00
Ocenjeno v	četrtek, 7. junij 2018, 14:44

NAVIGACIJA



Pregledna plošča

■ Prva stran

Strani spletnega mesta


Trenutni predmet


OS

Sodelujoči

Priznanja

Splošno

 O predmetu




















 Potek predmeta

 Samostojno delo

 Viri

 Dostop do Linux sistemov

 Obvestila

-  Forum
-  Opravljanje domačih nalog
-  Naloga 0 - Preizkus oddaje
-  Naloga 1
-  **Naloga 2**
-  proc-demo
-  Naloga 3
-  Primeri izvajanja
-  Točke iz izzivov
-  Dodatne točke
-  Primer izpita: os-1112-1
-  Primer izpita 10. 9. 2014
-  Primer izpita: 22.6.2017
-  Rezultati 1. rok - teorija
-  Rezultati 1. rok - praksa
-  Rezultati 2. rok - teorija
-  Rezultati 2. rok - praksa
-  Rezultati 3. rok - teorija
-  Rezultati 3. rok - praksa
- 5. marec - 11. marec
- 12. marec - 18. marec
- 19. marec - 25. marec
- 26. marec - 1. april
- 2. april - 8. april
- 9. april - 15. april
- 16. april - 22. april
- 23. april - 29. april
- 30. april - 6. maj
- 7. maj - 13. maj
- 14. maj - 20. maj
- 21. maj - 27. maj

Moji predmeti

Predmeti

NASTAVITVE



Skrbništvo predmeta