

Postupne vypracujte jednotlivé podúlohy zadaní.

Okrem manuálových stránok funkcii, bude užitočná aj manuálová stránka `man 7 signal`.

1. zadanie

Cieľom je naprogramovanie asynchrónneho spracovania signálov. Ak počas práce programu príde signál, tak ho obslúži asynchrónne.

- Stiahnite si súbor `signalReceiverAsync.c`. Program v nekonečnej slučke vypisuje bodky do riadku.
- Na začiatok programu doplňte výpis ID procesu.
- Vytvorte funkciu `SignalHandler`, ktorá bude slúžiť ako „obsluha signálu“ (bude zavolaná pri prijatí signálu). Nech funkcia vypíše číslo signálu. Nastavenie volania tejto funkcie je v nasledujúcom bode.
- Nastavte program tak, aby pri prijatí signálu `SIGUSR1` bola zavolaná funkcia `SignalHandler` (nastavte položky štruktúry typu `sigaction` a zavolajte funkciu `sigaction`).
- Spustíte program a na inej konzole pomocou príkazu `kill`, pošlite vášmu programu signál `SIGUSR1`.
- Otestujte tiež poslanie signálu `SIGUSR2` alebo `SIGINT`. Signál `SIGINT` môžete procesu poslať aj stlačením `ctrl+c` (na tej istej konzole, kde ste spustili program).
- Do programu pridajte ešte jedno volanie funkcie `sigaction` tak, aby funkcia `SignalHandler` obsluhovala aj signál `SIGINT`. Teraz už nebude možné ukončiť program stlačením `ctrl+c`. Ukončíte program zaslaním signálu `SIGKILL` (z inej konzoly).
- Pridaním ďalšieho volania funkcie `sigaction` sa pokúste obslúžiť signál `SIGKILL`. Prečo to nie je možné?

2. zadanie

Cieľom je naprogramovanie synchrónneho spracovania signálu. Program bude čakať na signál, po prijatí signálu bude pokračovať vo svojej činnosti.

- Stiahnite si súbor `signalReceiverSync.c`.
- Na vyznačené miesto doplňte volanie funkcie `sigsuspend`, ktorá bude čakať na signál `SIGUSR1`. Správne nastavte masku signálov, ktorá je parametrom funkcie. Program ale ešte nebude fungovať správne.
- Otestujte program poslaním signálu `SIGUSR1` (z inej konzoly). Pri testovaní zistíte, že program pri prijatí signálu nefunguje správne. Program čaká na signál, ale po prijatí signálu nevypíše hlásenie „za `sigsuspend`“, namiesto toho sa vykoná default-ná akcia, ktorá vypíše „User defined signal 1“ a ukončí program. Dôvodom je to, že sme nenastavili obsluhu signálu.
- Preto rovnako ako v predchádzajúcom zadaní vytvorte funkciu `SignalHandler`, ktorá bude slúžiť ako „obsluha signálu“ (bude zavolaná pri prijatí signálu). Program teraz nemusíte testovať, prejdite na nasledujúci bod.

- e) Rovnako ako v predchádzajúcom zadaní nastavte program tak, aby pri prijatí signálu `SIGUSR1` bola zavolaná funkcia `SignalHandler` (nastavte položky štruktúry typu `sigaction` a zavolajte funkciu `sigaction`). Teraz by program mal fungovať.
- f) Doplníte volanie funkcie `sigaction` tak, aby funkcia `SignalHandler` obsluhovala aj signál `SIGINT`. Potom otestujte tieto postupnosti posielania signálov:
 - najprv signál `SIGINT`, potom signál `SIGUSR1`
 - najprv signál `SIGUSR2`, potom signál `SIGUSR1`
- g) Vytvorte kópiu programu a upravte ju tak, aby na signál čakala funkcia `sigwait` namiesto `sigsuspend`. Aké sú rozdiely v implementácii, je potrebné nastavovať „signal handler“? V manuály funkcie `sigwait` si všimnite, že signály na ktoré funkcia čaká, musia byť pri volaní funkcie blokované.

3. zadanie

Cieľom je vytvoriť program v ktorom budú rodičovský a detský proces synchronizované signálom. Dôležité je správne umiestnenie blokovania použitého signálu.

- a) Stiahnite si súbor `signalSynchronizacia.c`.
- b) Pred volanie funkcie `fork`, doplníte zablokovanie signálu `SIGUSR1` (`sigprocmask`).
- c) Doplníte funkciu `childProcess` tak, aby detský proces čakal na signál `SIGUSR1`. Čakanie realizujte funkciou `sigwait` (umiestnite ju na vyznačené miesto).
- d) Doplníte funkciu `parentProcess` tak, aby rodič poslal detskému procesu signál `SIGUSR1`.
- e) Simuláciou oneskorenia (volaním `sleep`) na začiatku funkcie `parentProcess` otestujte, či detský proces čaká na signál.
- f) Podľa manuálu funkcie `sigwait` by pred jej volaním mali byť signály v jej prvom argumente zablokované. Toto sme splnili v jednom z predchádzajúcich bodov. Teraz vykonajte nasledujúci test. Zakomentujte blokovanie signálu `SIGUSR1` v `main-e` pred volaním `fork`. Vo funkcii `childProcess` odkomentujte simuláciu oneskorenia (musí byť na začiatku funkcie). Za simuláciou oneskorenia v detskom procese umiestnite zablokovanie signálu `SIGUSR1`. Tým bude podmienka z manuálu funkcie `sigwait` splnená. Otestujte funkčnosť programu a prejdite na nasledujúci bod.
- g) Napriek splneniu podmienky z manuálu nebude program funkčný. Ak sa detský proces oneskorí tak, že rodičovský proces pošle signál skôr, ako bude v detskom procese zablokovaný signál `SIGUSR1`, tak detský proces pri prijatí signálu vykoná default-né spracovanie signálu. Preto je dôležité, aby signály použité pre synchronizáciu boli blokované už pred volaním `fork`.
- h) Opravte naspäť blokovanie signálu `SIGUSR1` (pôvodné umiestnenie pred `fork` bolo správne).
- i) Z manuálu funkcie `sigprocmask` zistite, ktoré signály nemôžu byť blokované.