

A feladatok megoldása során a jövőben NE használja a pause függvényt! A folyamatok felesleges várakozás nélkül, a lehető leghatékonyabban fussanak le, így a sleep és usleep függvényeket se használja, ha erre külön nem tér ki a feladat szövege (pl. várjon ... másodpercig).

1. Készítsen programot, mely

- A közös kódban egy handlert rendel SIGUSR1, SIGUSR2, SIGRTMIN valamint SIGRTMIN+1 szignálokhoz. A handler írja ki, hogy hányas számú szignál érkezett.
- Indítson egy gyerekfolyamatot.
- A szülőben hozzon létre maszkot, mely csak a SIGUSR2 szignált engedi át, majd e maszkot használva várakozzunk SIGUSR2 érkezésére. SIGUSR2 érkezését követően írja ki, hogy véget ért a felfüggesztés. Végül várja meg míg gyerek terminál, és írja ki, hogy szülő véget ért.
- Gyerek írja ki, hogy 1 mp várakozást követően küldeni fog szülőnek 5 db SIGUSR1, 5 db SIGRTMIN, 5 db SIGRTMIN+1 és 5 db SIGRTMAX szignált váltakozva, majd újabb 1 mp elteltével küld egy SIGUSR2-t. A kiírásban a szignál neve mellett azok számát is tüntesse fel! Ezt követően a kiírásnak megfelelően küldje is el szülőnek a szignálokat! Végül írja ki, hogy gyerek véget ért.

Figyelje és magyarázza meg a beérkezett szignálok sorrendjét és számát!

```
Vár 1 másodpercet, majd küld
5 db SIGUSR1 (10),
5 db SIGRTMIN (34) és
5 db SIGRTMIN+1 (35) és
5 db SIGRTMAX (64) szignált váltakozva,
végül újabb 1 másodperc elteltével küld egy SIGUSR2 (12) szignált.
Gyerek folyamat véget ért!
12 számú szignál érkezett
64 számú szignál érkezett
64 számú szignál érkezett
64 számú szignál érkezett
64 számú szignál érkezett
64 számú szignál érkezett
35 számú szignál érkezett
35 számú szignál érkezett
35 számú szignál érkezett
35 számú szignál érkezett
35 számú szignál érkezett
34 számú szignál érkezett
34 számú szignál érkezett
34 számú szignál érkezett
34 számú szignál érkezett
34 számú szignál érkezett
10 számú szignál érkezett
A program felfüggesztése véget ért!
Szülő folyamat véget ért!
```

2. Készítsen alkalmazást egy gyerek folyamattal. A gyerek rendre írja egy fájlba a számokat 1-től 100-ig, úgy, hogy a fájlt minden kiírt szám előtt létrehozza (vagyis „kinullázza”, tehát egy időben a fájl tartalma csak egyetlen szám lehet)! Minden alkalommal, miután a gyerek módosította a fájl tartalmát, szülő olvassa ki, és írja a képernyőre azt! A folyamatok szinkronizálásához használjon szignálokat!

3. Készítsen alkalmazást két gyerek folyamattal. Az egyik gyerek rendre írja egy fájlba a számokat 1-től 10-ig, úgy, hogy a fájlt minden kiírt szám előtt létrehozza. A másik gyerek rendre írja ugyanabba a fájlba a számokat 20-tól 11-ig visszafelé, úgy, hogy a fájlt minden kiírt szám előtt létrehozza. A két gyerek felváltva írja a számokat a fájlba (1, 20, 2, 19, 3, 18, ...). Minden alkalommal, miután egy gyerek módosította a fájl tartalmát, szülő olvassa ki, és írja a képernyőre azt! A folyamatok szinkronizálásához használjon szignálokat!

4. Készítsen programot, mely

- Tartalmaz egy handlert, mely kiírja, hogy „SIGUSR1 szignál érkezett!” vagy „SIGUSR2 szignál érkezett!” az alapján, hogy melyik szignál hatására futott le a handler.
- A közös kódban blokkolja az összes szignált, majd beállítja, hogy SIGUSR1 szignál legyen figyelmen kívül hagyva, SIGUSR2-höz pedig rendelje hozzá a handlert. Ezután bontsa két szárra a folyamatot.
- A szülő várakozzon SIGUSR1 és SIGUSR2 szignálok valamelyikére. A többi szignált továbbra is blokkolja. Szignál érkezését követően írja ki, „Szülő vége!”
- A gyerek 1 mp várakozás után küldjön SIGUSR1 szignált szülőnek, és írja ki, hogy „SIGUSR1 küldve!”. Újabb 1 mp elteltével küldjön szülőnek SIGUSR2-t, és ennek megtörténtét is írja a képernyőre!

Figyelje meg, melyik szignál hatására mi történik!

5. Készítsen programot, mely

- Tartalmaz egy handlert, mely kiírja, hogy „Handler PID=” és a saját pid-jét.
- A közös kódban létrehoz egy üres szignálkészletet, majd SIGUSR1 szignálhoz rendelje hozzá a handlert. Ezután indítson a szülőből két gyerekfolyamatot. Mindhárom folyamat (szülő és a két gyerek) azonosíthatóan írja ki a saját pid-jét. Ezt követően:
- A szülő 1 mp várakozást követően írja ki, hogy „Szignál küldése Gyerek1-nek!” és küldjön gyerek1-nek egy SIGUSR1 szignált. Újabb 1 mp várakozás után tegye meg ugyanezeket gyerek2-vel is! Ezután várakozzon a közös kódban létrehozott üres szignálkészletet használva egy szignál érkezésére. Végül várja meg, hogy az összes gyerek termináljon.
- A gyerek1 a pid-je kiírását követően szintén a közös kódban létrehozott üres szignálkészletet használva várakozzon egy szignál érkezésére, majd termináljon.
- A gyerek2 tegye ugyanazt, mint a gyerek1, de a szignál érkezését követően ne termináljon, hanem 1 mp várakozást követően írja ki, hogy „Szignál küldése Szülőnek!”, és küldjön SIGUSR1 szignált a szülő folyamatnak.

Figyelje meg, a handler pid-je mikor milyen értékű!

6. Készítsen programot, mely

- Tartalmaz egy handlert, mely kiírja, hogy „Handler - szignál érkezett” és mellé a szignál „leírását”. A következő sorba kiírja, „3 másodperc várakozás!”, és várakozik 3 másodpercet. Végül kiírja, hogy „Handler - szignál feldolgozása befejeződött” és mellé a szignál leírását.
- A közös kódban SIGRTMIN szignálhoz rendelje hozzá a handlert, majd indítson egy gyerekfolyamatot.
- A szülő várakozzon két szignál érkezésére. Amikor beérkezik egy szignál, írja ki, hogy „Szülő x. szignál érkezett” (x helyére az első szignál beérkezése után 1, a második után 2 kerüljön).
- A gyerek 1 másodperc várakozást követően írja ki „Gyerek küld SIGRTMIN és SIGUSR1 szignált szülőnek!” majd küldje el közvetlen egymás után a szignálokat a kiírásnak megfelelően.

Figyelje meg, mi történik, amikor a handler végrehajtása közben érkezik egy másik szignál (nem az, aminek az érkezése handlert elindította), mely nincs handlerhez rendelve vagy ignorálva!

7. Módosítsa a 6. feladatban leírt programot az alábbiak szerint!

- Tartalmaz egy handlert, mely kiírja, hogy „Handler - szignál érkezett” és mellé a szignál „leírását”. A következő sorba kiírja, „3 másodperc várakozás!”, és várakozik 3 másodpercet. Végül kiírja, hogy „Handler - szignál feldolgozása befejeződött” és mellé a szignál leírását.
- A közös kódban SIGRTMIN szignálhoz rendelje hozzá a handlert, majd indítson egy gyerekfolyamatot.
- A szülő várakozzon két szignál érkezésére. Amikor beérkezik egy szignál, írja ki, hogy „Szülő x. szignál érkezett” (x helyére az első szignál beérkezése után 1, a második után 2 kerüljön).
- A gyerek 1 másodperc várakozást követően írja ki „Gyerek küld SIGRTMIN és SIGUSR1 szignált szülőnek!” majd küldje el a szignálokat a kiírásnak megfelelően **úgy, hogy 1 másodpercet vár a két szignál küldése között.**

Figyelje meg, mi történik, amikor a handler végrehajtása közben érkezik egy másik szignál (nem az, aminek az érkezése handlert elindította), mely nincs handlerhez rendelve vagy ignorálva!

8. Módosítsa a 6. feladatban leírt programot az alábbiak szerint!

- Tartalmaz egy handler-t, mely kiírja, hogy „Handler - szignál érkezett” és mellé a szignál „leírását”. A következő sorba kiírja, „3 másodperc várakozás!”, és várakozik 3 másodpercet. Végül kiírja, hogy „Handler - szignál feldolgozása befejeződött” és mellé a szignál leírását.
- A közös kódban SIGRTMIN és SIGUSR1 szignálhoz rendelje hozzá a handler-t, majd indítson egy gyerekfolyamatot.
- A szülő várakozzon két szignál érkezésére. Amikor beérkezik egy szignál, írja ki, hogy „Szülő x. szignál érkezett” (x helyére az első szignál beérkezése után 1, a második után 2 kerüljön).
- A gyerek 1 másodperc várakozást követően írja ki „Gyerek küld SIGRTMIN és SIGUSR1 szignált szülőnek!” majd küldje el közvetlen egymás után a szignálokat a kiírásnak megfelelően. Végül 10 másodperc várakozást követően írja ki, hogy „Gyerek küld SIGUSR1 szignált szülőnek!”, és küldje el a szignált.

Figyelje meg, mi történik, amikor a handler végrehajtása közben érkezik egy másik szignál (nem az, aminek az érkezése handler-t elindította), mely szintén handler-hez van rendelve!

9. Módosítsa a 6. feladatban leírt programot az alábbiak szerint!

- Tartalmaz egy handler-t, mely kiírja, hogy „Handler - szignál érkezett” és mellé a szignál „leírását”. A következő sorba kiírja, „3 másodperc várakozás!”, és várakozik 3 másodpercet. Végül kiírja, hogy „Handler - szignál feldolgozása befejeződött” és mellé a szignál leírását.
- A közös kódban a SIGUSR1 szignálhoz rendelje hozzá a handler-t, majd indítson egy gyerekfolyamatot.
- A szülő várakozzon öt szignál érkezésére. Amikor beérkezik egy szignál, írja ki, hogy „Szülő x. szignál érkezett” (x helyére az első szignál beérkezése után 1, a második után 2 kerüljön, ...).
- A gyerek 1 másodperc várakozást követően írja ki „Gyerek küld 5 db SIGUSR1 szignált szülőnek!” majd küldje el közvetlen egymás után a szignálokat a kiírásnak megfelelően. Végül 20 másodperc várakozást követően írja ki, hogy „Gyerek küld SIGKILL szignált szülőnek!”, és küldje el a szignált.

Figyelje meg, mi történik, amikor a handler végrehajtása közben érkeznek ugyanolyan standard szignálok, mint aminek az érkezése handler-t elindította!

10. Módosítsa a 9. feladatban leírt programot úgy, hogy a standard szignál helyett realtime szignált használ!

Figyelje meg, mi történik, amikor a handler végrehajtása közben érkeznek ugyanolyan realtime szignálok, mint aminek az érkezése handler-t elindította!