

raise



<signal.h>

- **raise**({signal})

A folyamat önmagának küldi el a paraméterben megadott szignált.



Problémák



1. Ki küldte a szignált? (pid)
2. Mi generálta a szignált? (kill? Időzítő?,...)
3. Milyen esemény készítette a küldőt a szignál elküldésére?
(pl. SIGFPE: Nullával osztott?
Túlcsordult? Érvénytelen művelet? ...)
4. Milyen információt akar közölni a szignál segítségével a küldő?
(SIGUSR1 és SIGUSR2 kevés lehet.)



unió típus



A **union** egy speciális típus, mely lehetővé teszi különböző típusú adatok tárolását a memória ugyanazon helyén.

Deklaráció:

```
union azonosító {  
    típus1 adattag1;  
    típus2 adattag2;  
    ...  
}
```

Az unió típusú változó memórafoglalása akkora, amekkora a legnagyobb méretű adattag memóriaigénye.



Összetett handler függvény:

```
void handler(int signo, siginfo_t *info, void *context)
{
    ...
}
```

struct **siginfo_t** *info:

- info->si_code: Mi generálta a szignált? *(ld. Problémák 2.)*
 - SI_USER: kill küldte,
 - SI_QUEUE: sigqueue küldte (ld. később),
 - SI_TIMER: timer küldte (ld. később),
 - SI_MESGQ: üzenetsor küldte (ld. később),
 - ...
- info->si_pid: A küldő PID-je. *(ld. Problémák 1.)*
- info->si_value: A szignállal küldött információ. *(ld. Problémák 3., 4.)*
union sival_t típusú (int vagy pointer):
 - info->si_value.sival_int: egy int típusú érték,
 - info->si_value.sival_ptr: egy mutató.



Összetett handler rendelése szignálhoz



<signal.h>

- struct **sigaction**

Eseményleíró struktúra. Adattagjai:

- sa_sigaction: kezelő eljárás belépési pontjának mutatója, vagy SIG_DFL, SIG_IGN valamelyike.
- sa_mask: sigset_t típusú, a kezelő eljárás végrehajtása közben *további* blokkolandó szignálok készlete.
- sa_flags: jelzések, melyek a szignál viselkedését határozzák meg
 - 0: alap viselkedés
 - SA_SIGINFO: melyik handlert használja?

– ...



Melyik handler-típust rendelje a *ik* szignálhoz?

SA_SIGINFO:

handler paraméterezésének szabályozása

- ha nincs beállítva:
void handler(int signo)
(alap handler)
- ha be van állítva:
void handler(int signo, siginfo_t *info, void *context)
(összetett handler)



Handler rendelése szignálhoz^{ik}

<signal.h>

- **sigaction**({signum}, &{act}, &{oldact})
Megváltoztatja a {signum} szignál által kiváltott eseményt a sigaction típusú {act} struktúrában beállított értékeknek megfelelően.



Információ küldése szignállal

<signal.h>

- **sigqueue**({pid}, {signal}, {value})
{signal} szignált küld {pid} folyamatnak,
és a szignállal küldi {value} union sigval
típusú értéket is.



Egyszeri időzítés



Időzítő (intervall timer): szignált küld amikor az időzítő lejár, de a folyamat végrehajtását nem függeszti fel.

Egyszeri időzítés:

- **alarm({másodperc})**
A megadott idő leteltekor küld egy SIGALRM szignált.



<sys/time.h>

- struct **timeval**

- tv_sec: másodperc,
- tv_usec: milliomod (mikro) másodperc.

A két érték összege jelent egy időintervallumot.

- struct **itimerval**

- it_interval (timeval):
ismételt időzítés értéke
(ha 0, akkor egyszeri időzítés),
- it_value (timeval):
lejáratig hátralévő idő \equiv első időzítés értéke
(ha 0, akkor leállítja az időzítőt).



Ismétlődő időzítés



- **setitimer**({időzítő}, &{új timer}, &{régi timer})
Beállítja és elindítja az ismétlődő időzítőt.

{időzítő}:

- ITIMER_REAL: a ténylegesen eltelt időt méri, és lejáratkor SIGALARM szignált küld,
 - ITIMER_VIRTUAL: az időzítőt indító folyamat végrehajtásának idejét méri, és lejáratkor SIGVTALARM szignált küld.
 - ...
- **getitimer**({időzítő}, &{timer})
Lekérdezi az {időzítő} által meghatározott időzítőt.



Tetszőleges szignál és érték küldése időzítővel



<time.h>

- **timer_create**({clock id}, &{signal event}, &{timer id})
Létrehoz egy új időzítőt és beállítja annak viselkedését.

{clock id}:

- CLOCK_REALTIME: beállítható valós idejű óra
- ...

{signal event} sigevent típusú struktúra:

- sigev_notify: értesítési mód (int):
 - SIGEV_NONE: nem történik semmi az esemény bekövetkezésakor
 - SIGEV_SIGNAL: szignált küld az esemény bekövetkezésekor
 - ...
- sigev_signo: szignál, amit küld (int),
- sigev_value: a szignállal küldött union sigval tip. információ
- ...

{timer id}: a létrehozott timer azonosítója (timer_t)



`<sys/time.h>`

- struct **timespec**
 - tv_sec: másodperc,
 - tv_nsec: milliárdod (nano) másodperc.
- struct **itimerspec**
 - it_interval (timeval): ismételt időzítés értéke,
 - it_value (timeval): lejáratig hátralévő idő
=első időzítés értéke.



Tetszőleges szignál és érték küldése időzítővel



- **timer_settime**({timer id}, {flags}, &{új timer}, &{régi timer})
Beállítja és elindítja a {timer id} azonosítójú időzítőt itimerspec típusú {új timer} értékének megfelelően.
- **timer_gettime**({timer id}, &{aktuális érték})
Lekérdezi {timer id} azonosítójú időzítőt.
- **timer_delete**({timer id})
Törli a {timer id} azonosítójú időzítőt.

