Semnale

- Shell $
- $ kill pid → transmiterea semnalelor
  - se trimite un semnal: SIGTERM – terminarea proc.
- $ kill -nr pid → ne uităm în <signal.h> și vedem nr. semnalului.
  ↳ nr sem.
- $ kill -9 pid → procesul este sigur ~~terminat~~
  → sem. "sigkill" → "-9", nu se poate modifica.
- $ kill -sufix pid (ex: kill -term; kill -9 = kill -kill etc)

• int kill(int pid, int sig) → ret. 0 – succes
  → ret -1 – insucces ← nu ex proc cu pid; există proc. cu pid dar nu sunt proprietar; sig este ilegal/invalid (ex: am 32 sem, am trimis sem 77)

• Categorii de semnale

(1) Tastări unor caractere speciale ~~de terminalul de~~ de terminalul de control

• $stty -a → afișează o serie de informații care rep. parametrizarea leg. ~~dintre~~ ~~procese~~ utilizator și terminal.

• SIGINT – produs în urma apăsării de către utilizator a tastei intr
  → are val CTRL+C → oprește comanda.

  SIGINT intr = ^C

• SIGKILL kill = ^\ (CTRL+\) – terminarea procesului
  – face un fișier (core) în care scrie contextul procesului, pt depanare.

• SIGTSTP susp = ^@ (CTRL+@) – proc nu se mai termină și intră în stare stopat.

(2) În urma "prostiilor" făcute de noi în program.

• SIGSEGV – se af pe ecran un ~~sem~~ mesaj – segmentation fault (violation)
  – am încercat să scriem la o adresă ilegală
  – creată core

• SIGILL – se af- memory fault. // pointeri către funcții invalide
– excepția care – putem muta fix unde vrem noi

Pointeri către funcții

pointeri <
- int *x; — variabilă (are val aleatoare)
- int tab(10); — constantă
- x = tab; // este corect
- tab = x; // NU este corect
- int f(int a..)
- u = f(5); → var u primește f(5)
- f(5); // declanșează ex. celui f
- f; // pointer constant în adr de unde începe execuția lui f
- int (*p)(---) // p este o variabilă care memorează adr unei funcții
- int *p(--) // returnează int *
- if(---) // f, g - funcții def de noi
  p = f;
  else p = g;

La examen → (*p)(---) // execuția funcției alese ex anterior (ori f, ori g)

• void qsort(void *tab, // sortează un tablou după niște criterii
int nr, int size, int (*comp)(void *a, void *b))

- tab – tabel cu elem. de sortat
- nr – nr elem din tablou
- size – se dă cu sizeof (dim fiecărui elem)
- func. de comparare (ret.): <0 a<b; =0 a>b; >0 a>5

• SIGBUS – eroare de magistrală (de "autobuz")
pe RISC – citim sau vrem de la o adr. nealiniată. ⇒ SIGBUS.

```
struct mystruct
{
  char x;
  int y;
}
char buf[100];
```

- la struct - dc punem al doilea arg null - îmi dă cea mai bună aliniere

  la fel malloc, calloc. - aloc. dinamică

③ Semnale nemodificabile

- SIGKILL - se pune întotdeauna pe 9
- SIGSTOP - duce la punerea proc în stare stopat
- SIGCONT - duce la continuarea unui proc în stare stopat.

Blocarea și deblocarea semnalelor

Blocarea semnalelor → sem nemodificabile nu pot fi blocate

sigset_t

- int sigemptyset (sigset_t*p) - ret. numai 0 - init la mult vidă
- int sigfillset (sigset_t*p) - pune toți biții pe 1 - inițializează la mult. totală
- int sigaddset (sigset_t*p, int sig) - adaugă la mult. p un elem sig.
  - pune un bit pe 1
- int sigdelset (sigset_t*p, int sig) - pune elem sig pe 0 bitul
- int sigismember (sigset_t*p, int sig)
  - ret 0 - sig ∉ mult.
  - ret 1 - sig ∈ mult.

Deblocare / Blocare definitivă

- int sigprocmask (int op, sigset_t*p_nou, sigset_t*p_vechi)

~~SIGSETMASK~~

SIG_SETMASK - p_nou este noua mască de semnal
- blochează semn. din p_nou

SIG_BLOCK - la sem. deja blocate adaugă cele din p_nou

SIG_UNBLOCK - toate semnalele din p_nou vor fi deblocate (dacă nu erau deja deblocate)

- Handlere de semnal

void handler (int sig)

- Instalarea handler

void (*signal) → returnează un pointer către o funcție care returnează void

void (*signal)(int sig, void(*handler)(int sig1))(int sig 2)

sem pe care vrem să îl modif ← void handler

signal(SIGINT, handler)

SIG_DFL - DEFAULT (sem din default)
SIG_IGN - IGNORE (ignorarea sem)

Struct sigaction
{
  void (*sa_handler)(int sig);
  sigset_t sa_mask; → masca de semnale blocate pe durata execuției handlerului
  int sa_flags;
}

int sigaction(int sig, struct sigaction *p_nou, struct sigaction *p_vechi)

SA_RESETHAND - sa inst. comportamentul cu sigaction
- după aceea se restaurează cu opțiunea acesteia