UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Sisteme de operare

18. Semnale

Întreruperea unui proces în Unix

- Un proces este întrerupt prin recepția unui semnal
- Când se primește un semnal:
  - starea procesului este salvată
  - se execută o rutină de tratare a semnalului
  - se continuă execuția procesului
- Sunt rezolvate asincron față de fluxul de execuție al procesului

Semnale

- Sunt identificate prin numere
  - se recomandă folosire de macro-uri
  - SIGKILL – 9, SIGSEGV – 15
- Tipuri de semnale
  - asincrone – primite de la alte procese (programatic sau de la utilizator)
  - sincrone – procesul a generat o excepție (acces invalid, împărțire la 0, etc.)
- Ce se întâmplă la primirea unui semnal?
  - rularea unui handler
  - ignorarea semnalului
  - semnalul este blocat (fiecare proces are o mască de semnale)
  - comportament implicit (terminare sau ignorare) (SIGKILL și SIGSTOP nu pot fi ignorate)

Problemele folosirii semnalelor

- apelurile de sistem blocante pot fi întrerupte (errno = EINTR)
- lucrul cu semnale clasice este susceptibil la condiții de cursă
  - pot exista condiții de cursă dacă se partajează variabile între handler-ul de semnal și codul programului
- semnalele clasice pot fi pierdute
- nu se pot apela funcții non-reentrante (malloc)

Semnale de timp real

- Numerele cuprinse între SIGRTMIN (32) și SIGRTMAX (63)
- Nu au roluri predefinite
- Se pot pune într-o coadă mai multe semnale
- Semnalele cu număr mai mic au prioritate mai mare
- Se garantează ordinea în cazul transmiterii repetate a aceluiași semnal

Întreruperea unui proces în Windows

- Se datorează recepționării unei excepții
- Când un proces primește o excepție
  - starea procesului este salvată
  - se execută o rutină de tratare
  - se continuă execuția procesului
- Cauzele excepțiilor
  - o cerere de întrerupere a procesului
  - accesarea unei locații invalide de memorie
  - împărțire la zero

APC

- Asynchronous Procedure Call
- Asociate per thread; fiecare thread are o coadă de APC
- APC-urile se execută în momentul în care thread-ul curent intră într-o stare alertabilă
  - `WaitForSingleObjectEx`

Studiu de caz: comanda nohup

```
$ nohup btdownloadheadless --responsefile lin-
  kernel.torrent &> /dev/null &
```

- Comanda este pornită în background
- Procesul ignoră semnalul SIGHUP
- La încheierea unei sesiuni shell se transmite
  - SIGQUIT proceselor din foreground
  - SIGHUP proceselor din background
- Similar: disown (bash), dtach, screen