Huts egiten badu => -1 ---> perror(const char\*s)-k mezu bat sortzen du

**Erroreak:**

* 0:Success
* 1: Operation not permitted
* 2: No such file or directory
* 3: No such proccess
* 4: Interrupted system call
* 5: I/O error
* ...

**Etenak:**

* Hardware: Gailua IRQ bat bidaltzen dio PUZari honela PUZak egiten hari dena usten du etena kudeatzeko
* Software: Kernelak eskeintzen dituen zerbitzuei edo sistema deien deitzeko, baita gure prozesuak sartutako eszepzioei erantzuteko.

**PROZESUEN KONTROLERAKO SISTEMA-DEIAK**

**1. Prozesuen identifikazioa:**

* getpid: Prozesuaren identifikadorea
* getppid: Gurasoaren identifikadorea
* getuid: jabearen identifikadorea

**2. Prozesuen sorrera:**

* fork: Prozesu berri bat sortu, deitzailearen klona. Zer bueltatu? 0 (umeari), umearen pid (gurasoari) edo -1 (errorea)
* exec: Exekutatzen ari den programa aldatu, -1 bueltatu errorea badago

**3. Prozesuen bukaera/sinkronizazioa:**

* exit(int egoera): Prozesuen bukaera kontrolatuta, ez du ezer bueltatzen
* wait(int \*egoera): Deitzailearen umea bukatu arte gelditzen da, -1 edo umearen ida bueltatzen du
* kill(int pid, int SIGKILL): SIGKILL seinalea prozesuari bidaltzen dio

**4.Seinaleen kontrola:**

* kill(int pid, int SIGKILL): SIGKILL seinalea prozesuari bidaltzen dio
* pause(): Seinale bat jaso arte geratzen da
* signal(int seinale,void funtzioa()): Seinalea jasoz gero exekutatuko da

**5. Denbora kontrola:**

* alarm(unsigned int seg): Sartutako zenbakia segundutan pasa ostean seinalea bidaltzen du
* sleep(unsigned int seg): Seg segundutan SEak prozesua lotan uzten du
* time(time\_t \*t): 1970/01/01tik igarotako denbora bueltatzen du.

**KOMUNIKAZIORAKO ETA SINKRONIZAZIORAKO MEKANISMOAK**

Komunikatzeko mekanismo ezberdinak daude: Seinaleen bidez, fitxategien bidez, fitxategi bereziak (tutuak), aldagai partekagarriak, ...

**Prozesuak vs hariak:**

* Prozesuak:

Hurrengo ingurune konputazionaletan egikaritzen dira:

* + Testu segmentua: Programaren kode bitarra
  + Datuen segmentua: Aldagai globalak eta estatikoak
  + Zabaldutako fitxategiak eta beste baliabide batzuk: inpresora, teklatua,...
  + Egikaritze motorra:
    - -Pila segmentua: Como en KE
    - -SEko testuinguru informazioa: Como las etenas de KE

SEko prozesuek hari anitz izan ditzakete

Hari anitzeko prozesuetan informazioa partekatzen da baina bakoitzak bere pila eta testuinguru propioak ditu

Prozesu klasikoak: Prozesu bakoitza hari bakarra da, ez da ezer partekatzen

Hari anitzeko prozesuetan aldi berean hainbat hari aldi berean programatzen egon daitezke, nahiz eta datu eta kode bera eduki

**Hariaren ezaugarriak:**

* Prozesuak baino arinagoak sortzen eta bukatzen dira
* PUZ bat egonez gero haien arteko txandaketa azkarragoa da, soilik pila eta testuingurua gordetzen direlako
* Memoria gune bera konpartitzen dute, ondorioz haien arteko komunikazioa errazten da
* Ikusteko: ps -eLf
* pthread\_create: Hari bat sortzeko
* pthread\_join(thread\_id): Lo geratzen da thread\_id haria bukatu arte
* pthread\_exit(status): Haria bukatzeko

Exekuzioaren arabera emaitzak ezberdinak ez izateko, SK zein den detektatu behar da. Eta sarrera eta irteera protokolo bat ezarriko dugu. Hurrengo propietateak aztertuz:

* Elkarrekiko esklusioa:
  + Prozesu bat baino gehiago, ezin da SKan aldi berean egon.k graduko baliabidea partekagarria baldin bada, k prozesu baino gehiago ezin izango dira aldi berean bere SKan egon. Honen ondorioz prozesu bat instrukzio hau exekutatzerakoan beste batek ezin du
* Progresio finitua:
  + SKan ez dago prozesurik, SKan sartu nahi diren artean erabaki behar da ze prozesu sartu behar den.
* Itxarote mugatua:
  + Prozesu bat ezin da etengabe egon SKan sartzeko
* Elkar blokeaketarik ez:
  + Bi prozesu elkartzen dira baliabideak partekatzeko, blokeatuta geratu daiteke biak aldi berean behar duten baliabide bat ez badute

**Komunikatzeko metodoak:**

* Sekzio kritikoa: Kode zati bat, atzipen esklusiboa duen baliabide partekatu kudeatzeko, momentu berean ezin dira elkarbanatu X prozesu baino gehiagotan (normalean X = 1)
* SKaren kodea egikaritzeko orduan, momentu oro prozesu bakarra kontrolatzeko sarrera eta irtera protokoloak daude
* S/I protokoloak definitzeko metodo ezberdinak daude, itxarote aktiboa (inkesta) edo blokeatze bidezko itxarotea (Semaforoak)