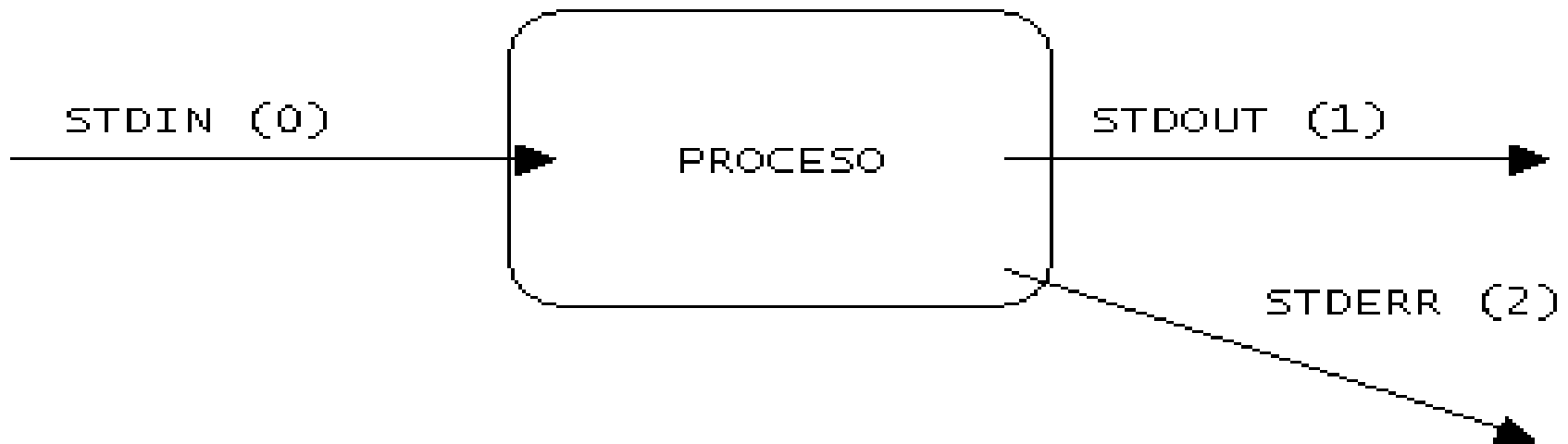


Berbideratzea

Kepa Bengoetxea Kortazar
email: kepa.bengoetxea@ehu.es

Berbideratzea

- Prozesu bakoitza fitxategi deskribatzaileen taula bat erabiltzen du, fitxategi horiek erraz erabili ahal izateko. Hasieran taula horretako 0, 1 eta 2 sarrerak, STDIN, STDOUT eta STDERR fitxategiengatik daude beterik, hau da, sarrera estandarra, irteera estandarra eta erroreen irteera estandarra:



Berbideratzea

- Fitxategi deskribatzaileen taula , ostatu bat bezala ikus genezake, lehenengo 3 gelak STDIN, STDOUT eta STDERR bezeroengatik beteta egongo dira. Bezero gehiago etortzen direnean (artxibo berriak irekitzerakoan), hurrengo gelatan sartuko dira. Horrela, prozesua hasi egiterakoan fitxategi bat irekitzen badugu, bere fitxategi deskribatzailea 2ren hurbil izango da. “Ostatu” honetan beti ematen zaie bezero berriei “gela” baxuenak. Hau kontutan euki beharko dugu etorkizuneko programetan.

Berbideratzea

- Edozein prozesu bat exekutatzerakoan sistemak 3 artxibo irekitzen ditu: stdin, stdout eta stderr.
- Komando-interpretatzailearentzat tekklatura apuntatzeko: stdin , eta monitoreari: stdout eta stderr. 3 fitxategi deskribatzaile hauek zenbaki batez adieraz dezakegu: stdin 0 ,stdout 1 eta stderr 2 gatik.
- Unixeko komando gehienak, datuak hartzeko sarrera estandarra erabiltzen dute eta datuak zuzenak eta erroreak kanporatzeko irteera estandarra, hau da, pantaila erabiltzen dute. Adb: cat
- Era honetan errorearen irteera estandarra ere berbideratu dezakagu pantailan ez ateratzeko.

Berbideratzea

- Hurrengo operazioak egin daitezke:
 - Stdout fitxategi batera berbideratu: `ls -l 1> kaixo.txt` (defektuz, `ls -l > kaixo.txt`)
 - Stderr fitxategi batera berbideratu: `grep da * 2> grep_okerrak.txt`
 - Stderr eta stdout fitxategi batera berbideratu: `grep da * &>grep_okerrak.txt`
 - Stdout stderr batera berbideratu: `grep da * 1> &2`
 - Stderr stdout batera berbideratu : `grep da * 2> &1`

probatzeko:(`ls /bin/ 1>&2`)>a.txt, non a.txt hutsik dago, irteera estandarra errore estandarrera berbideratu da-eta.

Berbideratzea

Sarrera Berbiredatzen:

ispell komandoa sarrera estandarretik STDIN (Teklatua) sartzen diren erroreak aurkitzeko balio du.

sudo apt-get install spell

ispell

Tis is a word

Ctrl+d

gedit testua.txt ->Tis is a word

word: how about: Dis, His, Is, ...

ok

ok

Berbideratzea

Irteera Berbideratzen:

• *less fitx_izen* komandoa, *fitxategiaren* edukiera pantailan erakusten du (*Irteera Standard-STDOUT*), irteera *fitxategi* batera *berbideratu* nahi badugu.

Adb:]\$less fitx_izen >fitx_izen_berria

• *Eta irteera berbideratzea baina eransketa moduan; hau da, zegoen fitxategi batean gehituko da komandoaren irteera.*

Adb:]\$less fitx_izen >> fitx_izen_berria

Pipea edo kanalizazioa

- Zer dira eta zergatik erabili nahiko ditugu?
 - Pipeak edo kanalizazioa, ezkerreko komandoaren irteera eskuineko komandoaren sarrerara bideratzea. Komando baten irteera hurrengoaren sarrera izan behar denean erabiliko da, horrela fitxategi iragankor bat sortzeko eta ezabatzeko beharra saihesten baita.
 - Adb: `ls -l | less` edo `dmesg | grep "sd"`

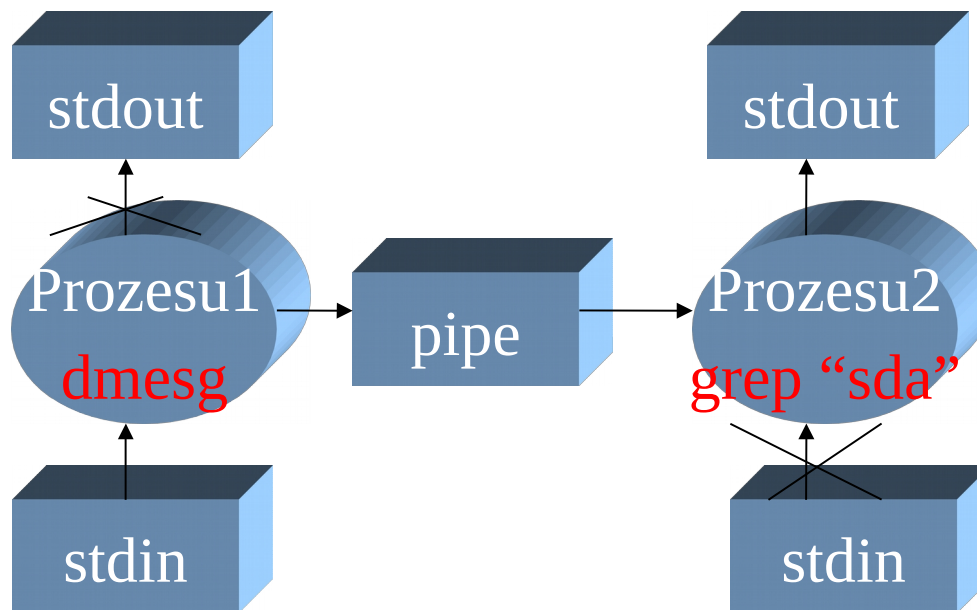
bestela hau egin beharko zen

`$ls -l > fitx` eta `less fitx`

`$dmesg > kernelog.txt` eta `grep "sd" kernelog.txt`

Pipea edo kanalizazioa

Adibidez: `dmesg | grep "sda"`



Some limits?

- The maximum number of processes available to a single user?
 - the number of processes in execution: `ps aux | wc -l`
->226
 - The maximum number of processes available to a single user: `ulimit -u` → 63652 prozessu
 - File descriptor limits per process: `ulimit -n` → 1024
(1021+stdin+stdout+stderr)
 - $63652 \times 1024 = 65179648$
- The maximum number of open file descriptors: `cat /proc/sys/fs/file-max` → 1625478 .To change it: `sudo sysctl -w fs.file-max=65179648`
- the number of processes in execution: `ps aux | wc -l` ->226