

# Εισαγωγή στα Λειτουργικά Συστήματα



SET ΔΙΑΦΑΝΕΙΩΝ 8

ΔΙΕΡΓΑΣΤΕΣ

ΑΝΤΩΝΗΣ ΣΙΔΗΡΟΠΟΥΛΟΣ

# διεργασίες

2

- Μια **διεργασία (process)** είναι ένα στιγμιότυπο ενός προγράμματος που βρίσκεται σε εκτέλεση.
- Κάθε διεργασία έχει τον δικό της «χώρο» μνήμης (memory space).
- Μερικές φορές χρησιμοποιείται και ο όρος Task.  
πχ:
  - ο Task Manager των Windows μας δείχνει τις ενεργές διεργασίες.
  - Λέμε “multi-tasking” Λειτουργικό Σύστημα.
  - Ο όρος Task φαίνεται να τείνει να εγκαταλειφθεί πλέον και χρησιμοποιείται ο όρος process.

# Διεργασίες & Λειτουργικό

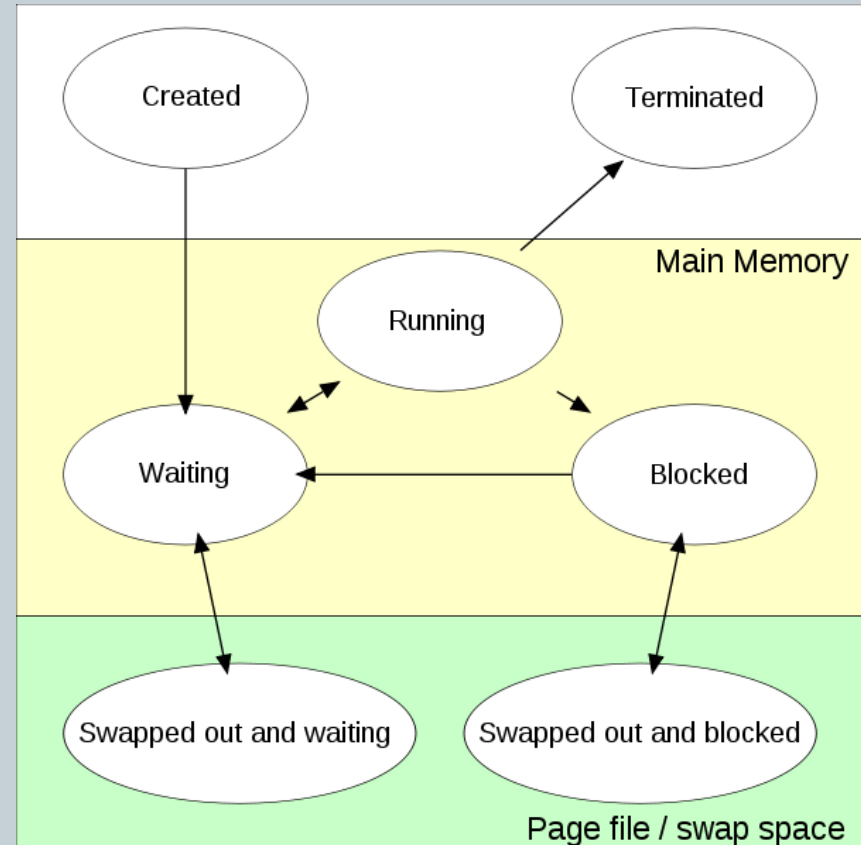
3

- Μια από τις κύριες αρμοδιότητες του ΛΣ είναι η διαχείριση των διεργασιών.
- Το ΛΣ δημιουργεί, «καταστρέφει», συγχρονίζει τις διεργασίες.
  - πχ: όταν ο χρήστης πληκτρολογεί μια εντολή (πχ την ls) στο κέλυφος, τότε το κέλυφος δίνει την οδηγία στον πυρήνα να δημιουργήσει μια νέα διεργασία από το εκτελέσιμο αρχείο “ls”. Ο πυρήνας θα κάνει τις απαραίτητες ενέργειες ώστε να δημιουργηθεί η νέα διεργασία.
  - Σε ένα λειτουργικό σύστημα Unix/Linux, ο πυρήνας θα αποφασίσει ποια διεργασία θα προωθηθεί στην CPU και για πόσο χρόνο.

# Καταστάσεις Διεργασιών

4

- Στα περισσότερα λειτουργικά συστήματα έχουμε τις εξής καταστάσεις
  - Created – μόλις δημιουργήθηκε.
  - Waiting – η διεργασία είναι σε αναμονή για την CPU
  - Running – η διεργασία καταλαμβάνει την CPU και εκτελείται
  - Blocked – η διεργασία ζήτησε έναν πόρο και περιμένει απάντηση (πχ: να διαβάσει ένα τμήμα ενός αρχείου)
  - Swapped out – η μνήμη που αντιστοιχεί στην διεργασία έχει αποθηκευτεί στην μνήμη swap (στον δίσκο)



# Διεργασίες & Λειτουργικό

5

- Ο πυρήνας διατηρεί στην μνήμη του μια λίστα με τις διεργασίες.
- Για κάθε διεργασία αποθηκεύονται διάφορες πληροφορίες:
  - Ιδιοκτήτης
  - Ιδιοκτήτρια ομάδα
  - PID (Process ID – ένας μοναδικός αριθμός που είναι η ταυτότητα της διεργασίας)
  - PPID (Parent PID – Η ταυτότητα της γονικής διεργασίας)
  - STIME (Start Time – Η ώρα εκκίνησης της διεργασίας)
  - TIME (Πόσο χρόνο έχει καταλάβει η διεργασία στην CPU)
  - TTY (Σε ποιο τερματικό τρέχει η διεργασία)
  - CMD (Ποια ήταν η εντολή που δημιούργησε την διεργασία)
  - CWD (Ποιος είναι ο κατάλογος εργασίας της διεργασίας)
  - ENV (Ο πίνακας με τις μεταβλητές περιβάλλοντος της διεργασίας – προσοχή ΌΧΙ τις μεταβλητές του προγράμματος).
  - STATUS (Η κατάσταση της διεργασίας).
  - Priority (Προτεραιότητα – μπορεί να την ορίσει και ο χρήστης)
  - κ.α.

# ps

6

- Η εντολή `ps` χρησιμοποιείται για να δούμε τις διεργασίες ή για να δούμε πληροφορίες γι' αυτές.
- Syntax: `ps [options]`
- προσοχή: κάποια options της `ps` σηματοδοτούνται με το «-» ενώ κάποια όχι.

```
asidirop@aetos:~$ ps xlw
F  UID    PID  PPID  PRI  NI     VSZ   RSS  WCHAN  STAT TTY          TIME COMMAND
0  1143   2708 31892   25   5    8424   744  -      RN+  pts/0        0:00 ps xlw
5  1143  31887 31832   25   5   97344  1648  ?      SN   ?            0:00 sshd: asidirop@pts/0
0  1143  31892 31887   25   5   19828  2096  -      SNs  pts/0        0:00 -bash
asidirop@aetos:~$ ps xuw
USER      PID %CPU %MEM    VSZ   RSS  TTY      STAT START   TIME COMMAND
asidirop  3214  0.0  0.0  16840  1036 pts/0    RN+  20:13   0:00 ps xuw
asidirop 31887  0.0  0.0  97344  1648 ?        SN   18:44   0:00 sshd: asidirop@pts/0
asidirop 31892  0.0  0.0  19828  2096 pts/0    SNs  18:44   0:00 -bash
asidirop@aetos:~$
```

- Συνήθη options για την ps
- ps uxww  
Το ww θα εμφανίσει ολόκληρη την εντολή και τα ορίσματά της  
Το x εμφανίζει όλες τις διεργασίες του τρέχοντος χρήστη  
Το u δηλώνει την μορφοποίηση του αποτελέσματος

```
asidirop@aetos:~$ ps uxww
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
asidirop	439	2.9	1.1	149988	58444	?	SN	19:45	1:01	spamd child
asidirop	4041	0.0	0.0	11116	1172	?	SNs	20:15	0:00	/bin/sh -c (cd /home/staff/ekta
asidirop	4046	0.0	0.0	11116	632	?	SN	20:15	0:00	/bin/sh -c (cd /home/staff/ekta
asidirop	4049	0.0	0.0	11124	1348	?	SN	20:15	0:00	/bin/sh ./r2e run
asidirop	4057	42.0	0.4	134488	22488	?	SNl	20:15	2:28	python rss2email.py feeds.dat ru
asidirop	7134	0.0	0.0	14956	876	?	Ss	20:20	0:00	procmail -f asidirop@aetos.it.te
asidirop	7154	0.0	0.0	14956	156	?	S	20:20	0:00	procmail -f asidirop@aetos.it.te
asidirop	7155	0.0	0.0	23684	1484	?	S	20:20	0:00	/usr/bin/spamc
asidirop	7160	0.0	0.0	16872	1144	pts/0	RN+	20:20	0:00	ps uxww
asidirop	31887	0.0	0.0	97344	1648	?	SN	18:44	0:00	sshd: asidirop@pts/0
asidirop	31892	0.0	0.0	19840	2120	pts/0	SNs	18:44	0:00	-bash

- Συνήθη options για την ps
- ps lxww  
Το ww θα εμφανίσει ολόκληρη την εντολή και τα ορίσματά της  
Το x εμφανίζει όλες τις διεργασίες του τρέχοντος χρήστη  
Το l δηλώνει την μορφοποίηση του αποτελέσματος

```
asidirop@aetos:~$ ps lxww
```

F	UID	PID	PPID	PRI	NI	VSZ	RSS	WCHAN	STAT	TTY	TIME	COMMAND
4	1143	4041	4034	25	5	11116	1172	-	SNs	?	0:00	/bin/sh -c (cd /home/staff/
1	1143	4046	4041	25	5	11116	632	-	SN	?	0:00	/bin/sh -c (cd /home/staff/
0	1143	4049	4046	25	5	11124	1348	-	SN	?	0:00	/bin/sh ./r2e run
0	1143	4057	4049	25	5	134488	22488	-	SNl	?	3:39	python rss2email.py feeds.da
0	1143	8967	31892	25	5	8456	744	-	RN+	pts/0	0:00	ps lxww
5	1143	31887	31832	25	5	97344	1648	?	SN	?	0:00	sshd: asidirop@pts/0
0	1143	31892	31887	25	5	19840	2120	-	SNs	pts/0	0:00	-bash



- Συνήθη options για την ps

- ps uw -e

Το w θα εμφανίσει σχεδόν ολόκληρη την εντολή και τα ορίσματα της (όσα χωράνε στο τερματικό)

Το -e εμφανίζει όλες τις διεργασίες όλων των χρηστών

Το u δηλώνει την μορφοποίηση του αποτελέσματος

```
asidirop@aetos:~$ ps uw -e
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	8404	160	?	Ss	Apr03	0:43	init [2]
root	2	0.0	0.0	0	0	?	S	Apr03	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	Apr03	0:26	[ksoftirqd/0]
root	6	0.0	0.0	0	0	?	S	Apr03	0:05	[migration/0]
root	7	0.0	0.0	0	0	?	S	Apr03	0:02	[watchdog/0]
root	8	0.0	0.0	0	0	?	S	Apr03	0:05	[migration/1]
root	10	0.0	0.0	0	0	?	S	Apr03	0:21	[ksoftirqd/1]
root	12	0.0	0.0	0	0	?	S	Apr03	0:01	[watchdog/1]
root	13	0.0	0.0	0	0	?	S	Apr03	0:04	[migration/2]
.....										
root	31832	0.0	0.0	97344	3876	?	SNs	18:44	0:00	sshd: asidirop [priv]
asidirop	31887	0.0	0.0	97344	1648	?	SN	18:44	0:00	sshd: asidirop@pts/0
asidirop	31892	0.0	0.0	19840	2120	pts/0	SNs	18:44	0:00	-bash
psarnik	31944	0.0	0.0	19600	1652	?	S	20:11	0:00	imap [psarnik ::ffff:79.103.211.1]
www-data	32457	0.0	0.1	320604	7440	?	Sl	19:45	0:01	/usr/sbin/apache2 -k start
www-data	32488	0.0	0.1	320496	7308	?	Sl	19:45	0:01	/usr/sbin/apache2 -k start
www-data	32489	0.1	0.1	386072	7416	?	Sl	19:45	0:02	/usr/sbin/apache2 -k start
psarnik	32723	0.0	0.0	19680	1724	?	S	20:12	0:00	imap [psarnik ::ffff:79.103.211.1]

# pstree

10

- Αντίστοιχη εντολή με την ps, μόνο που μας δείχνει τις διεργασίες σε μορφή δέντρου (με βάση τη σχέση γονέα-τέκνου)
- όλες οι διεργασίες έχουν ως πρόγονο την “init”. Η init είναι η βασική διεργασία του πυρήνα. Στις νεότερες εκδόσεις linux έχει αντικατασταθεί με την “systemd”.

```
asidirop@antonis-PC:~$ pstree
init--NetworkManager--dhclient
                        |_2*[{NetworkManager}]
--VBoxSVC--VirtualBox--24*[{VirtualBox}]
                        |_11*[{VBoxSVC}]
--VBoxXPCOMIPCD
--VirtualBox--5*[{VirtualBox}]
--acpid
--apache2--6*[apache2]
--atd
--avahi-daemon--avahi-daemon
--avgd--avgavid--3*[{avgavid}]
        |_avgoad--avgscand--6*[{avgscand}]
        |       |_6*[{avgoad}]
        |_avgsched--3*[{avgsched}]
        |_avgtcpd--avgscand--6*[{avgscand}]
        |       |_8*[{avgtcpd}]
        |_5*[{avgd}]
--bamfdaemon
--3*[bonobo-activati--2*[{bonobo-activat}]]
--chrome--chrome
        |_chrome--{chrome}
        |_chrome--2*[{chrome}]
        |_24*[{chrome}]
--chrome--8*[chrome--2*[{chrome}]]
        |_2*[chrome--3*[{chrome}]]
--2*[clock-applet--{clock-applet}]
--console-kit-dae--64*[{console-kit-da}]
--cpufreq-applet--{cpufreq-applet}
--cron
--cupsd
--5*[dbus-daemon]
--3*[dbus-launch]
```

# pstree

11

- Η pstree μπορεί να δεχθεί διάφορα ορίσματα που θα καθορίσουν τον τρόπο εμφάνισης των αποτελεσμάτων.

```
asidirop@antonis-PC:~$ pstree -u
init--NetworkManager--dhclient
                        |
                        |_2*[{NetworkManager}]
--VBoxSVC(asidirop)--VirtualBox--24*[{VirtualBox}]
                        |
                        |_11*[{VBoxSVC}]
--VBoxXPCOMIPCD(asidirop)
--VirtualBox(asidirop)--5*[{VirtualBox}]
--acpid
--apache2--6*[{apache2(www-data)}]
--atd(daemon)
--avahi-daemon(avahi)--avahi-daemon
--avgd--avgavid--3*[{avgavid}]
        |
        |_avgoad--avgscand--6*[{avgscand}]
        |           |
        |           |_6*[{avgoad}]
        |           |_avgsched--3*[{avgsched}]
        |           |_avgtcpd--avgscand--6*[{avgscand}]
        |           |           |
        |           |           |_8*[{avgtcpd}]
        |           |_5*[{avgd}]
--bamfdaemon(asidirop)
```

# Διεργασίες

12

- Κάθε διεργασία έχει μια γονική διεργασία.
- Η γονική διεργασία είναι η διεργασία που έδωσε «εντολή» στο Λειτουργικό να δημιουργηθεί η νέα.
- Η ταυτότητα κάθε διεργασίας είναι το PID (Process ID).
- Για κάθε διεργασία υπάρχει η πληροφορία, PID, PPID (Parent Process ID).
- Δεν υπάρχει καμία διεργασία χωρίς γονέα
- Μόνο η process 0 (μηδέν) δεν έχει γονέα. Η process 0 είναι η πρώτη διεργασία που εκτελείται με το boot του συστήματος.
- Η process 0 αναφέρεται και ως “swapper” ή “sched”. Είναι ο πυρήνας και δεν εμφανίζεται στην λίστα των διεργασιών.

# Διεργασίες

13

- Τι γίνεται αν ο γονέας μιας διεργασίας τερματίσει πριν τα τέκνα? Τα τέκνα θα έχουν PPID μια διεργασία που δεν υπάρχει?
- ΌΧΙ
- Η συμπεριφορά σε αυτήν την περίπτωση εξαρτάται από την έκδοση του πυρήνα. συνήθης τακτική είναι να ορίζεται ο γονέας στις «ορφανές» διεργασίες η διεργασία 1.

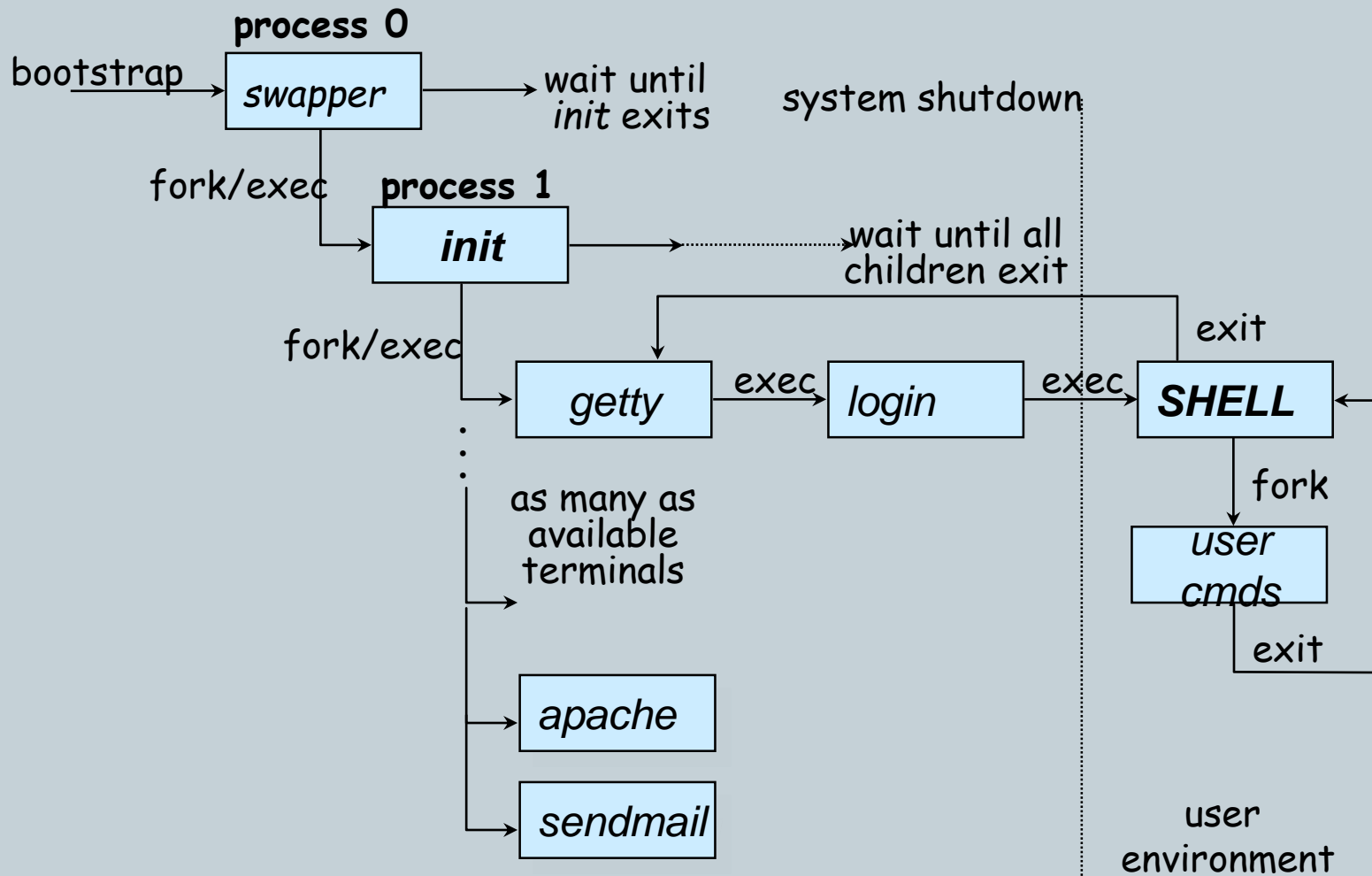
# Διεργασίες

14

- Κατά την εκκίνηση ενός υπολογιστή:
  - Ο boot loader ξεκινά τον πυρήνα (PID 0)
  - Ο πυρήνας ξεκινά την διεργασία init, η οποία ανάλογα με τις ρυθμίσεις θα ενεργοποιήσει όλες τις υπόλοιπες διεργασίες στο σύστημά μας.
- όταν τερματίσει η init, τότε συμβαίνει shutdown – το κλείσιμο του λειτουργικού.

# Αρχικοποίηση συστήματος

15



# fork / exec

16

- 2 τρόποι για να δημιουργηθεί μια νέα διεργασία:
  - **fork:** όταν μια διεργασία κάνει fork (καλεί την αντίστοιχη συνάρτηση του πυρήνα) τότε δημιουργείται ένα ακριβές αντίγραφο της τρέχουσας διεργασίας. Το μόνο διαφορετικό που έχουν οι 2 διεργασίες είναι τα PID,PPID
    - ✦ έστω η διεργασία A με PID=9, PPID=3
    - ✦ κάνει fork και δημιουργείται μια νέα, έστω η B που είναι ακριβές αντίγραφο της A.
    - ✦ η B θα έχει PPID=9 και PID ένα νέο μοναδικό αριθμό (θα μπορούσε να είναι το 10)
    - ✦ όλα τα υπόλοιπα στοιχεία των 2 διεργασιών παραμένουν ίδια (owner, group, cwd, environment, etc)



# fork / exec

17

- 2 τρόποι για να δημιουργηθεί μια νέα διεργασία:
  - **exec:** όταν μια διεργασία κάνει exec με όρισμα ένα εκτελέσιμο πρόγραμμα (καλεί την αντίστοιχη συνάρτηση του πυρήνα) τότε δημιουργείται μια νέα διεργασία που τρέχει το εκτελέσιμο. Η νέα διεργασία κληρονομεί τα ΠΑΝΤΑ από την γονική της, ακόμη και τα PID,PPID. Η παλιά διεργασία καταστρέφεται – ουσιαστικά αντικαθίσταται από την νέα.

# fork / exec

18

- συνήθως οι λειτουργίες fork/exec εκτελούνται σε συνδυασμό. πχ:
  - Σε ένα shell έχουμε πληκτρολογήσει την εντολή ls.
  - Το shell κάνει fork τον εαυτό του.
  - Η γονική διεργασία του shell απλά περιμένει την θυγατρική να τερματίσει
  - Η θυγατρική διεργασία του shell κάνει “exec /bin/ls”, οπότε και τελικά εκτελείται η εντολή ls. Η ls θα έχει ως PPID το PID του γονικού shell.
- με τον ίδιο τρόπο ένα πρόγραμμα(διεργασία) μπορεί να εκτελέσει κάποιο άλλο πρόγραμμα.

# Signals

19

- ένα σήμα είναι μια μέθοδος για επικοινωνία με μια διεργασία
- Signals χρησιμοποιούνται στα συστήματα Unix και σε όλα τα λειτουργικά που ακολουθούν το πρότυπο POSIX.
- Είναι μια ασύγχρονη ενημέρωση της διεργασίας ότι συνέβη ένα γεγονός (event)
- όταν φτάνει ένα σήμα σε μια διεργασία, ο πυρήνας διακόπτει την κανονική εκτέλεση της διεργασίας και εκτελείται ο αντίστοιχος signal handler (μια συνάρτηση της διεργασίας που θα χειριστεί το συγκεκριμένο σήμα). όταν ολοκληρωθεί η συνάρτηση, η διεργασία επιστρέφει στην κανονική της εκτέλεση, στο σημείο που σταμάτησε.
- εάν μια διεργασία λάβει ένα σήμα για το οποίο δεν έχει οριστεί signal handler , τότε εκτελείται ο default του λειτουργικού.

# Signals

20

- Κάθε σήμα, αναπαρίσταται με κάποιο νούμερο. Η αρίθμηση μπορεί να διαφέρει από σύστημα σε σύστημα

Signal	Description	Signal number on Linux x86
SIGABRT	Process aborted	6
SIGALRM	Signal raised by alarm	14
SIGBUS	Bus error: "access to undefined portion of memory object"	7
SIGCHLD	Child process terminated, stopped (or continued*)	17
SIGCONT	Continue if stopped	18
SIGFPE	Floating point exception: "erroneous arithmetic operation"	8
SIGHUP	Hangup	1
SIGILL	Illegal instruction	4
SIGINT	Interrupt	2
SIGKILL	Kill (terminate immediately)	9
SIGPIPE	Write to pipe with no one reading	13
SIGQUIT	Quit and dump core	3
SIGSEGV	Segmentation violation	11
SIGSTOP	Stop executing temporarily	19
SIGTERM	Termination (request to terminate)	15
SIGTSTP	Terminal stop signal	20
SIGTTIN	Background process attempting to read from tty ("in")	21
SIGTTOU	Background process attempting to write to tty ("out")	22

# Signals

21

- Κάθε σήμα, αναπαρίσταται με κάποιο νούμερο. Η αρίθμηση μπορεί να διαφέρει από σύστημα σε σύστημα

Signal	Description	Signal number on Linux x86
SIGUSR1	User-defined 1	10
SIGUSR2	User-defined 2	12
SIGPOLL	Pollable event	29
SIGPROF	Profiling timer expired	27
SIGSYS	Bad syscall	31
SIGTRAP	Trace/breakpoint trap	5
SIGURG	Urgent data available on socket	23
SIGVTALRM	Signal raised by timer counting virtual time: "virtual timer expired"	26
SIGXCPU	CPU time limit exceeded	24
SIGXFSZ	File size limit exceeded	25

# Signals - kill

22

- Μπορεί ο χρήστης (ή μια διεργασία) να στείλει ένα σήμα σε μια διεργασία με την εντολή kill.
- Η εντολή kill ΔΕΝ «ΣΚΟΤΩΝΕΙ» μια διεργασία, αλλά στέλνει ένα σήμα σε αυτήν.
- Κάποια σήματα έχει νόημα να τα στείλει ο χρήστης σε μια διεργασία, ενώ κάποια άλλα όχι (χρησιμοποιούνται για εσωτερικούς σκοπούς από το λειτουργικό).

# Signals - kill

23

- Τα περισσότερα χρησιμοποιούμενα σήματα είναι:
  - 15 (TERM): Λέμε στην διεργασία να τερματίσει. Η διεργασία θα τερματίσει αν το θέλει.
  - 9 (KILL): Τερματίζουμε βίαια την διεργασία. Η διεργασία θα τερματίσει είτε το θέλει είτε όχι.
  - 19 (STOP): Λέμε στην διεργασία (ουσιαστικά στο λειτουργικό) να μπει σε STOP mode. Η διεργασία σταματά να εκτελείται, αλλά δεν τερματίζεται.
  - 18 (CONT): Λέμε στην διεργασία που ήταν σε STOP mode να συνεχίσει να εκτελείται.

# Signals - kill

24

- Κάποια σήματα μπορούμε να τα στείλουμε σε μια διεργασία από το πληκτρολόγιο (δεδομένου ότι η διεργασία εκτελείται σε ένα τερματικό):
  - Ctrl-C sends an INT signal (SIGINT); by default, this causes the process to terminate.
  - Ctrl-Z sends a TSTP signal (SIGTSTP); by default, this causes the process to suspend execution.
- Δείτε τις εργαστηριακές ασκήσεις για παραδείγματα....



# στοιχεία διεργασιών

25

- ένα τυπικό σύστημα Unix/Linux μας δίνει πρόσβαση σε όλα τα στοιχεία που διατηρεί ο πυρήνας για κάθε διεργασία μέσα από τον κατάλογο /proc

```
asidirop@aetos:~$ ls -l /proc
total 0
dr-xr-xr-x  7 root      root          0 Apr 26 17:11 1
dr-xr-xr-x  7 root      root          0 Apr 26 17:11 10
dr-xr-xr-x  7 dovecot   dovecot       0 Apr 26 20:03 10359
dr-xr-xr-x  7 dovecot   dovecot       0 Apr 26 19:48 10566
dr-xr-xr-x  7 dimver    x0607        0 Apr 26 19:48 10617
dr-xr-xr-x  7 dovecot   dovecot       0 Apr 26 19:48 10618
dr-xr-xr-x  7 dimver    x0607        0 Apr 26 19:48 10620
dr-xr-xr-x  7 dimver    x0607        0 Apr 26 19:49 10656
dr-xr-xr-x  7 dimver    x0607        0 Apr 26 19:49 10682
dr-xr-xr-x  7 kourou    conit        0 Apr 26 08:01 10717
dr-xr-xr-x  7 dovecot   dovecot       0 Apr 26 20:04 11076
dr-xr-xr-x  7 dovecot   dovecot       0 Apr 26 16:02 1122
dr-xr-xr-x  7 bind      bind         0 Apr 26 10:58 11570
dr-xr-xr-x  7 root      root         0 Apr 26 20:04 11669
dr-xr-xr-x  7 root      root         0 Apr 26 13:15 1176
dr-xr-xr-x  7 root      root         0 Apr 26 18:05 11798
dr-xr-xr-x  7 dovecot   dovecot       0 Apr 26 18:06 11866
```

θυμηθείτε πως  
στο UNIX τα  
πάντα  
αναπαρίστανται  
με αρχεία και  
καταλόγους

στον κατάλογο  
/proc ένας  
κατάλογος για  
κάθε διεργασία  
με όνομα το PID.

# στοιχεία διεργασιών

26

- ένα τυπικό σύστημα Unix/Linux μας δίνει πρόσβαση σε όλα τα στοιχεία που διατηρεί ο πυρήνας για κάθε διεργασία μέσα από τον κατάλογο /proc

```
asidirop@aetos:~$ ls -l /proc/22967
total 0
dr-xr-xr-x 2 asidirop conit 0 Apr 26 20:19 attr
-r----- 1 asidirop conit 0 Apr 26 20:19 auxv
-r--r--r-- 1 asidirop conit 0 Apr 26 20:19 cgroup
--w----- 1 asidirop conit 0 Apr 26 20:19 clear_refs
-r--r--r-- 1 asidirop conit 0 Apr 26 20:09 cmdline
-rw-r--r-- 1 asidirop conit 0 Apr 26 20:19 comm
-rw-r--r-- 1 asidirop conit 0 Apr 26 20:19 coredump_filter
-r--r--r-- 1 asidirop conit 0 Apr 26 20:19 cpuset
lrwxrwxrwx 1 asidirop conit 0 Apr 26 20:19 cwd -> /home/staff/ektaktoi/asidirop
-r----- 1 asidirop conit 0 Apr 26 20:19 environ
lrwxrwxrwx 1 asidirop conit 0 Apr 26 20:19 exe -> /bin/bash
dr-x----- 2 asidirop conit 0 Apr 26 20:09 fd
dr-x----- 2 asidirop conit 0 Apr 26 20:19 fdinfo
-r--r--r-- 1 asidirop conit 0 Apr 26 20:19 io
-r--r--r-- 1 asidirop conit 0 Apr 26 20:19 limits
-rw-r--r-- 1 asidirop conit 0 Apr 26 20:19 loginuid
-r--r--r-- 1 asidirop conit 0 Apr 26 20:19 maps
-rw----- 1 asidirop conit 0 Apr 26 20:19 mem
-r--r--r-- 1 asidirop conit 0 Apr 26 20:19 mountinfo
-r--r--r-- 1 asidirop conit 0 Apr 26 20:19 mounts
-r----- 1 asidirop conit 0 Apr 26 20:19 mountstats
```

θυμηθείτε πως  
στο UNIX τα  
πάντα  
αναπαρίστανται  
με αρχεία και  
καταλόγους

# στοιχεία διεργασιών

27

- οι πληροφορίες που μας εμφανίζει η εντολή ps «προέρχονται» από τα αρχεία status των διεργασιών.

```
asidirop@aetos:~$ more /proc/22967/status
Name:      bash
State:     S (sleeping)
Tgid:      22967
Pid:       22967
PPid:      22963
TracerPid: 0
Uid:       1143      1143      1143      1143
Gid:       993       993       993       993
FDSize:    256
Groups:    993
VmPeak:    19904 kB
VmSize:    19840 kB
VmLck:     0 kB
VmHWM:     2096 kB
VmRSS:     2096 kB
VmData:    424 kB
VmStk:     136 kB
VmExe:     868 kB
VmLib:     1972 kB
VmPTE:     60 kB
VmSwap:    0 kB
Threads:   1
SigQ:      0/16382
SigPnd:    0000000000000000
ShdPnd:    0000000000000000
SigBlk:    0000000000010000
SigIgn:    0000000000384004
SigCgt:    000000004b813efb
CapInh:    0000000000000000
CapPrm:    0000000000000000
CapEff:    0000000000000000
CapBnd:    ffffffff
Cpus allowed:  ff
```

# Περιβάλλον

28

- Για κάθε διεργασία διατηρείται ένας πίνακας με "μεταβλητές περιβάλλοντος" (Environment variables)
- Οι μεταβλητές περιβάλλοντος δεν υπάρχουν μόνο στο shell αλλά σε όλες τις διεργασίες.
- οι μεταβλητές περιβάλλοντος χρησιμοποιούνται:
  - για να δώσουν πληροφορίες από το σύστημα στον χρήστη (πχ, USER, SHELL, PWD)
  - για να δώσει ο χρήστης πληροφορίες (ρυθμίσεις) στο σύστημα ή σε διεργασίες (πχ: PATH, LANG, LS\_COLORS)
  - για να δώσει μια διεργασία πληροφορίες σε κάποια θυγατρική της (πχ: XDG\_SESSION\_COOKIE=ad911ed4a37cbe23249debec4bc87826-1335941063.445049-1438191725)
- ΟΛΕΣ οι μεταβλητές περιβάλλοντος κληρονομούνται από τις διεργασίες στις θυγατρικές τους.

# Περιβάλλον

29

- εμφάνιση των μεταβλητών περιβάλλοντος:

```
asidirop@aetos:~$ env
TERM=xterm
SHELL=/bin/bash
SSH_CLIENT=94.68.138.167 55038 22
SSH_TTY=/dev/pts/0
USER=asidirop
MAILCHECK=3600
MAIL=/var/mail/asidirop
PATH=/usr/local/bin:/usr/bin:/bin:/usr/games:/home/staff/ektaktoi/asidirop/bin
PWD=/home/staff/ektaktoi/asidirop
LANG=en_US.UTF-8
SHLVL=1
HOME=/home/staff/ektaktoi/asidirop
LOGNAME=asidirop
SSH_CONNECTION=94.68.138.167 55038 195.251.123.232 22
DISPLAY=:0
_=/usr/bin/env
```

# Περιβάλλον

30

- ορισμός μεταβλητής περιβάλλοντος μέσα από το sh (bash)
  - αν υπάρχει ήδη η μεταβλητή στον πίνακα μεταβλητών περιβάλλοντος

```
asidirop@aetos:~$ LANG=el_GR.UTF-8
asidirop@aetos:~$ env
TERM=xterm
SHELL=/bin/bash
SSH_CLIENT=94.68.138.167 55038 22
SSH_TTY=/dev/pts/0
USER=asidirop
MAILCHECK=3600
MAIL=/var/mail/asidirop
PATH=/usr/local/bin:/usr/bin:/bin:/usr/games:/home/staff/ektaktoi/asidirop/bin
PWD=/home/staff/ektaktoi/asidirop
LANG=el_GR.UTF-8
SHLVL=1
HOME=/home/staff/ektaktoi/asidirop
LOGNAME=asidirop
SSH_CONNECTION=94.68.138.167 55038 195.251.123.232 22
DISPLAY=:0
_=/usr/bin/env
```

# Περιβάλλον

31

- ορισμός μεταβλητής περιβάλλοντος μέσα από το sh (bash)
  - αν ΔΕΝ υπάρχει ήδη η μεταβλητή στον πίνακα μεταβλητών περιβάλλοντος

```
asidirop@aetos:~$ TEST=xyz
asidirop@aetos:~$ export TEST
asidirop@aetos:~$ env
TERM=xterm
SHELL=/bin/bash
SSH_CLIENT=94.68.138.167 55038 22
SSH_TTY=/dev/pts/0
USER=asidirop
MAILCHECK=3600
MAIL=/var/mail/asidirop
PATH=/usr/local/bin:/usr/bin:/bin:/usr/games:/home/staff/ektaktoi/asi
dirop/bin
PWD=/home/staff/ektaktoi/asidirop
LANG=el_GR.UTF-8
TEST=xyz
SHLVL=1
```

# Περιβάλλον

32

- ορισμός μεταβλητής περιβάλλοντος μέσα από το sh (bash)
  - εναλλακτικά:

```
asidirop@aetos:~$ export TEST2=qwe
asidirop@aetos:~$ env
MAIL=/var/mail/asidirop
PATH=/usr/local/bin:/usr/bin:/bin:/usr/games:/home/staff/ektaktoi/asidirop/bin
PWD=/home/staff/ektaktoi/asidirop
LANG=el_GR.UTF-8
TEST=xyz
SHLVL=1
HOME=/home/staff/ektaktoi/asidirop
LOGNAME=asidirop
SSH_CONNECTION=94.68.138.167 55038 195.251.123.232 22
TEST2=qwe
```



# Περιβάλλον

33

- Οι ρυθμίσεις γλώσσας γίνονται μέσα από τις μεταβλητές περιβάλλοντος:

```
asidirop@aetos:~$ ls /dfg
ls: cannot access /dfg: No such file or directory
asidirop@aetos:~$ export LANG=el_GR.UTF-8
asidirop@aetos:~$ ls /dfg
ls: cannot access /dfg: Δεν υπάρχει τέτοιο αρχείο ή κατάλογος
asidirop@aetos:~$
```

# Περιβάλλον

34

- σημαντική μεταβλητή περιβάλλοντος είναι η PATH. μέσω αυτής ορίζεται σε ποιους καταλόγους θα ψάχνει το shell να βρει το εκτελέσιμο της εντολής που έχουμε πληκτρολογήσει.
- PATH=/usr/local/bin:/usr/bin:/bin:/usr/games:/home/staff/ektaktoti/asidirop/bin
- Η μεταβλητή περιέχει διαδρομές σε καταλόγους. Οι διαδρομές μεταξύ τους χωρίζονται με τον χαρακτήρα :
- Εάν ένα εκτελέσιμο δεν βρίσκεται στο PATH (δηλαδή δεν είναι αποθηκευμένο σε κατάλογο που περιλαμβάνεται στο PATH) για να το εκτελέσουμε θα πρέπει να δώσουμε την διαδρομή προς αυτό (απόλυτη ή σχετική)

```
asidirop@aetos:~$ fsck
-bash: fsck: command not found
asidirop@aetos:~$ /sbin/fsck
fsck from util-linux-ng 2.17.2
If you wish to check the consistency of an XFS filesystem or
repair a damaged filesystem, see xfs_check(8) and xfs_repair(8).
e2fsck 1.41.12 (17-May-2010)
```

...

# Περιβάλλον

35

- Στις μεταβλητές περιβάλλοντος μιας διεργασίας έχουμε πρόσβαση με μέσω του /proc
- (δεν εμφανίζονται όμως "όμορφα" με μια απλή cat/more)

```
asidirop@aetos:~$ ps
  PID TTY          TIME CMD
 20870 pts/0    00:00:00 bash
 29411 pts/0    00:00:00 ps
asidirop@aetos:~$ cat /proc/20870/environ
LANG=en_US.UTF-
8USER=asidiropLOGNAME=asidiropHOME=/home/staff/ektaktoi/asidiropPATH=
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/gamesMAIL=/var/mail/as
idiropSHELL=/bin/bashSSH_CLIENT=94.68.138.167 55038
22SSH_CONNECTION=94.68.138.167 55038 195.251.123.232
22SSH_TTY=/dev/pts/0TERM=xterm
```