Εισαγωγή στα Λειτουργικά Συστήματα

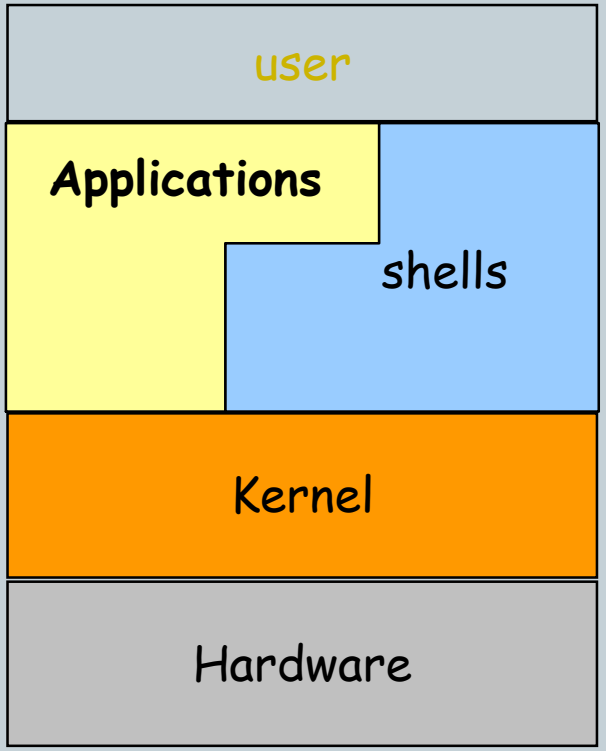
SET ΔΙΑΦΑΝΕΙΩΝ 10

SCRIPTS

ΑΝΤΩΝΗΣ ΣΙΔΗΡΟΠΟΥΛΟΣ

Φλοιός (Shell)

- Είναι το πρόγραμμα που διερμηνεύει (interpreter) τις εντολές που εισάγονται από το πληκτρολόγιο.
- Οι εντολές είναι δυνατόν να τοποθετηθούν σε ένα αρχείο που ονομάζεται shell script.
- Κάθε φορά που εισάγεται από το πληκτρολόγιο μια γραμμή το Λ.Σ. χρησιμοποιεί ένα τμήμα του για να αναλύσει το περιεχόμενο της γραμμής, τον command line processor (CLP). Στο UNIX ο CLP είναι εξ ολοκλήρου διαχωρισμένος από το υπόλοιπο Λ.Σ.



φλοιοί

- Σε κάθε σύστημα UNIX υπάρχουν τουλάχιστον δύο διαθέσιμοι φλοιοί: **Bourne shell (sh)** και **C-shell (csh)** (αναπτύχθηκε στα πλαίσια του BSD Unix). Ωστόσο η γλώσσα που χρησιμοποιείται για την επικοινωνία με καθένα από τους φλοιούς αυτούς είναι διαφορετική.
- Άλλοι φλοιοί:
  - Korn shell (ksh)
  - terminal-based C shell (tcsh)
  - Bourne again shell (bash) – βελτιωνένη έκδοση του sh και είναι πλέον ο εξ ορισμού φλοιός σε συστήματα linux. Μάλιστα σε αρκετές περιπτώσεις το Bourne Shell δεν υπάρχει κάν, και η εντολή sh είναι link στην εντολή bash.
- **Ανεξάρτητα από τον φλοιό που χρησιμοποιείται, όλοι έχουν ως στόχο να παρέχουν στο UNIX μια διεπαφή χρήστη (user interface).**

Χαρακτηριστικά φλοιών

- Διερμηνεία (interpretation) και επεξεργασία της γραμμής εντολών
- Δεσμευμένες λέξεις - εντολές
- Μετα-χαρακτήρες φλοιού (wildcards)
- **Εκτέλεση εντολών** (προγραμμάτων ή εσωτερικών εντολών του φλοιού - built-in commands)
- Διαχείριση αρχείων: ανακατεύθυνση εισόδου / εξόδου και διασωληνώσεις
- Υποστήριξη μεταβλητών
- Καθορισμό ψευδωνύμων (aliases)
- Έλεγχο του περιβάλλοντος και δημιουργία περιβάλλοντος προσαρμοσμένου στις απαιτήσεις του χρήστη
- Δημιουργία σεναρίων φλοιού (shell scripts)
- Εντολές if/for/while/until

shell scripts

- Η εντολή sh δέχεται ορίσματα. Αν δεχτεί όρισμα όνομα αρχείου, τότε από μόνη της διαβάζει το αρχείο (αντί από το stdin) και θεωρεί τα περιεχόμενά του ως εντολές προς εκτέλεση.

```
bash-2.05a$ cat file1
ls -l
whoami
date
bash-2.05a$ sh file1
total 24
-rw-r--r--    1 asidirop it             314 Jan 11  2003 cc
-rw-r--r--    1 asidirop it              18 Nov 20 15:22 file1
-rw-r--r--    1 asidirop it             183 Jan  8  2003 list
asidirop
Mon Nov 20 15:23:05 EET 2006
bash-2.05a$
```

shell

- Όταν ένα αρχείο περιέχει text το οποίο είναι εντολές, τότε αυτό το αρχείο ονομάζεται script.
- Όταν το αρχείο περιέχει εντολές για το shell, ονομάζεται shell script
- Ο χαρακτήρας # σηματοδοτεί σχόλιο για το shell

shell

- Στο UNIX τα scripts χρησιμοποιούνται συχνά. Για να μην χρειάζεται κάθε φορά να βρίσκει ο χρήστης για ποιο «πρόγραμμα» (ή shell) είναι οι εντολές που περιέχονται σε ένα script, υπάρχει η σύμβαση στην πρώτη γραμμή του αρχείου να περιέχεται το όνομα της «εντολής» που πρέπει να εκτελέσει το script.
  Η πρώτη γραμμή ενός script πρέπει να είναι:
  #!εντολή
- Για ένα shell script η πρώτη γραμμή πρέπει να είναι:
  #!/bin/sh
  ή
  #!/bin/bash
- Όταν το # βρίσκεται στην πρώτη γραμμή ενός αρχείου και ακολουθείται από το !, σημαίνει πως αυτό που ακολουθεί είναι η εντολή προς εκτέλεση.

shell

- Έστω ένα αρχείο με όνομα file1 που περιέχει:

```
#!/path/command
Lala
Foo
```

- Όταν προσπαθήσουμε να «εκτελέσουμε» το file1 με την εντολή:
  ./file1
  τότε το UNIX θα αναγνωρίσει ότι δεν πρόκειται για binary αρχείο αλλά για text αρχείο.
- Θα διαβάσει την πρώτη γραμμή και θα εκτελέσει την εντολή:

```
/path/command ./file1
```

shell

- Σε ένα shell script (έστω με όνομα αρχείου file1) βάζουμε:

```
#!/bin/sh
Εντολές
```

- Όταν προσπαθήσουμε να το εκτελέσουμε με την εντολή ./file1
  το UNIX θα δει ότι δεν πρόκειται για binary εκτελέσιμο αρχείο, θα διαβάσει την πρώτη γραμμή, βρεί το #! Και θα εκτελέσει την εντολή:

```
/bin/sh ./file1
```

shell

- Ένα text αρχείο (script) για να μπορέσουμε να το εκτελέσουμε πρέπει να έχουμε άδεια πρόσβασης “execute”.
- Συνεπώς όταν δημιουργούμε scripts, πρέπει να τους δώσουμε την άδεια πρόσβασης “execute”, η οποία όπως έχουμε δει δεν τίθεται από μόνη της όταν δημιουργούμε αρχεία.
- Αν το αρχείο δεν έχει άδεια “execute” θα πάρουμε το μήνυμα:

```
aetos_test_27_$./file1
./file1: Permission denied.
aetos_test_27_$ls -l
total 24
-rw-r--r--    1 asidirop it           314 Jan 11  2003 cc
-rw-------    1 asidirop it            28 Nov 20 18:40 file1
-rw-r--r--    1 asidirop it           183 Jan  8  2003 list
```

shell

- Εφόσον σε ένα script έχουμε άδεια “execute” όταν το εκτελούμε διαβάζει το λειτουργικό την πρώτη γραμμή κειμένου για να βρει ποιος είναι ο interpreter. Αφού αναγνωρίσει ποιος είναι ο interpreter (πχ /bin/sh) καλεί την εντολή:
  /bin/sh file1
  Ο interpreter αναγνωρίζει ότι δόθηκε όρισμα όνομα αρχείου και προσπαθεί να το ανοίξει για ανάγνωση.
- Αν δεν έχουμε δικαίωμα ανάγνωσης στο αρχείο τότε θα πάρουμε το μήνυμα λάθους:

```
aetos_test_27_$./file1
/bin/sh: ./file1: cannot open: Permission denied
aetos_test_45_$/bin/sh file1
/bin/sh: file1: cannot open: Permission denied
aetos_test_27_$ls -l
total 24
-rw-r--r--    1 asidirop it           314 Jan 11  2003 cc
--wx------    1 asidirop it            28 Nov 20 18:48 file1
-rw-r--r--    1 asidirop it           183 Jan  8  2003 list
```

Διαφορετικά κελύφη

- Η πρώτη γραμμή στο script ορίζει το κέλυφος που θα χρησιμοποιηθεί για την εκτέλεση των εντολών που περιέχει το σενάριο (το ! ακολουθεί αμέσως μετά το # χωρίς ενδιάμεσα κενά) :

`#!/bin/sh`	**Bourne shell**
`#!/bin/csh`	**C-shell**
`#!/bin/tcsh`	**TC-shell**
`#!/bin/bash`	**BASH shell**
`#!/usr/bin/perl`	**Perl script**

- Όλα τα κελύφη παρέχουν την επιλογή **–v** για να εμφανίζεται στην οθόνη κάθε γραμμή του σεναρίου καθώς αυτό διαβάζεται και την επιλογή **–x** για να εμφανίζονται οι εντολές, καθώς εκτελούνται.
- Οι επιλογές αυτές είναι πολύ χρήσιμες για τη συντακτική ανάλυση των εντολών.
- Εμφάνιση του φλοιού που χρησιμοποιείται :
  echo $SHELL
  - **SHELL** είναι η μεταβλητή περιβάλλοντος που κρατά το όνομα του τρέχοντος φλοιού, ενώ **$SHELL** είναι η τιμή αυτής της μεταβλητής.

- όρισμα -v

```
bash-2.05a$ cat file1
#!/bin/sh -v
whoami # dixnei poios eimai
#Twra typonoyme tin hmeromhnia
date
bash-2.05a$ ./file1
#!/bin/sh -v
whoami # dixnei poios eimai
asidirop
#Twra typonoyme tin hmeromhnia
date
Mon Nov 20 19:16:30 EET 2006
bash-2.05a$
```

- όρισμα -x

```
bash-2.05a$ cat file1
#!/bin/sh -x
whoami # dixnei poios eimai
#Twra typonoyme tin hmeromhnia
date
bash-2.05a$ ./file1
+ whoami
asidirop
+ date
Mon Nov 20 19:17:13 EET 2006
bash-2.05a$
```

shell

- Για να εκτελέσουμε ένα πρόγραμμα ή script πρέπει να αναφερθούμε στο όνομα αρχείου είτε με σχετική είτε με απόλυτη διαδρομή. Πχ:

```
aetos_test_51_$cat file1
#!/bin/sh
whoami
date
aetos_test_52_$./file1
asidirop
Mon Nov 20 18:54:36 EET 2006
aetos_test_53_$~/test/file1
asidirop
Mon Nov 20 18:54:42 EET 2006
aetos_test_54_$pwd
/usr/people/staff/ektaktoi/it/asidirop/test
```

shell

- Αν δεν δώσουμε διαδρομή για το πρόγραμμα που προσπαθούμε να εκτελέσουμε, το shell δεν ψάχνει την εντολή στον τρέχον κατάλογο, αλλά στους καταλόγους που είναι ορισμένοι στο PATH.

```
bash-2.05a$ ./file1
asidirop
Mon Nov 20 18:57:17 EET 2006
bash-2.05a$ file1
bash: file1: command not found
bash-2.05a$ bash-2.05a$
```