11

Το κέλυφός σου

Μέχρι τώρα σε αυτό το βιβλίο έτρεχες εντολές στο κέλυφός σου, αλλά δεν εστίασες στο ίδιο το κέλυφος. Σε αυτό το κεφάλαιο, θα κοιτάξεις δύο εντολές που επηρεάζουν τη χρήση του κελύφους: την `history`, που παραθέτει όλα όσα έχεις εισάγει στη γραμμή εντολών, και την `alias`, που σου επιτρέπει να δημιουργείς συντομεύσεις για εντολές. Και οι δύο είναι χρήσιμες και μπορούν να σου γλιτώσουν πολύ χρόνο όταν χρησιμοποιείς τη γραμμή εντολών. Η τεμπελιά είναι καλό πράγμα όταν αφορά χρήστες υπολογιστών, και αυτές είναι σίγουρα δύο εντολές που θα σε βοηθήσουν να είσαι όσο το δυνατόν πιο τεμπέλης, όταν χρησιμοποιείς το Linux σου.

Εμφάνιση του ιστορικού της γραμμής εντολών

`history`

Κάθε φορά που πληκτρολογείς μια εντολή στο κέλυφός σου, αυτή η εντολή καταχωρείται σε ένα αρχείο με το όνομα `.bash_history` στον κεντρικό κατάλογό σου (η τελεία μπροστά από το όνομα του αρχείου σημαίνει πως είναι κρυφό εκτός εάν χρησιμοποιήσεις την

`ls -a`). Από προεπιλογή, αυτό το αρχείο κρατά τις τελευταίες 500 γραμμές που εισήγαγες στη γραμμή εντολών. Εάν θέλεις να δεις αυτή τη λίστα, απλά χρησιμοποίησε την εντολή `history`.

```
$ history

  496  ls
  497  cd rsync_ssh
  498  ls
  499  cat linux
  500  exit
```

Επειδή κοιτάς 500 αποτελέσματα, θα περάσουν από μπροστά σου τόσο γρήγορα που δεν πρόκειται να δεις κανένα πριν φτάσεις στο τέλος. Θέλεις να δεις τα αποτελέσματα μια σελίδα κάθε φορά; Απευθύνσου στην παλιά σου γνώριμη τη `less`:

```
$ history | less
```

Τώρα μπορείς να κινηθείς μέσα στα αποτελέσματα με πολύ περισσότερη ευκολία.

ΠΡΟΕΙΔΟΠΟΙΗΣΗ: Τώρα καταλαβαίνεις γιατί πρέπει να είσαι προσεκτικός όταν πληκτρολογείς κωδικούς και άλλα ευαίσθητα δεδομένα στη γραμμή εντολών: Οποιοσδήποτε μπορεί να δει το αρχείο `.bash_history`, μπορεί να δει και αυτούς τους κωδικούς. Να είσαι προσεκτικός και να σκέπτεσαι τι εισάγεις απευθείας στη γραμμή εντολών!

Επανεκτέλεση της τελευταίας εντολής

```
!!
```

Εάν θέλεις να τρέξεις μια δεύτερη φορά την τελευταία εντολή που χρησιμοποίησες, βάλε δύο θαυμαστικά. Αυτό κοιτάζει στο αρχείο με το ιστορικό και τρέχει την τελευταία εντολή στη λίστα.

```
$ pwd
/home/scott
$ !!
pwd
/home/scott
```

Πρόσεξε ότι πρώτα βλέπεις την εντολή που πρόκειται να εκτελεστεί και στη συνέχεια τα αποτελέσματά της. Αυτός είναι ένας πραγματικά χρήσιμος τρόπος για να βάζεις τον υπολογιστή σου να κάνει την κουραστική δουλειά για σένα.

Εκτέλεση μιας προηγούμενης εντολής με τη χρήση αριθμών

`![##]`

Όταν τρέχεις τη `history`, αυτή αυτομάτως βάζει έναν αριθμό μπροστά από κάθε προηγούμενη εντολή. Εάν θέλεις να τρέξεις μια προηγούμενη εντολή και γνωρίζεις τον αριθμό που έχει καθορίσει η `history` για αυτήν, απλά βάλε ένα θαυμαστικό ακολουθούμενο από τον αριθμό της εντολής και αυτή θα τρέξει ξανά.

```
$ pwd
/home/scott
$ whoami
scott
$ !499
pwd
/home/scott
```

Εάν δεν είσαι σίγουρος για τον αριθμό, τρέξε τη `history` ξανά για να τον μάθεις. Πρόσεξε ότι η εντολή `pwd` σε αυτό το παράδειγμα είχε τον αριθμό 449 την πρώτη φορά, αλλά αφότου την έτρεξα ξανά με την `!499`, απόκτησε τον αριθμό 498, διότι κατέβηκε στη λίστα λόγω της νέας μου εντολής.

Εκτέλεση μιας προηγούμενης εντολής με τη χρήση μιας συμβολοσειράς

`![string]`

Η δυνατότητα να τρέχεις ξανά μια εντολή με αναφορά του αριθμού της είναι καλή, αλλά απαιτεί να γνωρίζεις τον αριθμό αυτό, κάτι που μπορεί να είναι κουραστικό να ανακαλύψεις (θα βοηθούσε να έστελνες την έξοδο της `history` στη `grep`, αλλά και πάλι δεν είναι το ιδανικό). Συχνά, ένας καλύτερος τρόπος να αναφερθείς σε μια προηγούμενη εντολή είναι με το πραγματικό της όνομα. Εάν προσθέσεις μετά το θαυμαστικό τα πρώτα γράμματα μιας εντολής, το κέλυφός σου τρέχει την πρώτη εντολή που βρίσκει κοιτάζοντας προς τα πίσω στο `.bash_history`.

```
$ cat /home/scott/todo
Buy milk
Buy dog food
Renew Linux Magazine subscription
$ cd /home/scott/pictures
$ !cat
cat /home/scott/todo
Buy milk
Buy dog food
Renew Linux Magazine subscription
```

Εάν η εντολή `cat` βρίσκεται τρεις φορές στο ιστορικό -στο 35 (`cat /home/scott/todo`), στο 412 (`cat /etc/apt/sources.list`) και στο 496 (`cat /home/scott/todo`)- και πατήσεις `!cat`, αυτή με τον αριθμό 496 είναι αυτή που τρέχει. Εάν θέλεις να τρέξεις την `cat` που βρίσκεται στο 412, πρέπει να τρέξεις είτε `!412` είτε να προσθέσεις μετά το θαυμαστικό αρκετές πληροφορίες ώστε να ξέρει ότι αναφέρεσαι στην εντολή με τον αριθμό 412.

```
$ !cat /etc
cat /etc/apt/sources.list
deb http://us.archive.ubuntu.com/ubuntu breezy main
➥restricted
deb-src http://us.archive.ubuntu.com/ubuntu breezy
➥main restricted
```

Επειδή για τους ανθρώπους είναι πολύ πιο εύκολο να θυμούνται λέξεις αντί για αριθμούς, μάλλον θα καταλήξεις να χρησιμοποιείς αυτή τη μέθοδο για να ανακαλείς παλιές εντολές. Εάν κάποια στιγμή δεν είσαι σίγουρος, τρέξε την `history` και ρίξε μια ματιά.

Εμφάνιση όλων των ψευδωνύμων εντολών

alias

Εάν χρησιμοποιείς μια εντολή όλη την ώρα ή εάν μια εντολή είναι ιδιαίτερα εκτενής και ανοικονόμητη στην πληκτρολόγηση, τότε αρμόζει να δημιουργήσεις γι' αυτήν ένα ψευδώνυμο. Αφού δημιουργήσεις το ψευδώνυμο, πληκτρολογείς το όνομά του και τρέχει η εντολή στην οποία αναφέρεται. Βέβαια, εάν μια εντολή είναι ιδιαίτερα περίπλοκη ή περιέχει πολλές γραμμές, θα έπρεπε αντ' αυτού να τη μετατρέψεις σε ένα σενάριο ή μια συνάρτηση. Αλλά για μικρά πράγματα, τα ψευδώνυμα είναι τέλεια.

Τα ψευδώνυμα βρίσκονται αποθηκευμένα σε ένα αρχείο στον κεντρικό κατάλογό σου. Μπορεί να τα βρεις στο `.bashrc`, αλλά είναι πιο πιθανό (ή μάλλον πιο σωστό) να βρίσκονται στο `.bash_aliases`. Οι περισσότερες διανομές Linux έρχονται με διάφορα ψευδώνυμα ήδη καθορισμένα. Για να δεις αυτή τη λίστα, απλά γράψε `alias` στη γραμμή εντολών.

```
$ alias
alias la='ls -a'
alias ll='ls -l'
```

Οι περισσότερες διανομές κρατούν σκοπίμως τα προεπιλεγμένα ψευδώνυμα στο ελάχιστο. Είναι στο χέρι σου να προσθέσεις νέα, όπως θα δεις σύντομα.

Εμφάνιση ενός συγκεκριμένου ψευδωνύμου εντολής

alias [όνομα ψευδωνύμου]

Αφού ορίσεις αρκετά ψευδώνυμα, μπορεί να είναι δύσκολο να βρεις ένα συγκεκριμένο πληκτρολογώντας την εντολή `alias`. Εάν θέλεις να ξαναδείς τι κάνει ένα συγκεκριμένο ψευδώνυμο, απλά βάλε το όνομά του μετά την `alias`.

```
$ alias wgetpage
alias wgetpage='wget --html-extension --recursive
➥--convert-links --page-requisites --no-parent $1'
```

Τώρα γνωρίζεις ακριβώς τι κάνει το ψευδώνυμο `wgetpage`, εύκολα και γρήγορα.

ΣΗΜΕΙΩΣΗ: Θα μάθεις περισσότερα για τη wget στο κεφάλαιο 15, "Εργασία στο δίκτυο".

Δημιουργία ενός νέου προσωρινού ψευδωνύμου

alias [ψευδώνυμο]= '[εντολή]'

Εάν βρεθείς να πληκτρολογείς την ίδια εντολή ξανά και ξανά, μάλλον είναι ώρα να φτιάξεις ένα ψευδώνυμο. Για παράδειγμα, για να δεις μόνο τους υποκαταλόγους του τρέχοντος καταλόγου, θα χρησιμοποιήσεις την εντολή `ls -d */`. Για να δημιουργήσεις

ένα προσωρινό ψευδώνυμο για αυτή την εντολή, χρησιμοποίησε τα παρακάτω:

```
$ ls -d */
by_pool/  libby_pix/  on_floor/
$ alias lsd='ls -d */'
$ lsd
by_pool/  libby_pix/  on_floor/
```

Θα πρέπει να καταλάβεις ορισμένα πράγματα για αυτό τον τρόπο χρήσης της `alias`. Το όνομα του ψευδωνύμου δεν μπορεί να περιλαμβάνει το =, κάτι που είναι λογικό διότι ακολουθείται από ένα = όταν το ορίζεις. Μπορείς όμως να έχεις ένα = στο ίδιο το ψευδώνυμο. Επίσης, ένα ψευδώνυμο που θα δημιουργηθεί με αυτόν τον τρόπο, διαρκεί μόνο όσο είναι ενεργή η συγκεκριμένη συνεδρία κελύφους. Εάν αποσυνδεθείς, το ψευδώνυμο χάνεται. Θέλεις να δημιουργήσεις ένα ψευδώνυμο που θα παραμένει και αφότου αποσυνδεθείς; Τότε διάβασε την επόμενη ενότητα, "Δημιουργία ενός νέου μόνιμου ψευδωνύμου".

Δημιουργία ενός νέου μόνιμου ψευδωνύμου

`alias [όνομα ψευδωνύμου]= '[εντολή]'`

Εάν θέλεις τα ψευδώνυμά σου να παραμένουν, πρέπει να τα προσθέσεις στο αρχείο που χρησιμοποιεί το κέλυφος για να καταχωρεί τα ψευδώνυμα. Τις περισσότερες φορές αυτό το αρχείο είναι το `.bashrc` ή το `.bash_aliases`. Σε αυτή την περίπτωση, θα χρησιμοποιήσουμε το `.bash_aliases`. Ανεξάρτητα από το ποιο αρχείο είναι, να είσαι προσεκτικός όταν το επεξεργάζεσαι, διότι μπορεί να δημιουργήσεις προβλήματα που θα φανούν την επόμενη φορά που θα συνδεθείς. Εάν θέλεις να είσαι πραγματικά προσεκτικός, φτιάξε ένα εφεδρικό αρχείο προτού το επεξεργαστείς. Καλύτερα ασφαλής παρά μετανιωμένος.

ΣΗΜΕΙΩΣΗ: Πώς μπορείς να βρεις ποιο αρχείο πρέπει να χρησιμοποιήσεις; Απλό: Πληκτρολόγησε `ls -a ~`. Εάν υπάρχει το `.bash_aliases`, χρησιμοποίησε αυτό. Αλλιώς, κοίταξε στο `.bashrc` και δες εάν έχουν οριστεί άλλα ψευδώνυμα εκεί. Εάν δεν υπάρχουν καθόλου ψευδώνυμα εκεί, ρίξε μια ματιά στο .profile, που χρησιμοποιείται περιστασιακά.

Για να προσθέσεις ένα ψευδώνυμο στο `.bash_aliases`, άνοιξέ το με τον συντάκτη κειμένου της προτίμησής σου και πρόσθεσε μια γραμμή σαν την παρακάτω:

```
alias lsd='ls -d */'
```

Ο ίδιος κανόνας που συζητήθηκε στην ενότητα "Δημιουργία ενός νέου προσωρινού ψευδωνύμου" ισχύει και εδώ: Το όνομα του ψευδωνύμου δεν μπορεί να περιλαμβάνει =. Αφού προσθέσεις το ψευδώνυμο στο `.bash_aliases`, σώσε το αρχείο και κλείσε το. Αλλά το ψευδώνυμο δεν λειτουργεί ακόμα. Το αρχείο `.bash_aliases` (ή το `.bashrc` εάν αυτό χρησιμοποίησες) πρέπει να επαναφορτωθεί για να λειτουργήσει το νέο ψευδώνυμο. Αυτό μπορείς να το κάνεις με δύο τρόπους. Μπορείς είτε να αποσυνδεθείς και να συνδεθείς ξανά, που είναι κόπος και δεν συνίσταται, ή μπορείς απλά να τρέξεις αυτή την εντολή:

```
$ . .bash_aliases
```

Μια τελεία, μετά ένα κενό και μετά το όνομα του αρχείου που ξεκινά με μια τελεία. Τώρα το νέο ψευδώνυμο θα λειτουργεί. Επειδή πρέπει να επαναφορτώνεις το αρχείο κάθε φορά που προσθέτεις ένα ψευδώνυμο, είναι καλή ιδέα να προσθέτεις πολλά ταυτόχρονα για να γλιτώνεις ταλαιπωρία.

Κατάργηση ψευδωνύμου

unalias

Όλα τα όμορφα πράγματα έχουν ένα τέλος και ορισμένες φορές ένα ψευδώνυμο παύει να είναι χρήσιμο. Για να καταργήσεις ένα

ψευδώνυμο, χρησιμοποίησε την εντολή `unalias`.

```
$ ls -d */
by_pool/  libby_pix/  on_floor/
$ alias lsd='ls -d */'
$ lsd
by_pool/  libby_pix/  on_floor/
$ unalias lsd
$ lsd
$
```

Πρόσεξε όμως ότι αυτή η εντολή λειτουργεί μόνιμα μόνο για τα προσωρινά ψευδώνυμα κελύφους, που συζητήθηκαν στην ενότητα "Δημιουργία ενός νέου προσωρινού ψευδωνύμου". Το ψευδώνυμο `lsd` του προηγούμενου παραδείγματος εξαφανίστηκε για τα καλά. Εάν χρησιμοποιήσεις την εντολή `unalias` σ' ένα ψευδώνυμο του `.bash_aliases`, θα καταργηθεί και αυτό, αλλά μόνο για όσο παραμένεις συνδεδεμένος. Όταν αποσυνδεθείς και συνδεθείς ξανά, ή επαναφορτώσεις το `.bash_aliases`, το ψευδώνυμο έχει επιστρέψει.

Για να αφαιρέσεις ψευδώνυμα από το `.bash_aliases`, πρέπει να επεξεργαστείς το αρχείο και να αφαιρέσεις τις γραμμές που περιέχουν τα ψευδώνυμα. Εάν πιστεύεις ότι υπάρχει περίπτωση να θέλεις να ξαναχρησιμοποιήσεις το ψευδώνυμο κάποια στιγμή, απλά βάλε μια δίεση μπροστά από το ψευδώνυμο (που το μετατρέπει σε σχόλιο) με αυτό τον τρόπο:

```
# alias lsd= 'ls -d */'
```

Σώσε το `.bash_aliases`, επαναφόρτωσέ το με την εντολή `. .bash_aliases` και το ψευδώνυμο δεν λειτουργεί πλέον. Αλλά εάν ποτέ το χρειαστείς ξανά, άνοιξε το `.bash_aliases`, αφαίρεσε τη δίεση, σώσε και επαναφόρτωσε το αρχείο και είναι πάλι έτοιμο προς χρήση.

Κατάληξη

Ένας από τους στόχους σου σαν χρήστης του Linux θα πρέπει να είναι η μείωση των χαρακτήρων που πρέπει να πληκτρολογήσεις για να πετύχεις τους στόχους σου. Οι αρχικοί υπεύθυνοι ανάπτυξης του Unix είχαν ισχυρή πίστη σε αυτή τη λογική — αυτός είναι ο λόγος που γράφεις `ls` αντί για `list` και `mkdir` αντί για `makedirectory`.

Οι εντολές που έμαθες σε αυτό το κεφάλαιο —η `history` και η `alias`— βοηθούν και οι δύο σε αυτή την κατεύθυνση. Βαρέθηκες να γράφεις μια μεγάλη και περίπλοκη εντολή; Κάνε αναφορά σε αυτή με την `history` ή φτιάξε της ένα ψευδώνυμο. Όπως και να 'χει γλιτώνεις γράψιμο και το πληκτρολόγιό σου (και, κυρίως, τα χέρια σου και οι καρποί σου) θα σε ευχαριστούν για αυτό.