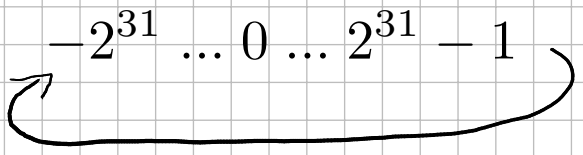
https://www.cs.unibo.it/~renzo/so/materiale2324/domande/20230928_domande_unix_c

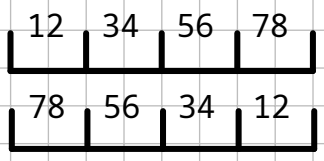
RAPPRESENTAZIONE DEI DATI

Come vengono rappresentati in memoria gli interi con e senza segno?



I negativi si salvano in complemento a 2

0x12345678



BIG endian

LITTLE endian

Big endian / little endian, cosa sono?

fino a quando non succede che...
non dovete far collaborare due macchine diverse...
(un esempio lo vediamo al corso di Reti)

A cosa servono i numeri ottali o esadecimali?

Se non ci fossero dovremmo scriverli solo in BINARIO :D ?
E quanti errori faremmo (sia il prof sia noi)? risposta TANTI, TANTISSIMI!

Le operazioni & (and), | (or), ~ (not): come funzionano?

Intanto dobbiamo distinguere tra operazioni ARITMETICHE e LOGICHE

& (and), | (or), ~ (not) sono operazioni ARITMETICHE.

A & B	0x34 & 0xC5
0x34	00110100
0xC5	11000101
AND	00000100

A | B	0x34 | 0xC5
0x34	00110100
0xC5	11000101
OR	11110101

A	~A (NOT A)
0	1
1	0

"Gli informatici sono uomini che sussurrano ai processori" cit. Renzo Davoli :3

FLAGS
- è acceso?
- accendilo / spegnilo
- sono tutti accesi?
- ce n'è almeno uno acceso?

queste cose (giocando con i FLAGS) si fanno comodamente con questi operatori

prendiamo un flag di 8 byte dove:

Vogliamo scrivere:
'se ho fame vado a magnà'

```
if (flag & 0x08)
        mangia()
```

00001000 [ovvero controlla questo bit (quello a 1)]

per questo parliamo di MASCHERE

e non intendo quelle del teatro o le mascherine

le maschere in tipografia --> fogli con buchi x vedere determinate cose

falso è 0, vero è qualsiasi cosa diversa da 0.

if (flag & 0x80) 'oh cavolo' prima avevo scritto 0x80 e nessuno mi aveva
        mangia()   corretto... ma come si fanno le cose CORRETTE?

scriverei if (flag & HO_FAME) mangia() così è più leggibile :3

--> questo simbolo io (Renzo Davoli) lo chiamo SHARP!
quindi mi sentirete dire /sharp define/ invece che /... define/

define HO_FAME 1 << 3
ovvero definisco HO_FAME come la costante che ha
1 spostato a sinistra di 3 posizioni

Come si fa ad "accendere" bit? flag |= HO_FAME

dopo che ho mangiato [mangia()] vorrò spegnerlo...

Come si fa a "spegnere" bit? flag &= ~ HO_FAME

e se voglio dire: 'se HO_FAME or HO_SETE allora mangia()' cosa scrivo?
Come si fa a considerare solo alcuni bit di un intero (mascheramento)?

#define OK_PASTO (HO_FAME | HO_SETE)
if ((flag & OK_PASTO)==OK_PASTO) questo è MOLTO USATO

--> per approfondire cercate 'come spaccare i bit in 4' sulla wiki del corso

https://www.cs.unibo.it/~renzo/so/misc/bit4.pdf

(come ho trovato il link?)

sito di Renzo Davoli
https://www.cs.unibo.it/~renzo/

corso di Sistaemi Operativi
https://www.cs.unibo.it/~renzo/so/

wiki del corso
so.v2.cs.unibo.it --> https://so.v2.cs.unibo.it/wiki/index.php/Main_Page

'Esempi, Esercizi ed Esperimenti 2023/24'
https://so.v2.cs.unibo.it/wiki/index.php/Esempi,_Esercizi_ed_Esperimenti_2023/24

'Esempi sulla rappresentazione dei dati'
https://so.v2.cs.unibo.it/wiki/index.php/Esempi_sulla_rappresentazione_dei_dati

'Come spaccare i bit in 4'
https://www.cs.unibo.it/~renzo/so/misc/bit4.pdf

Come fa l'operazione ^ (xor)?

v ^= 0xA 1010 (A in binario)
          flippa questi bits (quelli a 1)

          in inglese si dice TOGGLE

Come si può usare il mascheramento per fare l'operazione di modulo di una potenza di 2?

Il MASCHERAMENTO può servire per fare l'operazione MODULO

v % 256
questa è l'operazione di MODULO
e farla è pesante...
(compilatore furbo se fa l'altro metodo)

v & 255
ovvero 'guarda solo gli ultimi 8 bit'
si fa in un ciclo di macchina

quindi...

dati A e B come si fa a controllare se alcuni bit di A sono accesi in B?

if (B & A)...

dati A e B come si fa a controllare se tutti i bit di A sono accesi in B?

if (B & A) == A)...

GIT --> mandato volantino sul gruppo telegram di quest'anno (2324)

link al gruppo telegram:
https://t.me/+HRseokr7sxFhOWUy

link al messaggio:
https://t.me/c/1933494044/103

inviato da @LLibera (t.me/llibera)

messaggio di descrizione:

🔗 link typeform per iscrizione (https://lr533gb3hpt.typeform.com/to/m5fzNCsW)
@admstaff_Chat per info/domande

(di cui Samu ha fatto reply per specificare e dare spoiler su 'TERMINALE AVANZATO' di cui dobbiamo ancora scegliere le date)

"GIT è QUASI il social per gli informatici :3" cit Renzo Davoli oggi 2023/10/05

Cosa significa oggi UNIX? Linux è UNIX? (domanda su telegram)

UNIX è ciò che è coerente con POSIX (o meglio certificato POSIX).
In questo senso Linux è uno UNIX

BASH base

Qual è la sintassi tipica dei comandi UNIX?

verbo, complementi, complemento oggetto (separato da spazi).

Cosa è un file system? (prima di file system dobbiamo parlare di FILE)

FILE
cosa è? vi ricordate le PRATICHE (commercialista / notarile)?

in inglese è intuitivo
'open a file' --> aprire una pratica
'close a file' --> chiudere una pratica

il FILE SYSTEM

folder --> "il faldone" (tutte le pratiche del anno X)

il file system è un Grafo Diretto Aciclico
(in inglese Direct Acyclic Graph --> DAG
https://en.wikipedia.org/wiki/Directed_acyclic_graph)

ARCHIVIO

UNIX usa il file sustem come strumento di naming. Cosa significa?

"UNIX usa il file system come strumento di naming."
significa che è un metodo di nomenclatura --> un metodo per dare nomi alle cose

"EVERYTHING IS A FILE"
compreso disco, dispositivi di input/output (tastiera --> keyboard, mouse ecc.)

il fatto che "EVERYTHING IS A FILE" è molto utile perchè così
le system call non hanno niente di specifico per i ...
quindi di 'specifico' (per esempio il disco) ha solo il nome (/dev/vda1).

Cosa sono la home directory e la current working directory? (su telegram)

current dir: informazione specifica di ogni processo, è il nodo del file system che si usa per trovare un file o una directory se il path è relativo

home dir: è la current dir al momneto del login. Ci si può ritornare mediante il comando 'cd' senza parametri. Dai pathname si può indicare la home dir con una ~ (tilde): e.g. ~/.bashrc qualunque sia la vostra current dir indica ill file .bashrc nella vostra home dir.

Qual è la struttura tipica del file system di UNIX? (su telegram)

Cosa sono i Pathname assoluti e relativi? (su telegram)

pathname assoluti, scandiscono l'albero del file system dalla radice e.g. /etc/passwd (iniziano con /). I pathname relativi non iniziano con / e calcolano la posizoine nel file system a partire dalla current dir.

Cosa fanno i comandi ls, echo, date, cat, more, less?

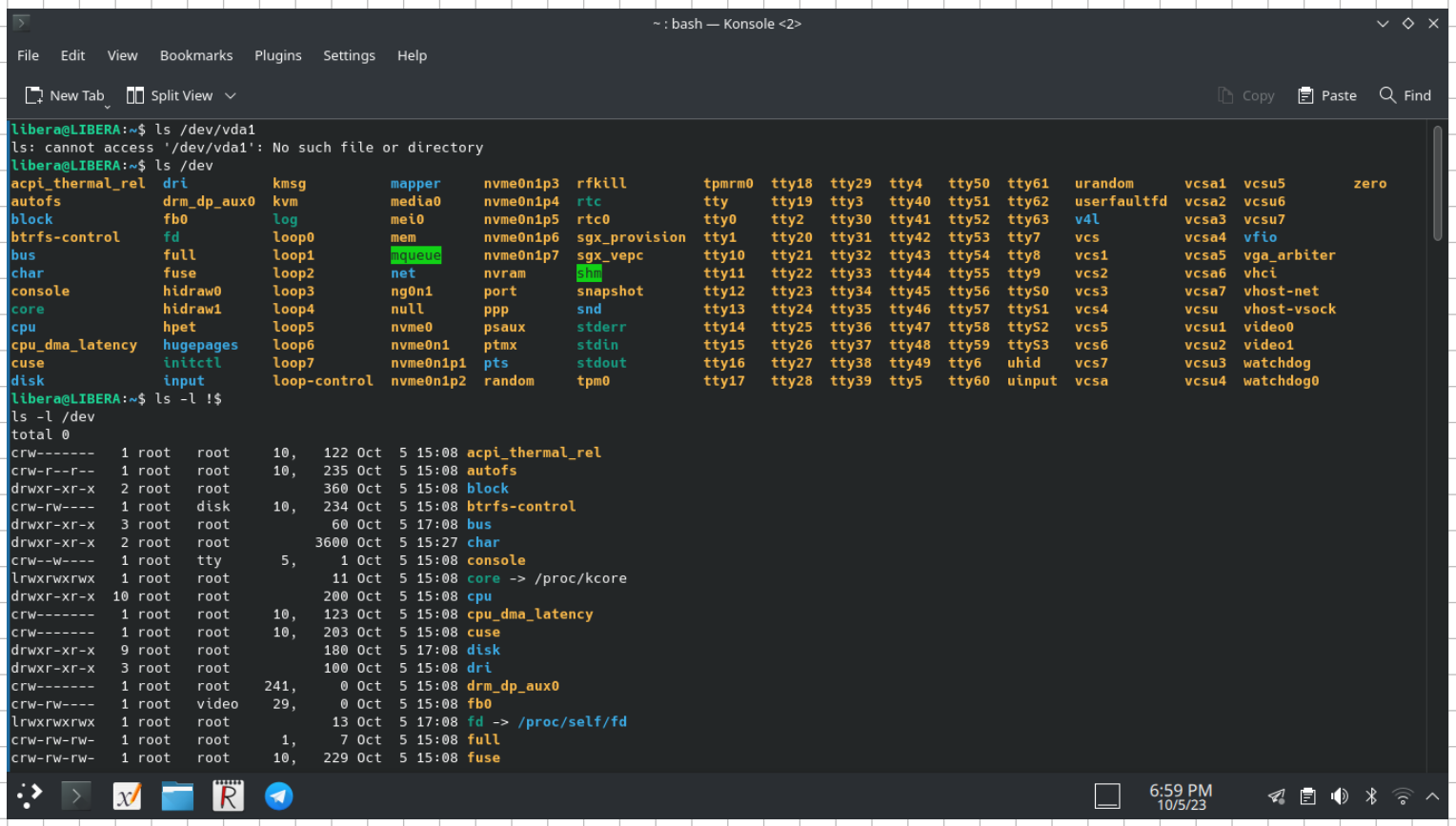
ls

```
libera@LIBERA:~$ ls
Desktop    Downloads  Pictures  Templates
Documents  Music      Public    Videos
```

ls -l

```
libera@LIBERA:~$ ls -l
total 32
drwxr-xr-x 2 libera libera 4096 Oct  5 11:18 Desktop
drwxr-xr-x 3 libera libera 4096 Oct  5 17:10 Documents
drwxr-xr-x 3 libera libera 4096 Oct  5 11:06 Downloads
drwxr-xr-x 2 libera libera 4096 Oct  2 12:50 Music
drwxr-xr-x 3 libera libera 4096 Oct  5 18:53 Pictures
drwxr-xr-x 2 libera libera 4096 Oct  2 12:50 Public
drwxr-xr-x 2 libera libera 4096 Oct  2 12:50 Templates
drwxr-xr-x 2 libera libera 4096 Oct  2 12:50 Videos
```

ls /dev/vda1



se /dev/vda1 non desse l'errore 'No such file or directory'
e facessi 'cat /dev/vda1' mi verrebbe l'errore 'Permission Denied', ovvero
non lo potrei fare pk si potrei leggere il disco
ma potrei anche scrivere facendo danni... questo ci fa introdurre il concetto
di utenti / root, ma in tanti anni di didattica ho notato che
'confondete root e utenti' con 'modo kernel e modo user' sono cose diverse!

ROOT UTENTI

vengono entrambi eseguiti in modo utente
sono un astrazione dati del sistema operativo

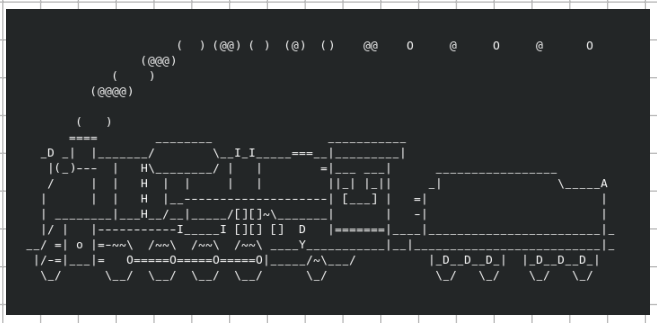
ROOT è "l'unico utente invenzione del Kernel", ovvero 'root' ha i permessi
tutti gli altri non li hanno.

```
libera@LIBERA:~$ ls /
bin   dev  home        initrd.img.old  lib32  libx32      media  opt   root  sbin  sys  usr  vmlinuz
boot  etc  initrd.img  lib             lib64  lost+found  mnt    proc  run   srv   tmp  var  vmlinuz.old
```

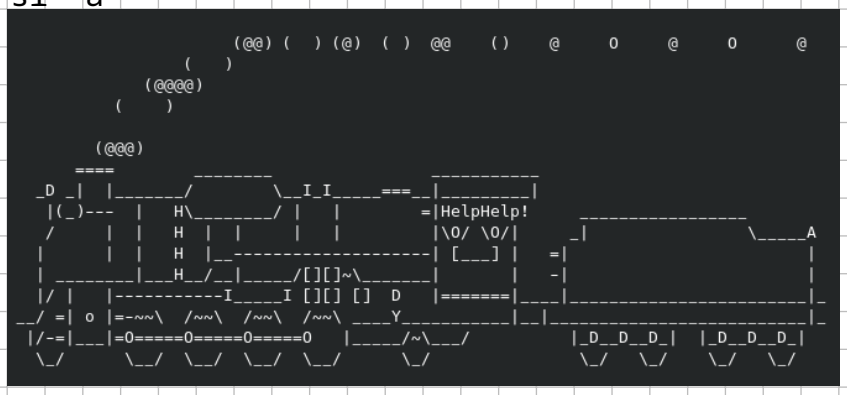
/bin --> eseguibili durante boot + librerie
/root --> home di root
/log --> log di sistema
/var --> archivi temporanei x i programmi
/tmp --> archivi temporanei x gli utenti
        /var vs /tmp? ad esempio in /var cè ...
        'mailbox degli utenti' di un programma di mail... spooler di stampa...
        mentre /tmp è nostro (gli utenti non possono scrivere dentro /var)
        da notare: /var/backups
        /var/log --> syslog --> demoni sempre attivi nel sistema
/usr
        /usr/games
        - fortunte (storicamente per aforismi, dopo ne hanno fatto
          una versione con parolacce (per imprecare senza sprecare intelletto
        (- cowsay (la mucca))

sl --> sapete come è nata la keyboard (tastiera) QWERTY?
        è nata per scrivere sulle telescriventi (macchine da scrivere)
        e per evitare martelletti che partono pk vicini incespichino
        per le frequenze + frequenti in inglese...
        il risultato? è demenziale per scrivere velocemente e BENE
        questo pk c'è il collo di bottiglia del PONTE ENCEFALICO
        tra l'emisfero sinistro e l'emisfero destro del cervello...
        ovvero ci sono buone probabilità di invertire l'ordine
        delle lettere pk c'è un problema di CONCORRENZA.
ma esiste un alternativa a questa tastiera che non ha questo problema?
        si esiste la DVORACK (it.wikipedia.org/wiki/Tastiera_semplificata_Dvorak)
        (c'è libertà) ma poi è difficile per le persone 'riprogrammarsi'.

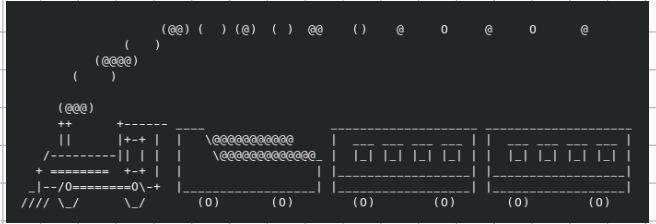
sl



sl -a



sl -l



sl -al

nota che con -a chiedono aiuto :3

video, linux day, Enrico Zini, 'metodi di cazzeggio avanzato utenti linux'
per caso è questo? https://youtu.be/L60IzQN9n-k?si=CHv-G280xAJ2BbqG

(ma ora torniamo a essere seri...)

SLASH '/' not REVERSE SLASH '\'

ogni processo ha la sua dir corrente

chdir c'è una syscall che permette di cambiare a quel processo lì

```
chdir(2)                    System Calls Manual                    chdir(2)

NAME
       chdir, fchdir - change working directory

LIBRARY
       Standard C library (libc, -lc)

SYNOPSIS
       #include <unistd.h>

       int chdir(const char *path);
       int fchdir(int fd);

   Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

       fchdir():
           _XOPEN_SOURCE >= 500
               || /* Since glibc 2.12: */ _POSIX_C_SOURCE >= 200809L
               || /* glibc up to and including 2.19: */ _BSD_SOURCE

DESCRIPTION
       chdir() changes the current working directory of the calling process to the directory specified in path.

       fchdir() is identical to chdir(); the only difference is that the directory is given as an open file descriptor.

RETURN VALUE
       On success, zero is returned.  On error, -1 is returned, and errno is set to indicate the error.

ERRORS
       Depending on the filesystem, other errors can be returned.  The more general errors for chdir() are listed below:

       EACCES Search permission is denied for one of the components of path.  (See also path_resolution(7).)

       EFAULT path points outside your accessible address space.
 Manual page chdir(2) line 1 (press h for help or q to quit)
```

```
libera@LIBERA:~$ man chdir
libera@LIBERA:~$ man cd
No manual entry for cd
```

cd è invece un comando interno della shell

echo ripetere parametri

ma a che serve poter ripetere i parametri?
la shell è anche un linguaggio esempio:

```
libera@LIBERA:~$ saluto=ciao
libera@LIBERA:~$ echo $saluto
ciao
```

diventa una print, serve per le variabili di environment --> strumento scripting

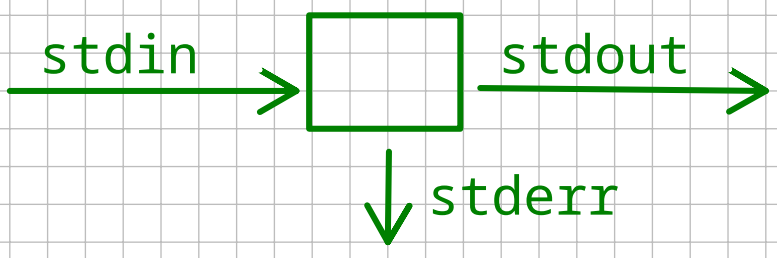
```
libera@LIBERA:~$ date
Sat Oct  7 09:45:33 PM CEST 2023
```

date

Cosa è e a cosa serve la ridirezione?

redirezione

usare i < o > in un comando shell si chiama REDIREZIONE



I file standard: stdin, stdout, stderr.
Cosa sono? Come si usano?

stdin --> standard input
stdout --> standard output
stderr --> standard error

^D --> Cntrl + D

cat

```
libera@LIBERA:~$ cat /etc/hostname
LIBERA
libera@LIBERA:~$ cat < /etc/hostname
LIBERA
libera@LIBERA:~$ cat > /etc/hostname
bash: /etc/hostname: Permission denied
libera@LIBERA:~$ su
Password:
su: Authentication failure
libera@LIBERA:~$ su
Password:
root@LIBERA:/home/libera# cat > /etc/hostname
DEBIAN
root@LIBERA:/home/libera# su libera
libera@LIBERA:~$
```

come si vede da questa sequenza di comandi...

/etc/hostname
indica il nome della macchina...

che io avevo settato male al installazione...

quindi ora riavvio il pc et voilà

```
libera@DEBIAN:~$
```

ma facciamo un altro esempio (quello fatto a lezione)

```
libera@DEBIAN:~$ cat > /tmp/saluti
scrivi sul file con il comando
cat > /tmp/saluti
poi scrivi qualcosa e chiudi con ^D
libera@DEBIAN:~$ cat < /tmp/saluti > hello
libera@DEBIAN:~$ cat hello
scrivi sul file con il comando
cat > /tmp/saluti
poi scrivi qualcosa e chiudi con ^D
libera@DEBIAN:~$
```

cat > FILE

scrive su FILE quello che hai scritto come stdin dopo aver premuto invio... (^D)

cat < INPUT > OUTPUT

fa la stessa cosa di

cp INPUT OUTPUT

poi c'è un comando bellissimo... "ho visto la luce" John Belushi, The Blues Brothers

^R (--> cntrl + R) per fare ricerca, ma anche edit della ricerca di un comando precedente!

mi serve un file lunghino... cat /etc/protocols

```
libera@DEBIAN:~$ cat /etc/protocols
# Internet (IP) protocols
#
# Updated from http://www.iana.org/assignments/protocol-numbers and other
# sources.
# New protocols will be added on request if they have been officially
# assigned by IANA and are not historical.
# If you need a huge list of used numbers please install the nmap package.

ip      0       IP              # internet protocol, pseudo protocol number
hopopt  0       HOPOPT          # IPv6 Hop-by-Hop Option [RFC1883]
icmp    1       ICMP            # internet control message protocol
igmp    2       IGMP            # Internet Group Management
ggp     3       GGP             # gateway-gateway protocol
ipencap 4       IP-ENCAP        # IP encapsulated in IP (officially ``IP'')
st      5       ST              # ST datagram mode
tcp     6       TCP             # transmission control protocol
egp     8       EGP             # exterior gateway protocol
igp     9       IGP             # any private interior gateway (Cisco)
pup     12      PUP             # PARC universal packet protocol
udp     17      UDP             # user datagram protocol
hmp     20      HMP             # host monitoring protocol
xns-idp 22      XNS-IDP         # Xerox NS IDP
rdp     27      RDP             # "reliable datagram" protocol
iso-tp4 29      ISO-TP4         # ISO Transport Protocol class 4 [RFC905]
dccp    33      DCCP            # Datagram Congestion Control Prot. [RFC4340]
xtp     36      XTP             # Xpress Transfer Protocol
ddp     37      DDP             # Datagram Delivery Protocol
idpr-cmtp 38    IDPR-CMTP       # IDPR Control Message Transport
ipv6    41      IPv6            # Internet Protocol, version 6
ipv6-route 43   IPv6-Route      # Routing Header for IPv6
ipv6-frag 44    IPv6-Frag       # Fragment Header for IPv6
idrp    45      IDRP            # Inter-Domain Routing Protocol
rsvp    46      RSVP            # Reservation Protocol
gre     47      GRE             # General Routing Encapsulation
esp     50      IPSEC-ESP       # Encap Security Payload [RFC2406]
ah      51      IPSEC-AH        # Authentication Header [RFC2402]
skip    57      SKIP            # SKIP
ipv6-icmp 58    IPv6-ICMP       # ICMP for IPv6
ipv6-nonxt 59   IPv6-NoNxt      # No Next Header for IPv6
ipv6-opts 60    IPv6-Opts       # Destination Options for IPv6
rspf    73      RSPF CPHB       # Radio Shortest Path First (officially CPHB)
vmtp    81      VMTP            # Versatile Message Transport
eigrp   88      EIGRP           # Enhanced Interior Routing Protocol (Cisco)
ospf    89      OSPFIGP         # Open Shortest Path First IGP
ax.25   93      AX.25           # AX.25 frames
ipip    94      IPIP            # IP-within-IP Encapsulation Protocol
etherip 97      ETHERIP         # Ethernet-within-IP Encapsulation [RFC3378]
encap   98      ENCAP           # Yet Another IP encapsulation [RFC1241]
#       99                      # any private encryption scheme
pim     103     PIM             # Protocol Independent Multicast
ipcomp  108     IPCOMP          # IP Payload Compression Protocol
vrrp    112     VRRP            # Virtual Router Redundancy Protocol [RFC5798]
l2tp    115     L2TP            # Layer Two Tunneling Protocol [RFC2661]
isis    124     ISIS            # IS-IS over IPv4
sctp    132     SCTP            # Stream Control Transmission Protocol
fc      133     FC              # Fibre Channel
mobility-header 135 Mobility-Header # Mobility Support for IPv6 [RFC3775]
udplite 136     UDPLite         # UDP-Lite [RFC3828]
mpls-in-ip 137  MPLS-in-IP      # MPLS-in-IP [RFC4023]
manet   138                     # MANET Protocols [RFC5498]
hip     139     HIP             # Host Identity Protocol
shim6   140     Shim6           # Shim6 Protocol [RFC5533]
wesp    141     WESP            # Wrapped Encapsulating Security Payload
rohc    142     ROHC            # Robust Header Compression
ethernet 143    Ethernet        # Ethernet encapsulation for SRv6 [RFC8986]
# The following entries have not been assigned by IANA but are used
# internally by the Linux kernel.
mptcp   262     MPTCP           # Multipath TCP connection
libera@DEBIAN:~$
```

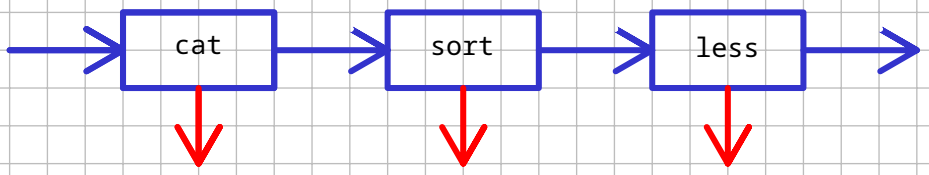
more con more ci si muove solo in basso (è storico)

less con less ci si muove in entrambe le direzioni

Cosa è e a cosa serve la pipe?

il simbolo '|' indica la presenza di una PIPE, ovvero:

```
cat /etc/protcols | sort | less
                  ^      ^
                pipe   pipe
```



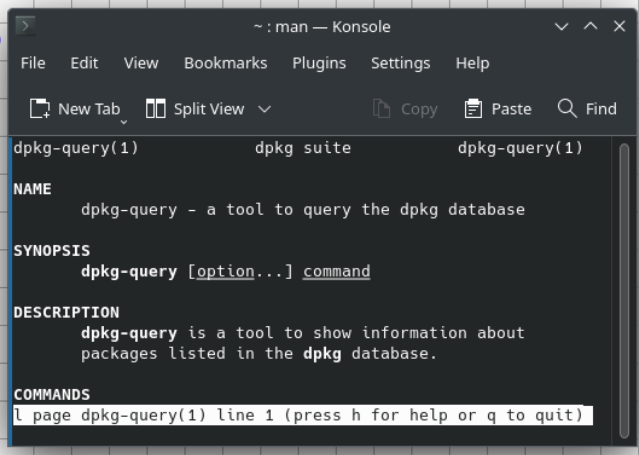
la pipe non ridireziona STDERR ! (se no sarebbe un casino)

domanda mia durante la pausa... prof ma come si scaricano i 'games' (sl, cowsay, fortune...) su debian?

dpkg-query -S /usr/bin/cowsay

```
libera@DEBIAN:~$ man dpkg-query
libera@DEBIAN:~$ dpkg-query -S /usr/bin/libreoffice
libreoffice-common: /usr/bin/libreoffice
```

libreoffice-common è quindi il nome del package!



Cosa fanno i comandi touch, cp, mkdir, rmdir, rm, ln?

touch crea file vuoto o aggiorna il tempo dell'ultima modifica (senza modificare il contenuto del FILE)

mv abbreviazione di move, rename

legenda del Golem
it.wikipedia.org/wiki/Golem

rm (unlink) la systemcall per... rimuovere l'ultimo nome associato al file

mkdir crea una directory

rmdir rimuove una directory vuota

per rimuovere una directory piena si fa 'rm -rf PATH' ma non fate mai 'rm -rf /'
(io l'ho fatto e mi chiedeva un parametro aggiuntivo tipo --no-warn-root (non ricordo),
ma cmq avrebbe cose eliminato cose che in quel momento usava... tipo la shell
e i processi aperti ora... quindi cmq non fatelo!)

cp copia, esempio 'cp INPUT OUTPUT'

ln link hard, per link simbolico si usa 'ln -s'

A cosa serve il comando man? Quali sono le sezioni di man?

man è il manuale, serve per chiedersi come si usano ...
è ricorsivo quindi puoi fare il comando 'man man' per sapere come si usa il manuale

eseguendo:
man
What manual page do you want?
For example, try 'man man'.

eseguendo:
man man
vediamo:
MAN(1) –> la pagina
Manual pager utils –> il significato della pagina 1

in DESCRIPTION vediamo:

man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.

The table below shows the section numbers of the manual followed by the types of pages they contain.

1 Executable programs or shell commands
2 System calls (functions provided by the kernel)
3 Library calls (functions within program libraries)
4 Special files (usually found in /dev)
5 File formats and conventions, e.g. /etc/passwd
6 Games
7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7), man-pages(7)
8 System administration commands (usually only for root)
9 Kernel routines [Non standard]

A manual page consists of several sections.

Conventional section names include NAME, SYNOPSIS, CONFIGURATION, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUE, ERRORS, ENVIRONMENT, FILES, VERSIONS, CONFORMING TO, NOTES, BUGS, EXAMPLE, AUTHORS, and SEE ALSO

quindi le sezioni sono:
- NAME
- SYNOPSIS
- CONFIGURATION
- DESCRIPTION,
- OPTIONS
- EXIT STATUS
- RETURN VALUE
- ERRORS
- ENVIRONMENT
- FILES
- VERSIONS,
- CONFORMING TO
- NOTES
- BUGS
- EXAMPLE
- AUTHORS
- SEE ALSO

```
libera@DEBIAN:~$ mkdir CARTELLA
libera@DEBIAN:~$ cd CARTELLA/
libera@DEBIAN:~/CARTELLA$ touch file
libera@DEBIAN:~/CARTELLA$ ls
file
libera@DEBIAN:~/CARTELLA$ mv file renamed-file
libera@DEBIAN:~/CARTELLA$ ls
renamed-file
libera@DEBIAN:~/CARTELLA$ cp renamed-file ciao
libera@DEBIAN:~/CARTELLA$ ls
ciao  renamed-file
libera@DEBIAN:~/CARTELLA$ ln -s ciao link
libera@DEBIAN:~/CARTELLA$ ls
ciao  link  renamed-file
libera@DEBIAN:~/CARTELLA$ ls -l
total 0
-rw-r--r-- 1 libera libera 0 Oct  9 18:32 ciao
lrwxrwxrwx 1 libera libera 4 Oct  9 18:33 link -> ciao
-rw-r--r-- 1 libera libera 0 Oct  9 18:31 renamed-file
libera@DEBIAN:~/CARTELLA$ rm ciao
libera@DEBIAN:~/CARTELLA$ ls -l
total 0
lrwxrwxrwx 1 libera libera 4 Oct  9 18:33 link -> ciao
-rw-r--r-- 1 libera libera 0 Oct  9 18:31 renamed-file
libera@DEBIAN:~/CARTELLA$ cd ..
libera@DEBIAN:~$ rm -rf CARTELLA/
libera@DEBIAN:~$ cd CARTELLA
bash: cd: CARTELLA: No such file or directory
libera@DEBIAN:~$
```

```
ln _______ __________
ls -l !*

ls -l /tmp/{hostname,newhost}

DUPLING LINK
ovvero link 'penzolante'

ls -d
    directory
```

who — show who is logged on

```
libera@DEBIAN:~$ who
libera   tty1         2023-10-07 22:13
libera   pts/0        2023-10-07 22:13 (:1)
libera   pts/1        2023-10-09 18:57 (:1)
libera   pts/2        2023-10-09 18:35 (:1)
libera@DEBIAN:~$ man who
```

rendere + difficili le profilazioni...

Cosa fanno i comandi file, whoami, groups?

file — determine file type

whoami — print effective user name

groups — print the groups a user is in

Quale è il significato dei singoli elementi delle righe ottenute con "ls -l"?

```
libera@DEBIAN:~$ ls -l
total 152
-rw-r--r-- 1 libera libera 117553 Oct  7 22:33 consensus.png
drwxr-xr-x 2 libera libera   4096 Oct  5 11:18 Desktop
drwxr-xr-x 3 libera libera   4096 Oct  6 15:05 Documents
drwxr-xr-x 3 libera libera   4096 Oct  9 10:15 Downloads
-rw-r--r-- 1 libera libera     85 Oct  8 19:15 hello
drwxr-xr-x 2 libera libera   4096 Oct  2 12:50 Music
drwxr-xr-x 4 libera libera   4096 Oct  9 18:36 Pictures
drwxr-xr-x 2 libera libera   4096 Oct  2 12:50 Public
drwxr-xr-x 2 libera libera   4096 Oct  2 12:50 Templates
drwxr-xr-x 2 libera libera   4096 Oct  2 12:50 Videos
libera@DEBIAN:~$ ls -d
.
libera@DEBIAN:~$
```

permessi

numero di link che puntano a quel 'file' (e dir)

utente, gruppo

dimensione

data (mese, giorno, ora)

nome file

e con 'ls -d' si listano anche i file nascosti tra cui vediamo la dir '.'

Come si cambiano i permessi di un file o una directory?

chmod — change file mode bits

Link fisici e simbolici: cosa sono e quali sono i comandi per crearli?

link fisici   --> ln
link simbolici --> ln -s

```
libera@DEBIAN:~$ man chmod
libera@DEBIAN:~$ cat hello.c
#include <stdio.h>
int main() {
printf("ciao mondo!\n");
}
libera@DEBIAN:~$ file *.c
hello.c: C source, ASCII text
libera@DEBIAN:~$ file *.*
consensus.png: PNG image data, 858 x 657, 8-bit/color RGB, non-interlaced
hello.c:       C source, ASCII text
libera@DEBIAN:~$ file [0-9].c
[0-9].c: cannot open `[0-9].c' (No such file or directory)
libera@DEBIAN:~$ ls hello.c
hello.c
libera@DEBIAN:~$ ls -l hello.c
-rw-r--r-- 1 libera libera 59 Oct  9 19:03 hello.c
libera@DEBIAN:~$ chmod u+x hello.c
libera@DEBIAN:~$ ls -l hello.c
-rwxr--r-- 1 libera libera 59 Oct  9 19:03 hello.c
libera@DEBIAN:~$ gcc hello.c
libera@DEBIAN:~$ ls
a.out          Desktop    Downloads  Music     Public     Videos
consensus.png  Documents  hello.c    Pictures  Templates
libera@DEBIAN:~$ ./a.out
ciao mondo!
libera@DEBIAN:~$ chmod u-x hello.c
libera@DEBIAN:~$ ls -a
.              .bashrc        Documents   .gnupg      .mozilla  .ssh
..             .cache         Downloads   .gtkrc-2.0  Music     Templates
a.out          .config        .face       hello.c     Pictures  Videos
.bash_history  consensus.png  .face.icon  .lesshst    .profile
.bash_logout   Desktop        .gitconfig  .local      Public
libera@DEBIAN:~$
```

file nascosti
che cominciano con .

Esecuzione in foregound e background, cosa significa?

esempio... un comando che dura un po di tempo...

sleep 5 --> dormi 5 secondi
sleep 5 & --> la & indica che il comando 'sleep 5' in questo caso deve essere lanciato
in 'background'
tutti i comandi e ... possono essere lanciati in background con & ? si!

cosa fa esattamente & ? la shell lancia ... e lo scollega

jobs --> lista i comandi in background
bg --> background
fg --> foreground

Cosa fanno i comandi find, grep?

mi sa che non ne ha parlato...

torniamo al C [...]

prin_C_ples sulla wiki

tutti i file di oggi li trovate al link:

https://www.cs.unibo.it/~renzo/so/lecture_examples2324/