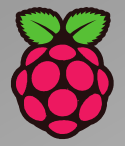
INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Raspberry Pi a LNL

Raspberry Pi a LNL



Uso della Pi:

La programmazione…

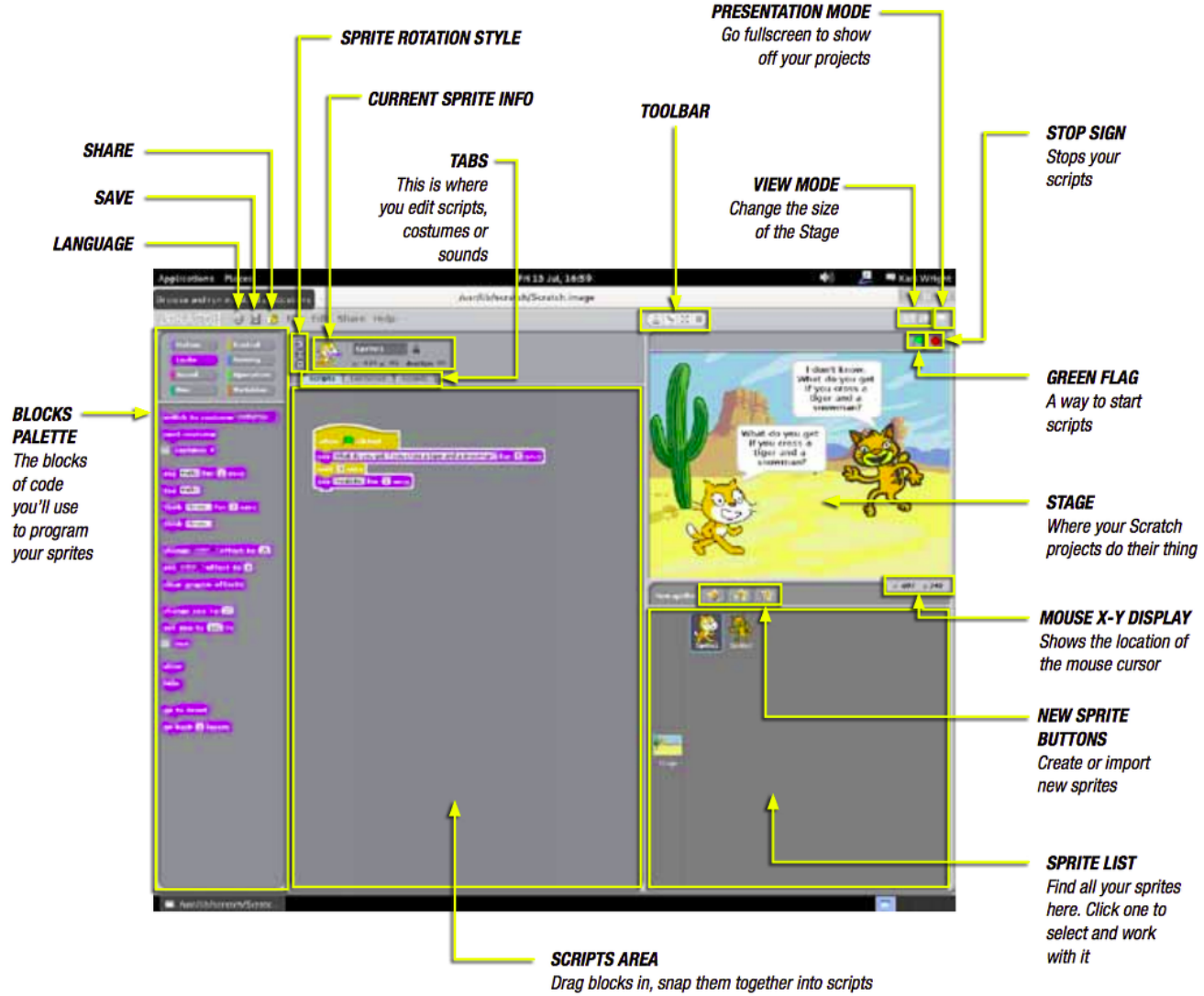
INFN
LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Scratch 1

Scratch è un ambiente grafico di programmazione (drag & drop) per creare animazioni e giochi.
È un ottimo modo per iniziare a programmare su Pi con bambini e ragazzi,
Senza scrivere codice.

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Scratch 2

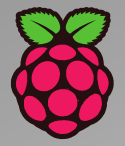


C (1)

Imparare a programmare in C ...

Lo trovate qui:
https://www.raspberrypi.org/magpi/issues/essentials-c-v1/

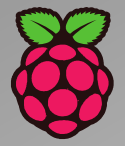
C (2)



BENVENUTI nel mondo del C-code !

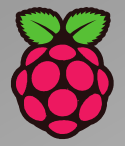
Il linguaggio di programmazione C è stato inventato nei primi anni 1970, e da allora è diventato uno dei linguaggi di programmazione più diffusi e ampiamente usati.
È utilizzato da programmatori professionisti, da dilettanti, su progetti per casa, e nell'industria. Viene usato per programmare di tutto: dal piccolo microcontrollori utilizzati in orologi e tostapane fino a enormi sistemi software.
È scritto in C gran parte di Linux e dei suoi tools (e Raspbian stesso).

C (3)



Il C è un linguaggio molto potente ma semplice: ha solo 20 parole chiave, ma diverse grandi librerie di funzioni aggiuntive, usabili al bisogno.
Altri linguaggi, come Python, sono interpretati; nei linguaggi interpretati il codice viene eseguito direttamente.
C invece è un linguaggio compilato. Questo significa che il codice (sorgente), non viene mai eseguito direttamente: viene prima “dato in pasto” a un programma chiamato compilatore che lo converte in un codice (eseguibile) comprensibile al computer.

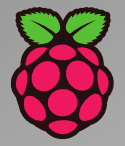
C (4)



L'uso del compilatore può sembrare complesso, ma ha alcuni grandi vantaggi:

1. non è necessario avere una copia di C (compilatore) su ogni computer, il programma compilato, (eseguibile) è autonomo e funziona anche senza compilatore;
2. durante la compilazione vengono trovati gli errori (di sintassi) ed è necessario correggerli tutti per produrre l'eseguibile, il compilato è sintatticamente corretto;
3. il codice compilato è generalmente molto più efficiente di quello interpretato (funziona meglio).

C (5) Ciao Mondo!

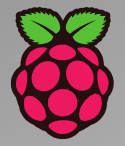


Si consideri il primo programma che tutti scrivono in qualsiasi linguaggio di programmazione, quella che stampa 'Ciao Mondo!' sullo schermo. Per inciso, la tradizione del “Ciao Mondo” è stata introdotta per la prima volta proprio con la documentazione originale del C-language. In un file ciao.c (con Geany) si scrive il testo:

```
#include <stdio.h>
void main (void)
{
/* A print statement */
printf ("Ciao Mondo!\n");
}
```

Per compilare il programma: > gcc –o ciao ciao.c

C (6) InOut



Lettura e interpretazione di dati forniti dall'utente (InOut)

```
#include <stdio.h>
void main (void)
{
char name[256];
int age;
printf ("Come ti chiami?\n");
scanf ("%s", name);
printf ("Ciao, %s. Quanti anni hai?\n", name);
scanf ("%d", &age);
printf ("Bene, %s, hai SOLO  %d anni ! ...\n", name, age);
}
```

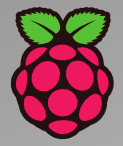
Python 1

Python è un meraviglioso e potente linguaggio di programmazione, facile da usare (da leggere e da scrivere). Con Raspberry Pi Python consente di collegare i progetti al mondo reale, a sensori e attuatori!

Python 2

Si inizia aprendo IDLE dal desktop.
IDLE è un ambiente di sviluppo per Python. Python 3 è la versione più recente ed è Quella consigliata, tuttavia Python 2 è disponibile per le vecchie applicazioni.

Python 3



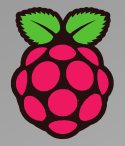
Per cominciare …
Si possono usare le variabili se necessario, ma python (IDLE) può anche essere usato come una calcolatrice, direttamente con i numeri. Ad esempio:

```
>>> 1 + 2
3

>>> nome = "Maria"
>>> "Ciao" + nome
'Ciao Maria'
```

IDLE ha anche l'evidenziazione della sintassi integrata e supporto per il completamento automatico.
La documentazione completa per Python è disponibile su python.org/doc

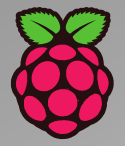
Altri tool su Pi 1



Mathematica

Mathematica è uno strumento utilizzato nelle scienze, in matematica, in informatica e ingegneria che risale al 1988.
Si tratta di software proprietario, ma che può essere utilizzato gratuitamente sul Raspberry Pi su Raspbian dal 2013.
(Guida introduttiva di Mathematica a:
https://projects.raspberrypi.org/en/projects/getting-started-with-mathematica)

Altri tool su Pi 2



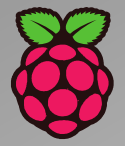
Sonic Pi

Sonic Pi è un ambiente di programmazione open source, progettato per creare suoni in modo “live” da Sam Aaron(Università di Cambridge). Per usarlo occorre avere un paio di cuffie o un altoparlante.
Guida introduttiva a Sonic:
https://projects.raspberrypi.org/en/projects/getting-started-with-sonic-pi)

…

```
play 60 sleep 0.5
play 62 sleep 0.5
play 64 sleep 0.5
play 60 sleep 0.5
```

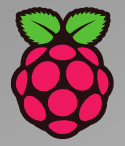
Raspberry Pi a LNL



Uso della Pi:

Shell e relativi comandi …

Linux (Raspbian)

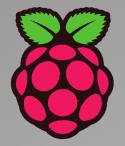


LINUX

Alcuni elementi base di Linux e i comandi essenziali per gestire la Pi, il suo file system, gli utenti e i processi (i programmi in esecuzione):

- Filesystem (home, SD) e backups
- Strumenti per l'uso: comandi, editors, gestione utenti e il superuser (root), cron e crontab (gestione di azioni periodiche), la shell bash e gli scripts (file di comandi).

Linux – File System

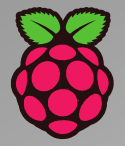


È importante conoscere la struttura del File System di LINUX: dove sono conservati i file, ovvero le aree con i dati e i programmi.

Filesystem (F.S.) di LINUX: comprende 2 aree

- **Computer F.S**.: il set di cartelle con programmi e dati comuni, comprende le *homes* degli utenti.
- **Home**: la cartella dell'utente, con dati e programmi personali (di default dell'utente pi).

File System - HOME



Dopo il boot e quando si apre un terminale o si accede alla Pi da ssh, si è nella cartella principale dell'utente, la **Home**;

questa è situata in /**home**/**pi** (assumendo l'utente standard pi),

Qui è dove si trovano i file e le cartelle dell'utente pi.

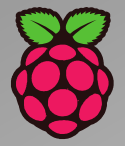
Il desktop dell'utente si trova in una cartella Desktop, insieme, a volte, ad altri file e cartelle, create dall'utente.

Per navigare nella cartella principale, dal terminale, scrivere semplicemente *cd* e premere invio.

Questo è l'equivalente di *cd /home/pi* , (dove pi è il nome utente).

Per sapere quali sono i file in una cartella usare il comando *ls*.

BACKUPs



BACKUP

Sono vivamente consigliati backup regolari di tutti i file importanti.

A volte conviene fare copie non solo dei file personali, ma anche di file di sistema e/o di configurazione e di software installato, o persino dell'intero sistema.

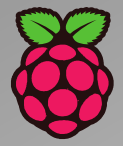
BACKUP di HOME

Per il backup della home si intende il salvataggio di tutti i file della cartella /home/pi, ovvero dei file utente.

BACKUP di SISTEMA (della SD)

Per il backup della SD si intende il salvataggio di tutto ciò che è contenuto nella SD, compresi i file utente in /home/pi.

BACKUP di HOME



BACKUP di HOME

Per il backup della home si può usare il comando tar, creare un archivio della cartella /home/pi e conservarne una copia su memoria USB, su PC o nel cloud. Per fare questo si usa il comando:

cd /home/ ; sudo tar czf pi_home.tar.gz pi

Questo crea un archivio tar con nome pi_home.tar.gz in /home. Si deve poi copiare questo file su memoria USB o trasferirlo su un'altra macchina sulla rete (SFTP), o in cloud.

AUTOMAZIONE

Si può preparare uno script *Bash* per eseguire automaticamente ciascuno di questi processi ed farlo eseguire periodicamente con *cron*.

BACKUP della SD 1

BACKUP dell' Immagine (personalizzata) della scheda SD

Potrebbe essere saggio conservare una copia dell'intera immagine della scheda SD, per poterla ripristinare in caso di smarrimento o danno.

Con Linux, sul PC e con la SD nello slot da SD si usa il comando:

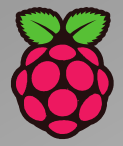
sudo dd bs=4M if=/dev/sdb of=raspbian_backup.img

Questo crea un file immagine usabile per preparare un'altra scheda SD, e mantenere esattamente gli stessi contenuti e le stesse impostazioni. Per ripristinare o clonare su un'altra SD, e utilizzare il comando *dd* in senso opposto:

sudo dd bs=4M if=raspbian_backup.img of=/dev/sdb

ATTENZIONE all'uso dei corretti file di input (if) e output (of)

BACKUP della SD 2



BACKUP compresso della SD

I file dei backup della SD possono essere molto grandi e vale la pena di comprimerli prima di memorizzarli. Per comprimere il backup di una SD si può reindirizzare l'output di dd a gzip (uility di compressione).

sudo dd if=/dev/sdb bs=4M | gzip > raspbian_backup.img.gz

Per ripristinare (scomprimere il file), si invia l'output di gunzip a dd:

gunzip --stdout raspbian_backup.img.gz | sudo dd bs = 4M di =/ dev/sdb

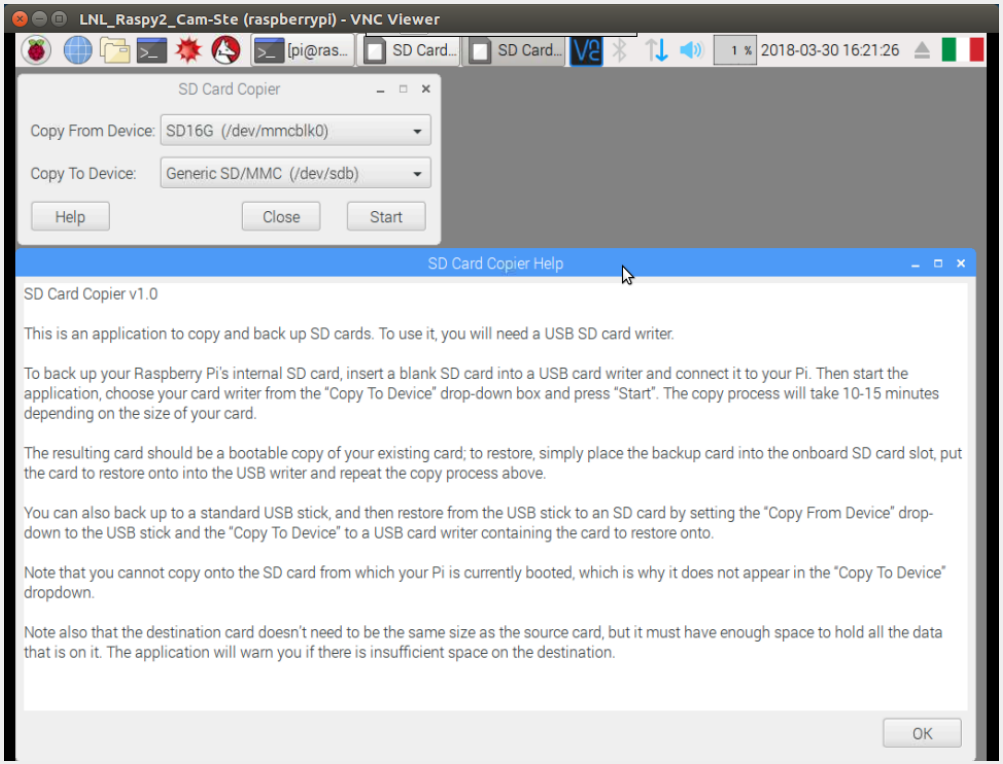
BACKUP DI Pi su memoria USB

Il backup della SD può essere eseguito anche da Pi con l'ausilio di una memoria flash USB abbastanza grande: di almeno 8GB senza compressione, di almeno 4GB per il file compresso.

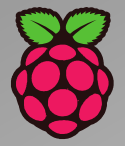
BACKUP della SD 3

BACKUP della SD da Pi

Su Raspbian è presente un tool di salvataggio della immagine personalizzata della SD, pur di connettere un USB card writer a una porta usb della Pi, con una seconda SD abbastanza grande.



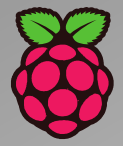
Strumenti per Linux



STRUMENTI BASE PER GESTIONE E USO DI LINUX

- Comandi: alcuni dei comandi più utili e frequenti.
- Utenti: creazione e cancellazione degli utenti, il superuser root
- Attività periodiche: Cron/Crontab
- Editors: alcuni text editors disponibili per Raspbian
- Interprete dei comandi (shell): scripts e inizializzazioni

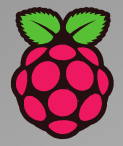
Comandi 1



Comandi di di shell, si danno solo da terminale

1. Per informazioni su comandi e programmi:
   whereis : indica la posizione del comando o programma nel f.s.
   man command : fornisce sintassi e opzioni per command
2. Per la posizione nel file-system:
   pwd : indica la posizione
   cd : consente di cambiare posizione
3. Per i files:
   ls, ls -l : danno la lista di file e cartelle nella posizione corrente
   rm filename : cancella il file filename
   cp sorgente destinazione : copia il file sorgente in destinazione
   mv sorgente destinazione : rinomina sorgente in destinazione
   find : cerca un file nel filesystem

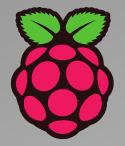
Comandi 2



4. Per informazioni sul sistema:
   uname -a : fornisce tutte le informazioni sul sistema in uso
   hostname : visualizza il nome corrente del sistema
5. Per la Rete (Networking) :
   ping : utilizzato per verificare se è possibile comunicare con un altro host (esempio ping 192.168.1.40)
   ifconfig : si usa ifconfig per visualizzare i dettagli di configurazione di rete per il il sistema in uso
6. Per la gestione degli utenti (in Raspbian solo da linea di comando) :
   sudo adduser nuovoutente : consente di creare il nuovoutente, con la sua password
   sudo userdel -r nuovoutente : consente di cancellare il nuovoutente, con la sua home

Osservazione importante: l'utente di default pi (con password di default raspberry) è normalmente un sudoer, ovvero abilitato a prendere il privilegi di superutente (root) e a diventare root con il comando *sudo su*, mentre questo non vale per gli utenti "normali".

Superutente e SUDO



SUPERUSER (ROOT) - SUDO

I sistema operativo Linux è multi-utente, ciò significa che consente l'accesso e l'utilizzo simultaneo del computer a più utenti. Per proteggere il computer (e la privacy di ciascun utente), le capacità degli utenti sono limitate.

La maggior parte degli utenti può eseguire la maggior parte dei programmi e salvare e modificare i file archiviati solo nella propria home. Gli utenti "normali" non sono autorizzati a modificare i file in altre cartelle, diverse dalle proprie o un file di sistema.

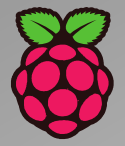
C'è però un utente speciale di Linux, il superuser (root) che ha accesso illimitato al computer e può fare quasi tutto (anche cancellare il sistema !!!). Quindi: usare sudo con prudenza !!!

Attività a orario (cron)

Programmazione di attività a orario con CRON

Cron è uno strumento per configurare attività periodiche pianificate a data e ora. Le attività sono comandi o script, ad esempio il backup della home dell'utente ogni giorno ad una data ora.
Il comando crontab serve a modificare l'elenco delle attività pianificate ed specifico per ogni utente; ogni utente ha il suo crontab.
Esiste anche una interfaccia grafica per cron, che può essere installata con il comando:
sudo apt-get install gnome-schedule
Si consulti il manuale (*man cron; man crontab*) per le modalità d'uso di cron e crontab.

Editors



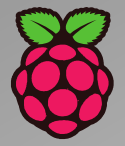
Su Raspbian sono presenti vari editor di testo. Alcuni sono facili da usare ma hanno funzionalità limitate; altri richiedono più tempo di apprendimento, ma offrono funzionalità più ampie.

Leafpad è un semplice editor grafico che si apre in una finestra come una normale applicazione. Permette l'uso del mouse e della tastiera e prevede l'evidenziazione della sintassi.

Nano è un editor da terminale. È installato di default, quindi si può usare *nano somefile.txt* per modificare un file e comandi da tastiera come Ctrl + O per salvare e Ctrl + X per uscire.

Vi è un editor da terminale molto vecchio (1976), preinstallato su Raspbian. Complicato da usare, ma potente. Ne esiste una versione più recente (vim) con interfaccia grafica (gvim), da intallare.

Shell



I comandi dati su terminale sono letti, interpretati ed eseguiti da una SHELL. Su Linux (e Raspbian) esistono diverse shell, leggermente differenti fra loro (sh, csh, tcsh, ksh, bash). Lo standard è BASH
Bash (ma anche le altre shell) è un vero e proprio linguaggio di programmazione, con strutture di controllo e funzioni predefinite.
Più comandi possono essere combinati insieme in un file (SCRIPT) che può quindi essere eseguito.

Esempio di script:

```
while true
do
echo Viva il Raspberry Pi! ; sleep 1 ;
done
```