

Open Source, Free Software e Programmazione

Il resto del Pinguino

0

Linux su una
chiavetta USB

Come installare
un sistema Linux
completo su una
chiavetta USB da
portare sempre
con sé

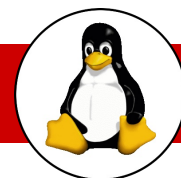
MeteoMachine
Linux

Prediciamo che
tempo farà con
Linux

Tiny Core Linux

Mini-distro per
rendere nuova
linfa a computer
datati o vecchio
hardware non
più usabile con
le distro odierne





Scrivere è sempre difficile. Lo è ancor di più se si tratta di un editoriale. E la fatica raddoppia quando devi scrivere un editoriale per annunciare la nascita di un nuovo progetto, come in questo caso, in cui devo annunciare l'uscita del numero 0 de "*Il resto del Pinguino*".

Che cos'è "*Il resto del Pinguino*"? E' una e-zine, una rivista in formato digitale dedicata al mondo del software libero, o *free software* secondo la locuzione anglosassone voluta da **Richard Stallman**, della Free Software Foundation. Ma qual è il significato di software libero? Per software libero si intende un software di cui l'utente finale può godere al 100%: ha accesso ai sorgenti, può modificarli a proprio piacimento, può distribuirlo. La libertà più totale. E questa e-zine è dedicata proprio a tale software: una rivista libera, dedicata al software libero e realizzata con strumenti liberi.

Ma da chi è scritta? Dalla comunità italiana dei programmatori Gambas. Gambas, per chi non lo conoscesse, è un potente linguaggio di programmazione visuale basato sul BASIC con il quale è possibile creare applicazioni con interfacce grafiche in pochi, semplici, gesti. La comunità italiana ha un sito di riferimento, www.gambas-it.org. Da alcuni suoi membri è nata l'idea di questa e-zine, un prodotto che, speriamo, sia di vostro gradimento. Da parte nostra ci metteremo passione e dedizione, cercando di sfornare un prodotto valido, ricco di contenuti interessanti e stimolanti, graficamente gradevole. Da parte vostra chiediamo solo il download per la lettura dell'e-zine. Se però vi sentite all'altezza, potete anche collaborare alla sua stesura (d'altronde, è una e-zine libera, no?) inviando articoli sull'argomento che più vi piace, basta che siano scritti bene, inerenti al mondo open-source (recensioni di programmi o siti utili ai programmatori di ogni linguaggio, guide varie, tutorial di ogni genere, prove su strada di distro più o meno conosciute, ecc...). Potete inviare il materiale a ilrestodelpinguino@gambas-it.org. Tutto il materiale sarà vagliato e quello ritenuto interessante pubblicato sulla e-zine.

Infine una parola su "**Pallinux**", la mascotte che avete visto in copertina. Essa è disegnata dal nostro Fabio "Pixel" Colinelli e ci accompagnerà in questa avventura. Il suo nome significa "*Ho il pallino per Linux*".

Per ora è tutto. Vi lascio alla lettura di questo numero di apertura.

Leonardo "leo72" Miliani

REDAZIONE

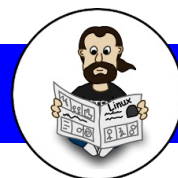
Impaginazione: Leonardo "Leo" Miliani - leonardo@leonardomiliani.com

Coordinazione articolisti: Francesco "Ceskho" Apuzzese - cescoap@gmail.com

Grafica: Fabio "Pixel" Colinelli - pixel.ubuntu@gmail.com

COLLABORAZIONI

Se volete pubblicare un articolo sulla rivista, inviate la vostra opera (preferibilmente in forma di pacchetto compresso) all'indirizzo ilrestodelpinguino@gambas-it.org in formato testuale o, eventualmente, in formato ODT (OpenOffice), con eventuali foto. Nota: l'invio del materiale non obbliga la redazione alla sua pubblicazione. Sarà nostro insindacabile giudizio pubblicare o meno l'articolo ricevuto con eventuali tagli dettate da motivi tipografici. Con l'invio del materiale l'autore dà tacito consenso all'utilizzo della sua opera ai fini della pubblicazione. Si ricorda che saranno pubblicati solo gli articoli inerenti i temi trattati dalla rivista.



Il resto del Pinguino

N° 0 - Marzo 2010

5. Linux su una chiavetta USB - di Leonardo Miliani

Come configurare una chiavetta USB per portare sempre con sé Linux



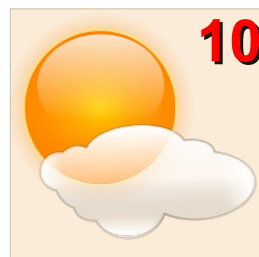
Linux su USB

8. La funzione Split di Gambas - di Pietro Catania

Come dividere una stringa utilizzando gli strumenti di Gambas

10. MeteoMachine Linux - Francesco Apuzzese

Piovono righe e righe di codice!



MeteoMachine
Linux

12. Chi cerca trova - di Mr. X

Prova della mini-distro Tiny Core Linux

14. Su che cosa e come programmare - di Picavbg

Il computer come intelligente strumento di sussidio alle attività umane

14. La cattedrale ed il bazar - di Eric S. Raymond

Due stili di sviluppo del software a confronto: il modello "cattedrale" e quello "bazar"



Tiny Core Linux

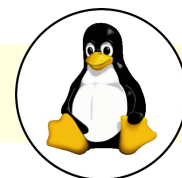
Hackles



<http://hackles.org>

Copyright © 2001 Drake Emko & Jen Brodzik

OpenSource



Linux su una chiavetta USB

Come configurare una chiavetta USB per avere il Pinguino sempre con sé

Mi sono sempre scontrato con un problema apparentemente insormontabile: installare Linux su una chiavetta USB!

Quanti di voi ci hanno provato? Tanti, penso... E quanti di voi hanno avuto problemi? Tanti lo stesso, vero?.

Mi son sempre chiesto il perché dei miei insuccessi durante i tentativi di installare le varie distribuzioni per chiave USB quali Damn Small Linux e simili. Eppure seguivo le istruzioni alla lettera delle varie guide, articoli FAQ ecc... che consultavo! Ore ed ore di tentativi sempre falliti! Alla fine o la chiavetta manco veniva riconosciuta (caso più frequente) oppure veniva riconosciuta ma non si avviava.... E poi ho scoperto il perché: se seguivate le istruzioni alla lettera ma avevate precedentemente formattato la chiavetta, allora vi mancava un passo fondamentale! La procedura che veniva sempre illustrata non prendeva mai in considerazione la possibilità che la chiavetta fosse stata appunto formattata. Seguendo invece il metodo che vi illustrerò a breve, sono riuscito a installare Feather Linux, una distribuzione derivata da Knoppix che sta in 128 MB, su una chiavetta USB 2.0 da 2 GB.

Se siete passati anche voi per questi insuccessi, allora il mio consiglio è uno solo: *attenzione* alle istruzioni che riportano online tanti siti, sono incomplete.

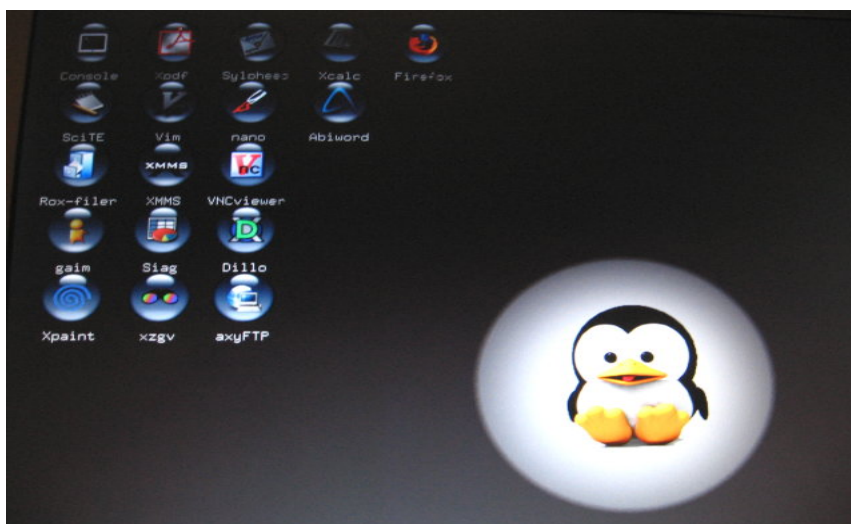
Faccio anche un piccolo, necessario, preambolo. Il supporto USB pienamente funzionante sotto Linux è roba di poco tempo fa... prima le cose erano diverse (peggiori) ed una periferica USB equivaleva a grattacapi perenni. Con l'introduzione del Kernel 2.4 prima e del 2.6 dopo, le cose sono notevolmente migliorate. Adesso Linux tratta le periferiche di massa USB a livello di Windows. Le istruzioni che riportano i vari Wiki, le FAQ e le documentazioni che trovate online sono ormai obsolete, non aggiornate, con l'obbligo di rispettare alcuni punti che potete oggi giorno anche saltare. Ecco i punti che non necessitano più di rispetto:

1. formattare la chiavetta con il filesystem FAT16;
2. utilizzare SysLinux 2.11 per rendere la chiavetta avviabile.

Ma il punto fondamentale che nessuna delle guide che ho trovato ha mai preso in considerazione è quello della formattazione

della chiavetta. Quanti di voi hanno provato a formattare la chiavetta con un filesystem Linux-friendly come ad esempio ext2, ext3, Reiser e compagnia bella? Io sì, anche perché questa è già una piccola protezione per rendere illeggibile la chiavetta in caso di smarrimento ad un utente poco esperto, che conosce solo Windows (inserendola nel PC e vedendo che non viene letta, sicuramente a pochi viene in mente che questa sia formattata con qualcosa di diverso rispetto ai filesystem gestibili da Windows: i più penseranno che sia difettosa). Se anche rimettete il filesystem VFAT, non reinstallate

Feather Linux è una distribuzione Linux, derivata da Knoppix, studiata per girare completamente in live da CD o penna USB. Occupa meno di 128 MB.



il MBR (Master Boot Sector), il settore che viene letto dal BIOS all'atto di avviare la chiavetta. E così, nonostante abbiate seguito tutti i punti, difficilmente riuscirete ad avviare una distribuzione Linux installata sulla vostra chiavetta USB...

Seguendo le istruzioni complete dell'installazione dell'MBR sono invece riuscito a far partire senza problemi Feather Linux: qui a destra potete infatti vedere uno screenshot di Feather Linux che gira sul mio PC, con il logo della distribuzione e le icone del software preinstallato.

Ecco i passaggi che ho seguito per ottenere il positivo risultato, precisando che ho lavorato su una installazione di Ubuntu 7.10 anche se ciò che leggerete vale per tutte le altre distribuzioni: basta installare il software citato con il gestore di pacchetti della propria distribuzione (URPMI per Mandriva, YAST per openSUSE, ecc...)

SysLinux è il software fondamentale per rendere avviabile la vostra chiavetta USB

La prima cosa da fare in assoluto è controllare se la propria scheda madre è capace di effettuare il boot da periferiche USB (tipo chiavette, HD od altro): controllate nelle impostazioni del BIOS, leggete il manuale della scheda madre o visitate il sito del produttore. Se così non fosse, terminate la lettura ed andate a guardarvi un altro articolo, che questo non fa più per voi.

Fatto questo doveroso controllo, bisogna procurarsi una chiavetta USB da destinare a Linux: vista la modesta richiesta di spazio delle distribuzioni appositamente sviluppate, ne basta anche una da 1 GB.

Poi procuratevi la distribuzione che volete installare. Ne esistono diverse per chiavetta USB: Damn Small Linux, Puppy Linux e Feather Linux sono le più famose. Ho scelto proprio quest'ultima, Feather Linux, scaricabile dal sito ufficiale (<http://featherlinux.berlios.de/>) in versione 0.7.4 come pacchetto ZIP per l'installazione su dispositivi USB perché è forse la più adatta, delle 3 citate, al nostro scopo per facilità d'installazione (viene infatti fornito un archivio contenente i file già pronti per la copia sul dispositivo USB) e per fornitura di software integrato. Feather Linux comprende infatti, nei suoi 128 MB (contro i 50 di Damn Small Linux ed i 90 di Puppy Linux) tutto l'occorrente per il lavoro di un utente medio: software di grafica, browser internet, gestore di posta elettronica, lettore di filmati e file musicali, programmi di chat, fogli di calcolo, editor di documenti, partizionatore per dischi, masterizzazione (tramite console), ecc... il tutto gestito con un minimalista ma gradevole desktop grafico (Fluxbox).

Mentre scarichiamo il file zippato della distribuzione, installiamo gli altri tool necessari all'operazione. Apriamo Synaptic (oppure apt-get se preferite la console) e selezioniamo il pacchetto mtools: questo pacchetto contiene degli strumenti per scrivere su partizioni DOS/Windows ed è necessario perché contiene il programma mcopy, usato da SysLinux per copiare sulla chiave USB il codice per avviare il kernel di Linux. Visto che lo abbiamo nominato, selezioniamo anche SysLinux con Synaptic.

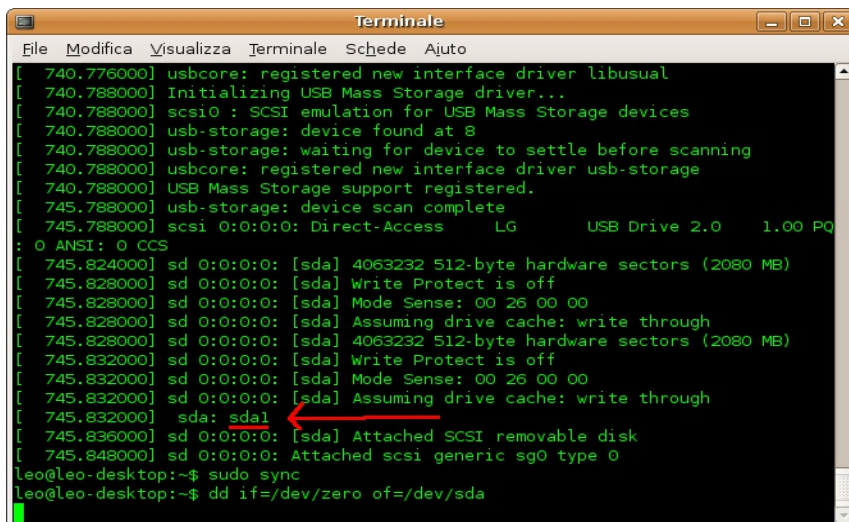
E qui veniamo al punto 2 delle indicazioni a mio modo di vedere obsolete. Tutte le guide, infatti, suggeriscono di usare SysLinux in versione 2.11 (che si può scaricare da un qualunque mirror di www.kernel.org) mentre tutte le distribuzioni più recenti hanno nei loro repository questo software in versione 3.xx (su Ubuntu 7.10, al momento della stesura dell'articolo, è presente la versione 3.36). Stando sempre alle guide, SysLinux con numero di versione superiore alla 2.11 non supporta correttamente le partizioni FAT16, il filesystem consigliato per formattare la vostra chiavetta USB (il punto 1 delle indicazioni obsolete). Io ho installato SysLinux in versione 3.36 e formattato la mia chiavetta

come FAT32.

L'ultimo pacchetto che ci serve è mbr, necessario per reinstallare l'MBR in caso avete precedentemente formattato la chiavetta. Fatto questo, possiamo iniziare a lavorare.

Prima di tutto va scompattato l'archivio della distribuzione in una cartella a vostra scelta. Poi bisogna collegare la chiavetta USB e subito dopo aprire un terminale e digitare il comando dmesg, che riporta i messaggi generati dal kernel, compresi i log della connessione delle periferiche. Il suo output dovrebbe essere simile a questo:

L'output che ci interessa è quello evidenziato in rosso: **sda1**.



```
Terminale
File Modifica Visualizza Terminale Schede Aiuto
[ 740.776000] usbcore: registered new interface driver libusual
[ 740.780000] Initializing USB Mass Storage driver...
[ 740.780000] scsi0 : SCSI emulation for USB Mass Storage devices
[ 740.780000] usb-storage: device found at 8
[ 740.780000] usb-storage: waiting for device to settle before scanning
[ 740.780000] usbcore: registered new interface driver usb-storage
[ 740.780000] USB Mass Storage support registered.
[ 745.780000] usb-storage: device scan complete
[ 745.780000] scsi 0:0:0:0: Direct-Access LG USB Drive 2.0 1.00 PQ
: 0 ANSI: 0 CCS
[ 745.824000] sd 0:0:0:0: [sda] 4063232 512-byte hardware sectors (2080 MB)
[ 745.828000] sd 0:0:0:0: [sda] Write Protect is off
[ 745.828000] sd 0:0:0:0: [sda] Mode Sense: 00 26 00 00
[ 745.828000] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 745.828000] sd 0:0:0:0: [sda] 4063232 512-byte hardware sectors (2080 MB)
[ 745.832000] sd 0:0:0:0: [sda] Write Protect is off
[ 745.832000] sd 0:0:0:0: [sda] Mode Sense: 00 26 00 00
[ 745.832000] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 745.832000] sda: sda1
[ 745.836000] sd 0:0:0:0: [sda] Attached SCSI removable disk
[ 745.848000] sd 0:0:0:0: Attached scsi generic sg0 type 0
leo@leo-desktop:~$ sudo sync
leo@leo-desktop:~$ dd if=/dev/zero of=/dev/sda
```

Questo ci dice che la nostra chiavetta è stata montata come disco sda e che la partizione usata è la 1. Ovviamente il vostro output potrebbe essere diverso (sdb1, ad esempio): prendetene comunque nota, perché i prossimi comandi faranno riferimento a questo dato. Se sbagliate a trascriverlo o ad inserirlo, potreste anche cancellare permanentemente i dati di uno dei vostri dischi fissi!!! Se la vostra penna USB era già usata, copiate tutti gli eventuali dati in essa contenuta sul PC, perché tra poco provvederemo a cancellare ogni suo bit...

Adesso va smontata la chiavetta: cliccate col tasto destro sulla sua icona e poi scegliete "Smonta volume" se lavorate da interfaccia grafica, altrimenti date da console il comando `sudo umount /dev/sdXy` dove sdXy sta per il valore trovato tramite il precedente comando `dmesg`. Attenzione: non rimuovete fisicamente la chiavetta, smontatela soltanto. In questa maniera il sistema non ne godrà più esclusivo possesso né la altererà per operazioni di lettura/scrittura, mentre voi potrete lavorarci con i tool che andremo tra poco ad richiamare da terminale.

Il passo successivo è la cancellazione dei dati della chiavetta. Se siete paranoici, potete effettuare una cancellazione totale di tutto quello che c'è sopra semplicemente dando

Prestate attenzione durante la formattazione della chiavetta USB: se sbagliate ad indicare la periferica, potreste cancellare tutti i dati del vostro disco fisso!

da terminale il comando `dd if=/dev/zero of=/dev/sdX`. L'operazione è lunga: in pratica, il comando preleva dalla periferica virtuale `/dev/zero` un flusso di zeri e li scrive sulla penna USB. Per cancellare la mia chiavetta, che è di 2 GB di capacità, ci sono voluti circa 15 minuti... Se non siete così paranoici (o se non avete così tempo a disposizione) potete limitarvi a cancellare il primo settore (nel primo settore di ogni dispositivo sono riportate le tabelle delle partizioni presenti) con il comando `dd if=/dev/zero of=/dev/sdX bs=512 count=1`

Fatto questo, la vostra chiavetta USB non contiene più traccia delle precedenti partizioni. Adesso possiamo procedere con la creazione di una nuova partizione FAT32. Sempre da terminale date il comando `fdisk /dev/sda` Si aprirà la console del programma di partizionamento di Linux. Digitate il tasto "p" seguito da Invio per avere la lista delle partizioni della vostra chiavetta: non ne dovrete visualizzare alcuna (infatti abbiamo appena cancellato la tabella delle partizioni). Adesso premete il tasto "n" ed Invio per creare una nuova partizione. Scegliete "Primaria", poi inserite come numero identificativo "1" e poi premete 2 volte invio per accettare i valori di inizio e fine della partizione (espressi in numero di blocchi) per utilizzare tutto lo spazio disponibile. Fatto questo, bisogna stabilire il tipo di partizione da usare. Come detto, ho scelto la FAT32 contro i suggerimenti delle guide che consigliavano la FAT16. Diamo il comando apposito col tasto "t" seguito da Invio e poi immettiamo come valore esadecimale del tipo di partizione il carattere "b". Il penultimo passo è impostare il flag di boot su on per la partizione appena creata, così che possa essere avviabile dal BIOS. Premiamo "a" seguito da invio e poi inseriamo il numero della partizione (se avete seguito questa guida, inserito "1"). Adesso non ci resta che scrivere le modifiche sulla chiavetta USB (fino ad ora non avete materialmente eseguito nessuna operazione): diamo il comando "w" seguito da Invio. Solo ora, `fdisk` scriverà le modifiche apportate al dispositivo.

Ora dobbiamo formattare la partizione, così da creare un filesystem utilizzabile. Per far ciò diamo da terminale il comando `mkfs.vfat -F 32 /dev/sdXy` dove indichiamo al sistema di formattare la partizione y del disco sdX (vi ricordate l'output del comando `dmesg`? Ecco, lo dovete usare qui. Attenti, però! Come ho detto potete sempre cancellare il vostro HD se sbagliate!) con filesystem FAT32 (l'opzione -F 32: se non la date, vi formatta la partizione come FAT16). Ora la chiave USB è vuota, partizionata e formattata. Bisogna riempirla, non trovate?

Allora montiamola e poi provvediamo a copiare la distribuzione. Per montare la chiavetta avete 2 metodi: o fate tutto da terminale, creando una cartella temporanea

(`mkdir tempdir`) e poi montando il dispositivo (`sudo mount /dev/sdXy tempdir`) oppure potete semplicemente sfilare la penna USB e ricollegarla: è un buon test per verificare che le operazioni di partizionamento e formattazione siano andate a buon fine, dato che se è vista e montata dal sistema allora non si sono verificati errori nella procedura. Aprite la cartella dove è stata montata e copiamoci dentro tutto il contenuto del file ZIP di Feather Linux.

Eccoci all'ultimo passaggio da effettuare prima della verifica pratica: copiare il codice per l'avvio della penna. Prima di tutto bisogna smontare la chiavetta (`sudo umount /dev/sdXy` da terminale oppure "Smonta volume" da GNOME) altrimenti non possiamo effettuare l'operazione. Poi da terminale digitiamo `syslinux -s /dev/sdXy` per rendere avviabile la nostra penna.

Se non avete mai formattato la vostra chiavetta allora potete proseguire con la lettura, altrimenti dovete compiere l'operazione di ripristino dell'MBR. Sempre da console, lanciate il comando `install-mbr -e 1 /dev/sdX`. Con questo comando installiamo l'MBR sul dispositivo sdX (quello della chiavetta USB come da output del comando `dmesg`) ed abilitiamo al boot la prima partizione, che è anche l'unica presente sulla chiavetta dato che, se vi ricordate, abbiamo creato una sola partizione sul nostro dispositivo.

Fatto questo, possiamo riavviare il sistema perché l'ultima parte della procedura va fatta modificando le impostazioni del BIOS della scheda madre. Riavviamo il sistema ma NON SCOLLEGHIAMO la chiavetta: bisogna che il BIOS la veda al momento del riavvio altrimenti non potete lanciare Feather Linux. Durante il riavvio della macchina, premendo CANCEL (nella maggior parte dei casi è questo tasto, altrimenti un altro segnalato comunque dalla scheda madre sullo schermo) si entra nel menu di modifica delle impostazioni del BIOS. Qui dobbiamo controllare l'ordine delle periferiche da controllare per l'avvio di un sistema operativo, impostando come primo dispositivo proprio la chiavetta USB. Le impostazioni variano da produttore a produttore, quindi ognuno dovrà controllare sul manuale della propria scheda madre. Salviamo le modifiche apportate ed usciamo dal BIOS. Il computer si riavvierà e lancerà il sistema operativo presente sulla chiavetta USB, Feather Linux.

Leonardo "leo72" Miliani

Se intendete salvare dei dati sulla vostra chiavetta, allora vi conviene creare 2 partizioni:

- 1) create una partizione primaria con numero 1 e dimensione di 256 MB (alla domanda sul blocco finale, digitate "+256M"), impostatela su tipo FAT32 e rendetela avviabile;
- 2) create una seconda partizione primaria con numero 2 dimensionandola con tutto lo spazio che resta sulla chiavetta, poi impostatene il tipo su Linux (valore "83");
- 3) scrivete le modifiche sulla chiavetta, poi formattate la partizione 1 come già detto;
- 4) per formattare la seconda partizione è necessario sfilare e riconnettere la chiavetta affinché il sistema veda anche la seconda partizione. Una volta riconnessa, smontatela e da terminale date il comando `mke2fs /dev/sdX2`. Formatterete così la seconda partizione del vostro dispositivo USB (sdX2) con ext2, un filesystem senza journaling. Questo perché per un dispositivo rimovibile un filesystem con journaling (ext3 o ReiserFS) potrebbe risultare insicuro: meglio affidarsi ad un filesystem di tipo tradizionale, che scrive le modifiche nel momento in cui lo richiedete voi.



La funzione "split"

Impariamo ad usare la funzione "split" per dividere una stringa

All'interno della struttura Gambas si possono incontrare diverse tipologie di trattamento dei dati che arrivano al programma. Detti dati (input), possono arrivare da diversi ingressi come la tastiera, un CD o DVD, una pendrive, una sorgente del mondo WEB, un floppy-disk. L'input proveniente dalla tastiera scatena eventi logici che vengono gestiti all'interno dei rispettivi controlli, mentre gli altri tipi di input, pur potendo produrre particolari eventi nei confronti del SO, sono ordinariamente richiesti dallo stesso programma.

Qualunque sia la provenienza, ogni input è affidato all'interpretazione, alla manipolazione ed a produrre un output da trasferire ad un nuovo supporto fisico, come una finestra del monitor, un destinatario web, un CD o DVD, una pendrive, un floppy-disk, attraverso una serie di istruzioni incapsulate dentro le classi di Gambas.

È durante la fase di interpretazione e manipolazione che intervengono tutta una serie di istruzioni, da quelle elementari a quelle più complesse. Queste ultime racchiudono al loro interno più istruzioni elementari e pertanto costituiscono vere e proprie funzioni. Fa parte di questo gruppo la funzione **Split**:

```
StringArray = Split ( String AS String  
[ , Separators AS String , Escape AS  
String , IgnoreVoid AS Boolean ] )
```

dove

StringArray è l'array dove verrà caricata ogni singola sottostringa

String è la stringa contenente il dato iniziale

Separators è il carattere speciale che separa le varie sottostringhe

Escape è un carattere speciale che annulla la funzione split quando un carattere separatore è compreso fra due escape. Se i caratteri escape sono due, allora il primo identifica il carattere escape di sinistra, mentre il secondo identifica il carattere escape di destra

IgnoreVoid il suo uso permette di abbandonare eventuali sottostringhe vuote

L'utilità della funzione Split si avverte quando occorre trattare un insieme di dati disomogenei da riconoscere e riorganizzare a seconda di uno o più elementi distintivi. Riporto qui di seguito un esempio esplicativo che permette di capire un articolato uso della

funzione Split.

Supponiamo di avere come input un file.txt avente record di lunghezza e contenuto diversi; dobbiamo rilevare solamente alcuni dei record presenti. Il file.txt è così formato:

La funzione **Split** divide il contenuto di una stringa

```
Savonarola 'Partite' ( 'Codice1' sempreverde, 'lampada'  
sempreverde regolare nullo, 'rivisitare' sempreverde,  
primario ( 'Codice1', 'lampada', 'rivisitare' ) );  
Savonarola 'Presto' ( 'Codice2' sempreverde, 'Data1'  
sempreverde, 'orario' sempreverde, 'Stato' crt(1) regolare  
nullo, primario ( 'Codice2', 'Data1', 'orario' ) );  
Savonarola 'riepilogando' ( 'Data2' sempreverde,  
'Codice3' sempreverde, 'Monitor' crt(1) regolare nullo,  
'imperioso' fluido regolare nullo, 'beneficiario' crt(1)  
regolare nullo, primario ( 'Data2' ) );  
Savonarola 'ripresa motivi' ( 'Data4' sempreverde,  
'Stallo' crt(1), 'Entrata' sempreverde regolare nullo,  
'TotUscg' sempreverde regolare nullo, 'riportato'  
sempreverde regolare nullo, primario ( 'Data4' ) );  
Savonarola 'Apertura' ( 'Data5' sempreverde primario );  
Savonarola 'Componenti' ( 'progressivo' sempreverde,  
'NomeCompFam' variabile(20) regolare nullo, primario ( 'progressivo' ) );  
inserimento "Componenti" VALUES(1,'Famiglia');  
inserimento "Componenti" VALUES(2,'Piero');  
inserimento "Componenti" VALUES(3,'Giuseppe');  
inserimento "Componenti" VALUES(4,'Giacomo');  
Savonarola 'Causali' ( 'descrizione' variabile(100),  
'Data5' sempreverde );  
inserimento "Causali" VALUES('Vitto',20090601);  
inserimento "Causali" VALUES('bar',20090525);  
inserimento "Causali" VALUES('fotografie per tessera  
mia',20125499);  
Savonarola 'Pianoforte' ( 'Numero' primario, 'Voce'  
variabile(50) regolare nullo, 'Contrario' crt(1) regolare  
nullo, 'Datauso' sempreverde regolare nullo );  
inserimento "Pianoforte"  
VALUES(10100000,'CASA','N',nullo);  
inserimento "Pianoforte"  
VALUES(10100100,'Emollienti','N',nullo);  
inserimento "Pianoforte" VALUES(10100102,'Pensione  
gallo','N',nullo);  
inserimento "Pianoforte" VALUES(10100600,'Riscossa  
avantitutta','S',nullo);  
inserimento "Pianoforte"  
VALUES(10100700,'Vittoria','N',nullo);
```

Di tutte le righe sopraelencate, dobbiamo estrarre solamente le righe contenenti la parola "Pianoforte". Eseguiamo prima il caricamento e subito dopo operiamo la

distribuzione dei record che ci interessano dentro un array. Per fare ciò eseguiamo le seguenti istruzioni:

```
'-- definizione dell'area buffer di caricamento del file.txt
DIM hFile AS File
'-- definizione della stringa di memorizzazione di tutti i record
DIM $Rec AS String...
'-- array dove andranno a memorizzarsi solamente i record contenenti la parola "Pianoforte"
DIM $RgFl AS NEW String[]
'-- esegue il caricamento del file senza dovere aprirlo alla lettura..
$Rec = File.Load($File)...
'-- esegue la valorizzazione dell'array con le sottostringhe, una in ciascun elemento dell'array.
$RgFl = Split($Rec, ";")
```

Ora le sottostringhe sono direttamente identificabili, però il nostro lavoro non è ancora finito; occorre ancora estrarre le righe contenenti la parola "Pianoforte" e redistribuirne il contenuto in un nuovo array

per ben identificare ulteriori sottostringhe da cui è composta ciascuna riga nell'array \$RgFl, di partenza.

Per cui, si avrà:

```
DIM bEsito AS Boolean
DIM i AS Integer
DIM iIndMax AS Integer
DIM iLun AS Integer
DIM iPoStri AS Integer
'-- array dove andranno a memorizzarsi solamente le righe
'-- di $RgFl contenenti la parola "Pianoforte"
DIM $RgRecDB AS NEW String[]...

iIndMax = $RgFl.Count - 1
FOR i = 0 TO iIndMax
  '-- Cerca dentro la stringa $RgFl(i) la sottostringa "Pianoforte"
  bEsito = $RgFl[i] LIKE "*" & "Pianoforte" & "*"...
  IF bEsito = TRUE THEN
    iPoStri = InStr($RgFl[i], "VALUES(")
    IF iPoStri <> 0 THEN
      iLun = Len($RgFl[i])
      $RgRecDB = Split(Right($RgFl[i], iLun - (iPoStri + 5)), ",")
```

La funzione agisce solo sulla parte più significativa di ciascuna riga di \$RgFl, che si trova nella sua estrema destra e cioè:

```
(10100000, 'CASA', 'N', nullo)
(10100100, 'Emollienti', 'N', nullo)
(10100102, 'Pensione gallo', 'N', nullo)
```

```
(10100600, 'Riscossa
avantitutta', 'S', nullo)
```

```
(10100700, 'Vittoria', 'N', nullo)
```

Dopo l'esecuzione della funzione Split il nostro array finale "\$RgRecDB" è così composto:

```
-- 1° elemento      'N'
(10100000          nullo)
'CASA'             -- 4° elemento
'N'                (10100600
nullo)             'Riscossa avantitutta'
-- 2° elemento      'S'
(10100100          nullo)
'Emollienti'       -- 5° elemento
'N'                (10100700
nullo)             'Vittoria'
-- 3° elemento      'N'
(10100102          nullo)
'Pensione gallo'
```

Esso però contiene ancora caratteri non significativi che la funzione split non ha potuto togliere. Bisogna perciò affidare tale

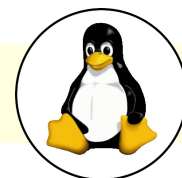
compito ad una funzione diversa, che illustrerò ampiamente nel prossimo numero.

Pietro Antonino "Picavbg" Catania

Spesso non importa reinventare la ruota. Basta usare gli strumenti messi a disposizione dal linguaggio stesso: Split è una di quelle funzioni che passa in second'ordine ma che spesso si rivelano di una utilità unica.

L'AUTORE: Pietro Antonino Catania
Sin da bambino ho avuto attrazione per le macchine calcolatrici e poi, più grandicello, per l'informatica. Diplomato in Ragioneria ho potuto dopo alcuni anni affacciarmi nel mondo del lavoro come bancario e lì, alla prima occasione, ho partecipato ad un test attitudinale di informatica che ho superato e da quel momento, ottenuta la qualifica di programmatore, ho potuto rattopparmi per quasi tutta la vita lavorativa con gli elaboratori elettronici e coi computer.

OpenSource



MeteoMachine Linux

Piovono righe e righe di codice!

Quante volte siete stati davanti alla TV in attesa che la pubblicità finisse per potervi accertare delle condizioni meteo della settimana? Quante volte avete organizzato pic-nic fidandovi dei satelliti dell'Aeronautica ma poi vi siete bagnati imprecaando verso Apollo? E poi, dite la verità, quante volte non sapevate se pioveva o ci sarebbe stato il sole solo perché il disegno della nuvola in TV è grande quanto tutto il sud Italia? Finalmente d'ora in poi potrete organizzare tranquillamente le vostre scampagnate in famiglia perché avrete un pinguino meteorologo. Ebbene sì, avete capito.... fate il gesto dell'ombrello alla TV e continuate a leggere.

COS'E'

Meteomachine Linux nasce dall'incontro delle passioni di tre persone che hanno unito idee, conoscenze e volontà per creare un software quasi unico nel suo genere. Si parla di un ampio progetto, scritto in Gambas solo per ambienti GNU/Linux, che comprende numerosi tool che permettono di monitorare, archiviare e gestire i dati meteorologici della propria stazione. Il suo uso è molto semplice ma tale semplicità non deve essere confusa con la povertà di funzioni. Attualmente il programma, rilasciato con licenza GNU GPL3, è alla versione 1.6.8 e presenta le seguenti caratteristiche:

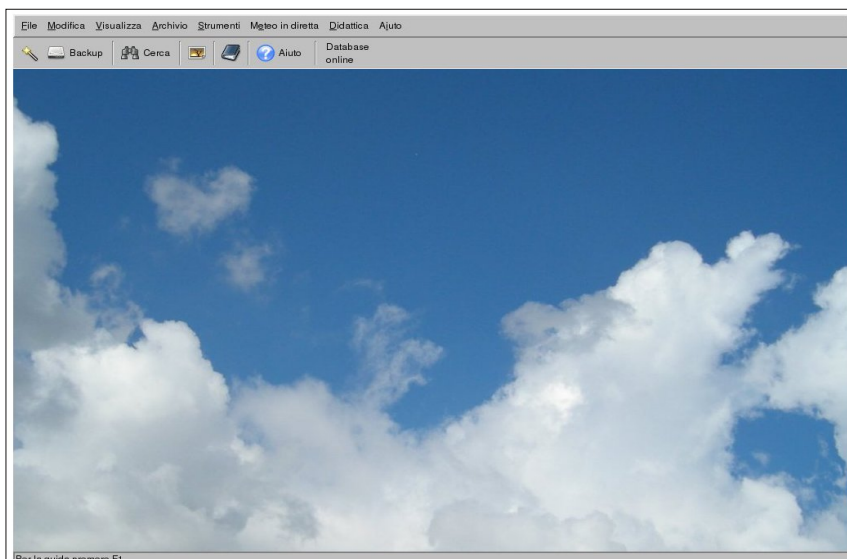
- archiviazione di dati (temperatura, pressione, precipitazioni, vento, umidità, stato del cielo) in locale mediante database SQLite, calcoli (medie e totali) e confronti degli stessi;
- visualizzazione precisa e dettagliata di grafici esportabili dei dati archiviati;
- importazione dei dati da fogli di calcolo;
- conversione di unità di misura utili;
- previsione approssimativa dell'andamento del tempo per le prossime 12/24 ore;
- visualizzazione di immagini satellitari statiche e dinamiche;
- visualizzazione di carte meteorologiche, divise per tipi e Paesi, in modo statico o animato;
- gestione di un database remoto da condividere.

Queste sono solo alcune delle potenzialità di questo programma che potrebbe piacere ed essere utile anche a chi vuole solo imparare

qualcosa in più sull'argomento o chi già lo conosce ma vuole approfondirlo. Tra i vari menu, infatti, è possibile incontrare delle voci che permettono la visualizzazione di informazioni a titolo educativo in modo da integrare un ambiente didattico all'interno della struttura gestionale. Questo permette di raggiungere un alto livello di completezza che rende MeteoMachine Linux un software adatto anche per scuole ed hobbysti.

L'INTERFACCIA

Fin dal primo avvio l'interfaccia risulta semplice e immediatamente comprensibile. Il menu è stato studiato per raccogliere le voci con criterio logico. La toolbar aiuterà l'utente a svolgere le più importanti operazioni con pochi e semplici click.



I form sono semplici ed intuitivi e permettono di eseguire personalizzazioni in ogni aspetto utile del programma. Si va dall'impostazione del proprio Paese alle unità di misura con le quali si vogliono salvare i propri valori.

Sarà possibile, inoltre, decidere se visualizzare all'avvio una semplice immagine satellitare nel form principale.

I form utili all'archiviazione dei dati permettono non solo di inserirne di nuovi ma anche di modificare i vecchi o semplicemente di visualizzarli per poter avere sempre a disposizione i dati. Gli strumenti per la conversione vanno dalla temperatura con i gradi Centigradi/Fahrenheit alle precipitazioni in millimetri/inch, passando per la pressione e la velocità del vento.

Programmatori:

Ripamonti "stef"
Stefano

Fea "fsurfing" Sergio

Apruzzese "Ceskho"
Francesco.

Per visualizzare le altre funzionalità visita il sito del progetto

www.altabrienza.org/meteomachine

Le immagini satellitari possono godere, rispetto all'ultima versione, di un'interfaccia pienamente rivisitata e di un codice completamente riscritto. La nuova grafica permette una gestione delle voci molto più comoda e una loro catalogazione semplice e precisa.

Tra le immagini possiamo distinguere quelle:

* *semplici*: immagini statiche scattate dalle più importanti agenzie del mondo tra cui figura anche la NASA;

* *animate*: immagini in formato GIF che raccolgono la situazione di un Paese in diverse fasi della giornata e la mostrano tramite semplice animazione;

* *carte meteorologiche statiche*: visualizzano una semplice carta in base al momento della giornata;

* *carte meteorologiche animate*: visualizzano una raccolta di immagini, per tutta la giornata, che può essere stoppata, fermata, riavvolta e visualizzata quante volte si vuole.

È stata introdotta, inoltre, la possibilità di visualizzare la situazione del proprio Paese mediante l'utilizzo di Google Maps integrato nel proprio browser di default.

La sezione didattica si presenta come una raccolta di form riguardanti lo studio e la constatazione pratica di alcuni argomenti importanti. È possibile, ad esempio, studiare gli effetti del vento in base alla sua velocità grazie alla scala di Beaufort oppure visionare una probabile previsione del tempo grazie ai dati di cui si dispone

Come si è visto il programma non pecca certo di mancanza di possibilità e si è ancora a lavoro per portare novità quanto prima possibile. Nel futuro la speranza è di poter interfacciare direttamente le stazioni meteorologiche tramite la porta USB così da permettere la lettura ed archiviazione automatica dei dati. Questo sicuramente renderà MeteoMachine Linux il miglior software del genere che si possa trovare sul mercato dell'opensource.

Non mi resta che augurarvi una bella giornata, di sole o di pioggia che sia... ora lo saprete prima!

Francesco "Ceskho" Apuzzese

The screenshot shows a weather application interface. At the top, there's a navigation bar with a calendar for December 2009. Below it, a table shows daily weather data for the month. A section titled 'Le situazioni di "non pervenuto" vengono indicate con un trattino "-"' includes a checkbox for 'Avanza di un giorno al salvataggio' and fields for 'Min °C' and 'Max °C' with a 'Salva' button. Below that, there's a 'Stato del cielo' dropdown menu set to 'Sereno'. A section for 'Media mensile minima: 9.33 °C' and 'Media mensile massima: 14.33 °C' is also present. A 'Grafico mese corrente' section shows a line graph for 'Temperature' and 'Stato del cielo'. On the right, a table lists daily weather data:

Giorno	Minima °C	Massima °C	Stato cielo
1	-	-	
2	-	-	
3	-	-	
4	-	-	
5	-	-	
6	-	-	
7	-	-	
8	-	-	
9	-	-	
10	-	-	
11	-	-	
12	-	-	
13	-	-	
14	-	-	
15	-	-	
16	-	-	
17	-	-	
18	-	-	
19	-	-	
20	1	7	Nuvoloso
21	-1	10	Nuvoloso
22	12	15	Nuvoloso
23	14	21	Nuvoloso
24	15	16	Sereno
25	15	17	Sereno
26	-	-	

At the bottom, there are buttons for 'Esporta valori', 'Stampa valori', 'Chiudi', and 'Prossimo form'.

The screenshot shows a 'Meteo dal web' interface. On the left, there's a menu for 'Operazioni' with options like 'Selezionare l'immagine da visualizzare', 'IMMAGINI SATELLITARI', 'Statica polare Italia', 'Statica Europa', 'Statica MODIS nord Italia', 'Animata Europa', 'NASA', 'Europa', 'Bacino mediterraneo', 'Nord Italia', 'Nord-Ovest Italia', 'Sud Italia', 'Europa', 'Mar Adriatico', 'Alpi', 'Etna', 'Sud Europa - Nord Africa', 'Francia', 'Corsica - Sardegna', 'Grecia', 'CARTE METEOROLOGICHE', 'Temperature', 'Precipitazioni', 'Nuvolosità', 'Venti', 'Mare', 'Pressione', 'Analisi sinottica', 'MODELLI PREVISIONALI', and 'Modello GFS'. On the right, there's a satellite image of Europe. At the bottom, it says 'Immagine ridimensionata alle normali dimensioni'.

OpenSource



Chi cerca trova

Prova della minidistro Tiny Core Linux

Ore 10.00 - Marco mi telefona, è un po' abbattuto, il suo vecchio PC lo ha abbandonato, e il consiglio che gli avevo dato (quello di passare a Linux) non è andato a buon segno.

Non ha bisogno di grandi utilità ma almeno vuole potersi collegarsi ad internet.

Cerca conforto e ascoltandolo penso a quanti come lui si sono trovati nella stessa situazione.

"Ecco, finalmente ho deciso: passo a Linux."

Quante volte abbiamo esclamato questa frase e contenti e felici ci siamo messi alla ricerca di cosa fare? Finalmente abbandoniamo Windows. Scarichiamo la nostra bella ISO e procediamo all'installazione. Ma quanti di voi una volta che hanno cercato di installare senza risultato una distro sono tornati indietro e hanno reinstallato Windows?

Linux?? Un sistema operativo per smanettoni o per *hacker*. Neanche riesco ad installarlo figurati cosa succede dopo in futuro quando avrò un problema. No no, scusatemi ho sbagliato: torno a Windows.

Questo è la situazione tipica in cui un *newbie* (persona inesperta che mostra l'intenzione di migliorarsi) alle prime difficoltà getta la spugna. Tanti sono i fattori determinanti che possono impedire la mancata riuscita di una installazione Linux anche se ai più esperti possa sembrare scontato, a volte per un neofita non è così.

Ma per risolvere questo inconveniente ecco che la comunità che utilizza e che ha creato l'opensource si mobilita creando sempre più semplificazioni per avvicinare più o meno tutti a questa stupenda realtà.

Lo scenario è semplice: Marco (*newbie*) del mondo Linux, con personal computer datato (Pentium con 300 MB di ram, scheda video da 2 MB) che sfortunatamente non si accende più. Si iscrive ad un forum e finalmente si decide a fare il grande passo. Passare a Linux.

Ha appena scaricato una distro di cui ha sentito molto parlare... ma, ecco nascere le prime difficoltà. Schermo nero, lettore CD non funzionante, la distro non parte, il PC si blocca. Marco molto deluso.

Ma caro Marco, è vero che sei un *newbie* ma sai che ci sono tante alternative e che Linux

non è soltanto il CD che tu hai masterizzato con tanto amore? Non tutte le distro vanno bene per tutti i computer. Anche se Marco ha sentito parlare bene di Ubuntu (per esempio), perché è particolarmente carino graficamente, Marco non sa che deve anche fare i conti con il suo personal computer. Ed ecco che arriva l'aiuto da parte della comunità

Esistono tante distro leggere che fanno sì che anche i computer datati resuscitino. Sì, perché la forza di Linux è anche quella di abbattere quella convinzione che dopo due anni il PC è diventato obsoleto e inutilizzabile.

Tra le tante in circolazione, vi segnalo l'uscita della nuova Tiny Core Linux 2.9 (4 marzo 2010). Distro minimale che si è presa lo scettro di distribuzione dalle dimensioni ultraridotte. Il programmatore nonché padre Robert Shingledecker, dopo aver abbandonato il progetto Damn Small Linux (altra distro minimale Linux), ha voluto

Per info:
<http://tinycorelinux.com/>

Lista programmi:
<http://distro.ibiblio.org/pub/linux/distributions/tinycorelinux/tce.html>

changelog:
<http://tinycorelinux.com/forum/index.php?topic=5193.0>



sfidarsi con questa mini ma accattivante distribuzione: 10 MB, basata sul kernel Linux 2.6 e con interfaccia grafica minimale. Ambiente shell Busybox (che racchiude vari strumenti Unix) e ambiente desktop basato su Tinyx.

La velocità di boot è impressionante, come dichiara lo stesso Shingledecker: questo perché viene usato il sistema di copiatura del core nella RAM, che offre dei tempi di accesso alle risorse molto veloci.

- Possibilità di installazione programmi da repository TCE

- Miglioramento del caricamento dei file in locale

- Librerie aggiornate,

- Udev 151 aggiornato

- libstdc++ 6.0.9 aggiornato

- glibc 2.9 supportano ora i486

- miglioramenti in wbar, appbrowser, appsaudit, tc-config.

Queste solo alcune caratteristiche di Tiny Core.

Abbastanza facile l'installazione da CD-live e il riconoscimento dell'hardware non presenta grosse difficoltà. Non ci sono programmi preinstallati, appunto per mantenere un sistema snello e veloce all'avvio. Tuttavia è possibile installare il tutto successivamente. Di default presenta una schermata davvero minimale con una barra stile Cairo in cui si visualizzeranno: il terminale, il pannello di controllo e il menù applicazioni.

"Menù applicazioni" consente una rapida installazione di tantissimi programmi; cliccando su "AppBrowser" si apre un elenco dei programmi più usati sulla nostra sinistra che ci porta ad una installazione semplice e abbastanza veloce del programma scelto. In questo caso potete vedere sulla barra il browser Firefox installato. Infatti, una volta scaricati i software voluti, Tiny Core si preoccupa di mettere direttamente l'icona sulla barra in modo da avere una visualizzazione immediata del programma scelto.

Cliccando sul Pannello possiamo visualizzare un menù da cui si possono fare le operazioni più svariate, dal cambiare l'ora alla possibilità di montare gli hard disk o al cambio dei colori per quanto riguarda il desktop. Io ho messo il verde e poi ho cambiato con un blu con Pidgin di default.

Da notare che non si parla di una distro completa sotto tutti i punti di vista ma la sua caratteristica principale è appunto la velocità di esecuzione e la possibilità di avere un desktop minimo con accesso a internet, adatta a chi vuole dare nuova vita a PC, notebook e netbook obsoleti.

Una pecca potrebbe essere il fatto di non avere una comunità in italiano con un forum dedicato, ma non si trovano problemi con la comunità internazionale.

L'alternativa a Windows esiste non abbattiamoci alle prime difficoltà. Con Windows un PC è da buttare dopo qualche anno, con Linux bisogna solo cercare la distro adatta.

Perché, alla fine dei conti, **chi cerca trova.**

E Marco è felice.

Gaetano "tangoku7" Lo Nigro



L'AUTORE: Gaetano Lo Nigro

Sono un segretario d'albergo con la passione per l'informatica. Lavoro all'estero come receptionist/night auditor. Mi piace scrivere e fare piccoli lavoretti grafici utilizzando solo software opensource. Mi piace stare a contatto con le persone e amo tenermi informato su tutto quello che succede nel mondo.

Programmazione



Su che cosa e come realizzare un programma Il computer come intelligente strumento di sussidio alle attività umane

Oggi, il nostro quotidiano è quasi totalmente immerso nella tecnologia (dico quasi, per lasciare ancora un briciolo di indipendenza umana dalla continua sottomissione al diffusissimo uso di mezzi e macchine che ci aiutano a svolgere le nostre azioni, anche le più comuni). Eppure raggiungere l'attuale livello di progredita combinazione fra macchinario e uomo-utente non è stato un percorso facile. Fermiamoci un attimo e cerchiamo di ricordare quello che abbiamo fatto oggi, da quando ci siamo svegliati dal sonno della notte. Probabilmente, ancora prima di uscire da casa abbiamo dato una sbirciatina ai messaggi ricevuti sul telefonino, poi abbiamo preso l'automobile, nostra fedele compagna di trasporto, e ci siamo recati dal giornalaio per comprare il nostro giornale quotidiano preferito, abbiamo fatto colazione al bar e subito ci siamo recati al nostro posto di lavoro; nel frattempo abbiamo finito il dentifricio e, a sera, siamo entrati in farmacia per acquistarne un altro. Fatta la spesa, rincasato, in attesa, di poter chattare, dopo la cena, con gli amici della comunità che frequentiamo in Internet. Quelle elencate sono tutte attività che svolgiamo normalmente ogni giorno, ma se riflettiamo un po' su quanta tecnologia ci ha investito in tutto quello che abbiamo fatto, possiamo renderci conto di quanto il rapporto fra tecnologia ed umanità ormai sia nettamente sbilanciato, perchè il piatto più pesante è quello della tecnologia.

Infatti, la prima cosa che abbiamo fatto nella giornata è stata quella di accendere il telefonino per interrogare eventuali messaggi arrivati; ebbene già con questo passo abbiamo utilizzato il software residente nel telefonino tramite la pressione dei tasti necessari per arrivare all'esplosione dei messaggi pervenuti; quel software ha acceduto alle celle di memoria dove risiedono i messaggi e ce li ha mostrati nel piccolissimo monitor del telefonino.

Poi abbiamo acceso l'automobile, e già all'accensione è stato attivato il programma di controllo di sicurezza dell'auto che cerca l'accoppiamento col codice contenuto nella chiave di accensione. Se poi l'automobile è fornita di navigatore satellitare, il nostro percorso è stato via via dispiegato sul monitor del navigatore, tramite il programma di lettura satellitare della percorrenza, con la nostra posizione sempre sotto controllo. Dal giornalaio abbiamo

acquistato il giornale preferito. Il giornalaio non ha fatto lo scontrino, ma ha smanettato sul suo PC per chiedere al magazziniere virtuale (altro programma) di scaricare la merce che ci ha venduto.

Al bar abbiamo consumato il primo degli 8 caffè della giornata, accompagnandolo con una brioscina vuota: sì, vuota perchè così abbiamo potuto illuderci di non ingrassare. Dopo la consumazione, alla cassa, abbiamo conquistato lo scontrino fiscale, senza pensare che nel registratore di cassa risiede un programma che ha memorizzato la nostra consumazione per i fini fiscali e forse anche per il lavoro del consulente contabile del bar; e fin quando non vengono raccolti i dati sulle calorie immesse nel nostro corpo, possiamo considerarci ancora fortunati per il briciolo di libertà di cui possiamo ancor oggi disporre.

Dopo la nostra pesante giornata nella postazione di lavoro, magari sempre davanti ad un PC, alla presa continua con programmi di tutti i tipi, finalmente abbia preso la via del ritorno verso la nostra dolce casa e, strada facendo, ci siamo fermati in farmacia per comprare il nostro dentifricio, ma siccome nel frattempo siamo stati presi da un certo mal di testa, abbiamo pensato di comprare anche un antinfiammatorio. Così il farmacista, al quale abbiamo dato la nostra tessera sanitaria per la detrazione fiscale, ci ha fornito uno scontrino che riporta oltre al totale da pagare la citazione della parte di spesa che potrà essere portata in detrazione nella prossima dichiarazione di reddito.

Anche qui un programma ha svolto in maniera impeccabile il suo lavoro riconoscendo il farmaco dal dentifricio, che pur essendo il risultato di una formula chimica medicamentosa, è escluso dalla detrazione fiscale annua. Finalmente a casa; è lì che dopo la cena e l'antinfiammatorio, non pensiamo più al mal di testa perchè vogliamo riposarci con la chat nel nostro PC, da intrattenere in Internet, con gli amici della nostra comunità preferita. Insomma... programmi ed ancora programmi.

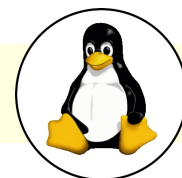
Ma cosa sono questi programmi? Come si fa a pensare ad un programma? E come si realizza un programma? Forse è meglio fermarci qui per ora, per riprendere il nostro percorso nel mondo della programmazione la prossima volta.

(Fine 1a parte)

Pietro Antonino "Picavbg" Catania

L'AUTORE: Pietro Antonino Catania
Sin da bambino ho avuto attrazione per le macchine calcolatrici e poi, più grandicello, per l'informatica. Diplomato in Ragioneria ho potuto dopo alcuni anni affacciarmi nel mondo del lavoro come bancario e lì, alla prima occasione, ho partecipato ad un test attitudinale di informatica che ho superato e da quel momento, ottenuta la qualifica di programmatore, ho potuto rappottarmi per quasi tutta la vita lavorativa con gli elaboratori elettronici e coi computer.

OpenSource



La cattedrale ed il bazar

Due stili di sviluppo del software a confronto: il modello "cattedrale" e quello "bazar"

Quella che segue è la mia analisi di un progetto open source di successo, fetchmail, deliberatamente utilizzato come test specifico per la verifica di alcune sorprendenti teorie sullo sviluppo del software suggerite dalla storia di Linux. Le mie argomentazioni su tali teorie mettono a confronto due diversi stili di sviluppo, il modello "cattedrale" in voga in gran parte del mondo commerciale, opposto al modello "bazaar" del mondo Linux. Da qui passo poi a dimostrare come tali modelli derivino da premesse divergenti sulla natura dell'attività di debugging del software. Arrivo quindi a stabilire la validità dell'esperienza di Linux riguardo l'affermazione "Con molti occhi puntati addosso, ogni bug diventa una bazzecola", per suggerire analogie produttive con altri sistemi di agenti indipendenti in grado di auto-correggersi, concludendo infine con una serie di riflessioni sulle implicazioni di queste analisi per il futuro del software. Eric. S. Raymond (1997)

1. La cattedrale e il bazar

Linux è sovversivo. Chi avrebbe potuto pensare appena cinque anni fa che un sistema operativo di livello mondiale sarebbe emerso come per magia dal lavoro part-time di diverse migliaia di hacker e sviluppatori sparsi sull'intero pianeta, collegati tra loro solo grazie ai tenui cavi di Internet?

Certamente non il sottoscritto. Quando Linux fece la sua comparsa nel mio raggio d'azione all'inizio del 1993, mi ero occupato dello sviluppo di Unix e di software open source per dieci anni. Ero stato uno dei primi collaboratori al progetto GNU a metà anni '80. Avevo distribuito su Internet un buon numero di software open source, realizzando da solo o in collaborazione con altri parecchi programmi (nethack, Emacs VC e GUD, xlife, etc.) ancor'oggi ampiamente utilizzati. Pensavo di sapere come bisognasse fare.

Linux stravolse gran parte di quel che credevo di sapere. Per anni avevo predicato il vangelo Unix degli strumenti agili, dei prototipi immediati e della programmazione evolutiva. Ma ero anche convinto che esistesse un punto critico di complessità al di sopra del quale si rendesse necessario un approccio centralizzato e a priori. Credevo che il software più importante (sistemi operativi e strumenti davvero ingombranti

come Emacs) andasse realizzato come le cattedrali, attentamente lavorato a mano da singoli geni o piccole bande di maghi che lavoravano in splendido isolamento, senza che alcuna versione beta vedesse la luce prima del momento giusto.

Rimasi non poco sorpreso dallo stile di sviluppo proprio di Linus Torvalds - diffondere le release presto e spesso, delegare ad altri tutto il possibile, essere aperti fino alla promiscuità. Nessuna cattedrale da costruire in silenzio e reverenza. Piuttosto, la comunità Linux assomigliava a un grande e confusionario bazaar, pullulante di progetti e approcci tra loro diversi (efficacemente simbolizzati dai siti contenenti l'archivio di Linux dove apparivano materiali prodotti da chiunque). Un bazaar dal quale soltanto una serie di miracoli avrebbe potuto far emergere un sistema stabile e coerente.

Il fatto che questo stile bazaar sembrasse funzionare, e anche piuttosto bene, mi colpì come uno shock. Mentre imparavo a prenderne le misure, lavoravo sodo non soltanto sui singoli progetti, ma anche cercando di comprendere come mai il mondo Linux non soltanto non cadesse preda della confusione più totale, ma al contrario andasse rafforzandosi sempre più a una velocità a malapena immaginabile per quanti costruivano cattedrali.

Fu verso la metà del 1996 che mi parve d'essere sul punto di capirne il perché. Il destino mi offrì l'occasione propizia per mettere alla prova la mia teoria, sotto forma di un progetto open source del quale decisi di occuparmi usando coscientemente lo stile bazaar. Ci provai, e il successo ottenuto fu piuttosto significativo.

Nella parte restante di questo saggio, racconto la storia di quel progetto, usandola per proporre alcuni aforismi sull'efficacia dello sviluppo open source. Non che li abbia imparati tutti dal mondo Linux, ma vedremo come le modalità offerte da quest'ultimo siano del tutto peculiari. Se non ho interpretato male, questi aforismi ci aiuteranno a comprendere con esattezza cos'è che rende la comunità Linux una sorgente così copiosa di buon software - e aiuteranno tutti noi a divenire più produttivi.

(Continua)

L'autore, *Eric Steven Raymond*, è un informatico statunitense nato nel 1957. È lo sviluppatore del software **fetchmail**, è l'attuale manutentore del **Jargon File** (meglio noto come "Il nuovo dizionario degli hacker") e l'autore del **Glider**, il simbolo identificativo degli hacker. È una figura prominente del panorama hacker internazionale e nel 1997 ha pubblicato un'analisi su due stili differenti di sviluppo del software libero. Il saggio, intitolato "**La cattedrale e il bazaar**", è generalmente considerato il manifesto del movimento open source.

Il resto del Pinguino pubblicherà il saggio a dispense per motivi di lunghezza dello stesso.

Il saggio è liberamente consultabile su WikiSource all'indirizzo: http://it.wikisource.org/wiki/La_cattedrale_e_il_bazaar

**"Il resto del Pinguino" è la rivista digitale ufficiale di Gambas-it.org,
la comunità italiana dei programmatori Gambas.
*Pubblicazione aperiodica***

Tutti i testi e le immagini contenuti in questa rivista sono rilasciati sotto la licenza **Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 2.5** (vedi sotto). Ciò significa che siete liberi di adattare, copiare, distribuire ed inviare gli articoli solo alle seguenti condizioni: la paternità dell'opera deve essere attribuita in qualsiasi modo (con almeno un nome, un'e-mail o un URL) all'autore originale e al nome di questa rivista digitale ("Il resto del Pinguino") e all'URL www.gambas-it.org (ma non si possono attribuire il o gli articoli in alcun modo che lasci intendere che gli autori e la rivista abbiano esplicitamente autorizzato voi o l'uso che fate dell'opera). Se alterate, trasformate, o aggiungete informazioni all'opera, dovete distribuire il lavoro risultante con la stessa licenza o una simile o compatibile.



Attribuzione-Non commerciale-Condividi allo stesso modo 2.5 Italia

Tu sei libero:



di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera



di modificare quest'opera

Alle seguenti condizioni:



Attribuzione — Devi attribuire la paternità dell'opera nei modi indicati dall'autore o da chi ti ha dato l'opera in licenza e in modo tale da non suggerire che essi avallino te o il modo in cui tu usi l'opera.



Non commerciale — Non puoi usare quest'opera per fini commerciali.



Condividi allo stesso modo — Se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con una licenza identica o equivalente a questa.

Prendendo atto che:

* **Rinuncia** — E' possibile rinunciare a qualunque delle condizioni sopra descritte se ottieni l'autorizzazione dal detentore dei diritti.

* **Pubblico Dominio** — Nel caso in cui l'opera o qualunque delle sue componenti siano nel pubblico dominio secondo la legge vigente, tale condizione non è in alcun modo modificata dalla licenza.

* **Altri Diritti** — La licenza non ha effetto in nessun modo sui seguenti diritti:

- o Le eccezioni, libere utilizzazioni e le altre utilizzazioni consentite dalla legge sul diritto d'autore;
- o I diritti morali dell'autore;

o Diritti che altre persone possono avere sia sull'opera stessa che su come l'opera viene utilizzata, come il diritto all'immagine o alla tutela dei dati personali.

* **Nota** — Ogni volta che usi o distribuischi quest'opera, devi farlo secondo i termini di questa licenza, che va comunicata con chiarezza.

La copia completa della licenza è reperibile a questo indirizzo:

<http://creativecommons.org/licenses/by-nc-sa/2.5/it/legalcode>