

folio

Il Magazine di fedoraOnline

In questo numero.

*Team traduttori,
come e perchè*



I nostri servizi:
FUDCON-Paris
FAD-Milano

Le nuove rubriche: software e gaming
Drupal, il CMS del momento

... e molto altro ancora



FOLIO - IL WEBMAGAZINE DI FEDORAONLINE

RIVISTA NON PROFESSIONALE TEMATICA E LIBERA CREATA DAI LETTORI DI FEDORAONLINE, SCARICABILE E DAI CONTENUTI FORNITI DAGLI UTENTI FEDORA IN ITALIA E NEL MONDO.



Folio ed i suoi contenuti sono distribuiti con licenza creative commons, reperibile a questo link: <http://creativecommons.org/licenses/by-sa/3.0/it/>

Questa rivista segue le linee guida dettate dal Fedora Project reperibile all'indirizzo http://fedoraproject.org/wiki/Legal:Trademark_guidelines:

Publications

"It is permissible to use the Fedora trademarks in the title and content of a publication, provided that: the use is clearly in reference to the Fedora Project or its software the use does not imply sponsorship or endorsement by Red Hat or the Fedora Project Proper trademark symbols are used in connection with the Fedora Trademarks and the trademark attribution statement must appear as explained in Proper Trademark Use."
Fedora® and the Infinity design logo are trademarks of Red Hat, Inc.

Redazione:

Gabriele TROMBINI mailga@fedoraonline.it

Robert MAYR robyduck@fedoraonline.it

Giuseppe DELVECCHIO virus@fedoraonline.it

Mario SANTAGIULIANA marios@fedoraonline.it

Giuseppe RAVEDUTO wardialer@fedoraonline.it

Alessandro BUGLIAZZINI yattatux@fedoraonline.it

Grafica:

Abaco Technology <http://www.abacotechnology.it>

Collaboratori:

Andrea VERI averi@fedoraproject.org

Marina LATINI deneb@fedoraproject.org

Gianluca SFORNA giallu@fedoraproject.org

Interventi:

Alberto CHIUSOLE bebo.sudo@gmail.com

Giulio ERLER julix.pigface@gmail.com

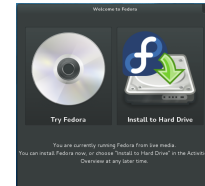
CONTENUTI:

Editoriale

(pag.1)

Fedora 18, spherical cow, come nascono i GO/NO-GO che hanno decretato forse il più lungo tempo d'attesa per una beta.

di Gabriele Trombini



Il Fedora Project

(pag.3)

Il Team Traduzioni italiano, un punto fermo per la nostra Fedora.

di Mario Santagiuliana

Intervista a: Antonio Trande

(pag.6)

Proviamo a mettere in difficoltà Antonio con le nostre dieci domande.

di Gabriele Trombini



FUDCON Paris

(pag.9)

Ecco cosa è il FUDCON.

di Robert Mayr

Linux Day 2012: FAD Milano

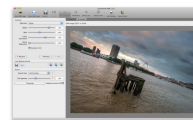
(pag.12)

Resoconto per coloro che si fossero persi un avvenimento primario, a Milano.

di Gabriele Trombini

fedoraactivityday

fad-milano.tk
MILANO 27 OTT 2012



Software: LuminanceHDR

(pag.15)

Diamo una occhiata a LuminanceHDR, progetto del mese di Maggio di Sourceforge.

di Marina Latini

Gaming: Urban Terror

(pag.21)

Ecco la nostra prima recensione di un gioco per pc, uno tra i più famosi "sparatutto" che promette adrenalina.

di Alberto Chiusole (con la collaborazione di Giulio Erler)





Un CMS con enormi potenzialità _____(pag.24)

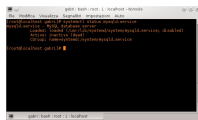
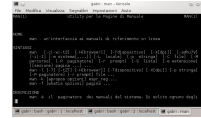
Considerazioni sul CMS del momento, cosa rende Drupal efficace.

di Robert Mayr

Primi passi: il terminale _____(pag.26)

Linea di comando: impariamo a conoscerla (seconda parte).

di Gabriele Trombini



Sistema base: la gestione dei servizi _____(pag.30)

Terminiamo il nostro viaggio alla scoperta dei servizi Fedora e della loro gestione (ultima parte).

di Gabriele Trombini

Sistema avanzato: creazione di rpm _____(pag.34)

Costruiamo il nostro pacchetto rpm (seconda parte).

di Gabriele Trombini



Sistema avanzato: il kernel Fedora _____(pag.40)

Introduzione al Kernel Fedora (ultima parte).

di Giuseppe Delvecchio

I comandi sconosciuti, vediamo come utilizzare la wake on lan _____(pag.44)

Andiamo a vedere l'utilità di comandi talvolta incomprensibili e talvolta sconosciuti.

di Giuseppe Delvecchio



Fedora 18 "Spherical cow" _____(pag.48)

Entriamo all'interno del nostro sistema operativo dando uno sguardo alle novità più interessanti della prossima release.

di Robert Mayr

Un po' di allegria con le Follie _____(pag.51)

Potevano mancare i post più pazzi nella storia di FOL?

Redazione



Editoriale

Gabriele Trombini (mailga@fedoraonline.it)

Ritardo sull'uscita della nuova release? - Un vantaggio per l'utente, ma anche la dimostrazione che Fedora non è una Rolling Release

Carissimi lettori, potremmo cominciare questo primo numero ufficiale di Folio nei più svariati modi, ma torneremo sempre a parlare di lei: Fedora 18!

Ci si ricorderà per molto tempo di questa release anche per il clamoroso ritardo con cui è uscita, ma soprattutto perchè introduce delle feature importantissime dalle quali approfitteranno anche le versioni successive.

E poi, perchè Fedora 18 è nettamente la migliore versione delle ultime 3 o 4.

Strano, penserà qualcuno di voi, avevo letto tutt'altro in giro.

Vero!

Le critiche a *"Spherical Cow"*, come viene chiamata F18, non sono mancate e non mancheranno, e sicuramente in tutto questo ciclo di rilascio qualcosa è andato storto, a partire dalla decisione di portare il nuovo Anaconda già in questa versione.

Il team di sviluppo, però, non ha mai abbandonato la strada che si era prefissata e non ha mai preso delle decisioni affrettate.

Nei meeting GO/NO-GO, in cui si decide se una versione possa venire rilasciata o meno, le discussioni erano serrate, spesso duravano ore e alla fine il Quality Assurance ha sempre preso le decisioni giuste, alla faccia di chi dice che Fedora è altamente instabile.

Nei blog e nelle mailing list parlavano di Fedora come una Rolling Release, ed il ciclo di vita delle versioni potrebbe davvero fare pensare che lo sia ma, proprio per via dei ritardi voluti dal FESCo e dal Q/A, si è dimostrato nei fatti che tale non sia.

E' evidente la cura a cui mirano sviluppatori e community anche solo per poter lanciare una Alpha od una Beta.

Per ciascuno di questi step, il Project ha stabilito dei parametri che possiamo verificare pubblicamente:

• Criteri:

- F18 Alpha http://fedoraproject.org/wiki/Fedora_18_Alpha_Release_Criteria;
- F18 Beta http://fedoraproject.org/wiki/Fedora_18_Beta_Release_Criteria;
- F18 Finale http://fedoraproject.org/wiki/Fedora_18_Final_Release_Criteria.

Chiaramente, quindi, l'esistenza di questi criteri fa sì che cada il concetto fondante di rolling release.

Sotto la responsabilità del Q/A, ad ogni GO/NO-GO meeting vengono esaminati i parametri per ogni rilascio e, se superati, la versione finale (e solo quella) viene dichiarata GOLD altrimenti si dovrà riunire ancora la settimana successiva finchè il rilascio non verrà definito, appunto, GOLD.

Esiste anche una modalità di controllo visuale che definisce lo stato in base alla risoluzione dei bugs:

• Stato dei rilasci:

- F18 Alpha <http://qa.fedoraproject.org/blockerbugs/milestone/18/alpha/stats>;
- F18 Beta <http://qa.fedoraproject.org/blockerbugs/milestone/18/beta/stats>;
- F18 Finale <http://qa.fedoraproject.org/blockerbugs/milestone/18/final/stats>.

A proposito di quanto sopra, ovviamente, essendo ormai a pochi giorni dal rilascio, l'unico grafico da tenere in considerazione è quello relativo alla versione finale.

Infine si può anche dare una occhiata ai bug che ne impediscono la dichiarazione di GOLD:

• Bugs bloccanti per rilascio:

- F18 Alpha <http://qa.fedoraproject.org/blockerbugs/milestone/18/alpha/buglist>;
- F18 Beta <http://qa.fedoraproject.org/blockerbugs/milestone/18/beta/buglist>;
- F18 Finale <http://qa.fedoraproject.org/blockerbugs/milestone/18/final/buglist>.

La comunità Fedora, quindi, ha dei criteri di controllo e verifica sui rilasci che ne determinano la pubblicazione o viceversa.

Tutto ciò dimostra (perchè questi ritardi ne sono la dimostrazione) che Fedora non è una Rolling Release e che i suoi utenti ne sono avvantaggiati e non possono essere considerate cavie per RedHat™. Per il resto....

...buona lettura!

Giusto per curiosità, i bugs NTH (Nice To Have) sono i bugs considerati non bloccanti ai fini di un rilascio, ma comunque di elevata rilevanza come:

- bugs che violano i criteri ma su desktop non predefiniti;
- bugs che non consentano di caricare il sistema grafico;
- bugs di installazione rilevanti che non sono compresi nei criteri.





Team traduzioni

Mario Santagiuliana (marios@fedoraonline.it)

Nel numero scorso abbiamo visto quali e quanti gruppi siano attivi nel progetto fedora: in questo numero cominciamo a scoprire uno tra i più importanti. Vi siete mai chiesti perchè potete installare la vostra lingua in Fedora?

Tradurre è un processo cui spesso è richiesta una competenza specifica e ci sono persone che studiano e si laureano appositamente in questa disciplina per offrire la massima qualità e professionalità a questo lavoro.

Nel Fedora Project non è necessario essere laureati in Traduzione per poter contribuire alla localizzazione del software o della documentazione e non è nemmeno necessario essere dei profondi conoscitori della lingua Inglese.

L'importante è aver la voglia di dedicare un po' del proprio tempo per far crescere il software libero in Italia.

Fedora Project:
traduzione?
localizzazione?
.po e .pot?
Ma che roba è?

Mario Santagiuliana (marios@fedoraonline.it)

Nella pagina wiki della comunità italiana di FOL (http://doc.fedoraonline.it/Tradurre_Fedora), sono presenti tutti i collegamenti necessari per reperire le informazioni su come entrare a far parte del team di traduzione Italiano.

Inoltre, tramite la pagina specifica delle linee guida per i nuovi traduttori (<https://fedoraproject.org/wiki/L10N/Teams/Italian/nuovitradduttori>), il team Italiano ha cercato di spiegare, nel modo più semplice possibile, come avviene il processo di traduzione e le regole che si sono dati.

Il tutto risulta, nonostante gli sforzi, ancora poco chiaro a molte persone. Specialmente quando si inizia a parlare di file **.po** e file **.pot**.

Cercherò di fare un po' di chiarezza in proposito rispondendo a delle domande che mi ero posto a suo tempo e, chiarendo fin da subito: *nonostante tutte le regole da osservare, tradurre un software in Fedora è davvero semplice, basta prendere un file .po o .pot e tradurre le varie stringhe con un software apposito.*

Ora ci si domanderà, come feci io, che diavole è un file .po? E un file .pot? E che software apposito dovrei utilizzare?

Procediamo con ordine; partiamo da **L10N**. Che è?

Si tratta dell'abbreviazione del termine anglosassone *Localization* cioè Localizzazione (L iniziale + 10 che sono i caratteri sostituiti + n finale).

La localizzazione è l'aggiunta di alcune caratteristiche che permettono ad un prodotto, nel nostro caso il software o la sua documentazione, di essere utilizzato all'interno di altre località.

Ad esempio, un software o una guida scritta in Inglese viene localizzata in Italiano tramite il processo di Traduzione.

La localizzazione è possibile tramite l'Internazionalizzazione (o vista con l'abbreviazione Inglese **I18n**) del software o del documento, cioè con l'adozione di tecniche e strumenti appositi per permettere ad una applicazione o ad un documento di essere poi localizzati, semplifichiamo con tradotti, in altre lingue. I concetti di L10N e I18n sono molto più articolati ed in rete si possono trovare spiegazioni molto più approfondite.

Ora vi starete chiedendo: si ma cosa sono i file .po o i file .pot? La risposta è molto semplice: sono dei semplici file di testo che vengono creati dagli sviluppatori del software libero con l'uso di Gettext (<http://www.gnu.org/software/gettext/>).

Gettext è una suite di strumenti liberi che permette agli sviluppatori di internazionalizzare il proprio software.

Ogni messaggio stampato nel terminale o in una finestra, ogni stringa di testo presentata dal programma, tramite gettext viene recuperata ed inserita all'interno di un file di testo, riconosciuto dall'estensione .pot.

Il file .pot è dunque un file grezzo che contiene tutte le stringhe del software, ma può essere anche di una guida o di un documento, che poi dovrà essere tradotto.

Il file .po non è altro che un file .pot rinominato in .po ed indica, generalmente, un file che si sta traducendo o che è stato tradotto.

Cerchiamo di capire meglio come funziona la cosa in modo molto sintetico.

Generalmente uno sviluppatore con un suo programma stampa a video dei messaggi di testo (stringhe) con delle funzioni particolari (echo, printf o altro).

Ad esempio un codice che stampa la frase *Hello world* potrebbe essere:

```
$ echo(Hello world!);
```

Per internazionalizzare il codice precedente con l'uso di gettext lo sviluppatore potrebbe riscrivere la riga così:

```
$ echo(gettext("Hello world!"));
```

In questo modo il codice del software viene internazionalizzato.

Ora lo sviluppatore, tramite dei programmini legge il codice sorgente ed estrae tutte le stringhe che ha internazionalizzato per metterle all'interno dell'ormai famoso file .pot.

Il file .pot risultante potrebbe avere un aspetto simile a questo:

```
#:file_sorgente:1
msgid "Hello world!"
msgstr ""
```

I traduttori dunque non fanno altro che prendere questo file .pot, rinominarlo in .po e tradurre il testo. Il file .po risulterà alla fine così:

```
#:file_sorgente:1
msgid "Hello world!"
msgstr "Ciao mondo!"
```

Dunque è davvero semplice tradurre, basta prendere un file .po esistente, probabilmente già semi-tradotto, o un file .pot da rinominare e poi si traduce.

Io a questo punto mi ero domandato: ma se ci sono stringhe uguali e ripetute? E se un software viene aggiornato ed introduce delle nuove stringhe da tradurre? Devo rifare tutto il lavoro anche per le stringhe già tradotte? Niente paura, gettext è molto potente e non ripresenta le stringhe doppie nel file .pot. Mentre per gli aggiornamenti o lo sviluppatore stesso o il traduttore può prendere il file .pot aggiornato ed il vecchio file .po già tradotto o semi-tradotto e aggiornarlo sempre con l'uso di gettext, evitando di perdere le cose già tradotte.

Anzi alcune frasi potrebbero essere anche ricreate in base ad una corrispondenza di traduzioni precedenti. Lo strumento che potrebbe utilizzare lo sviluppatore per fare questo è il comando *msginit* di gettext.

Per completare il processo di internazionalizzazione lo sviluppatore poi prenderà i file .po tradotti e li convertirà, sempre con l'uso degli strumenti di gettext, nello specifico *msgfmt*, in file con l'estensione .mo (sono file binari) che verranno caricati poi a seconda della lingua selezionata dall'utente sul proprio sistema.

Rimane ancora un quesito: che programma utilizzare per tradurre i file .po? Trattandosi di file di testo si può utilizzare un banale editor di testo come Vim. Ma per semplificarsi ulteriormente il lavoro

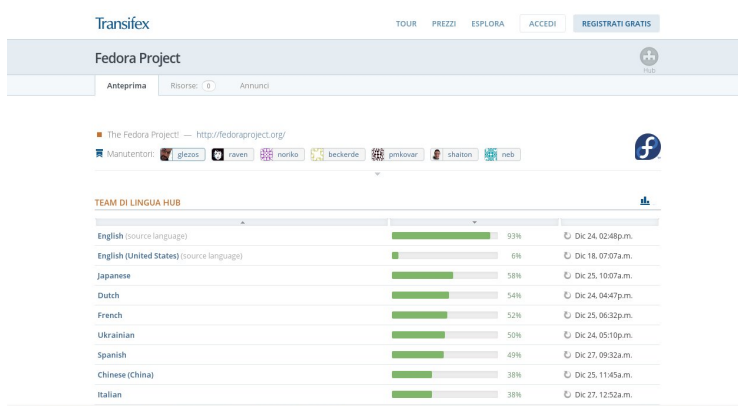


Figura 1: stato dei lavori su Transifex

di traduzione è più utile utilizzare le potenzialità di software appositi che permettono la creazione di un glossario personale e che creano dei database con delle memorie di traduzione interne in grado di suggerire e parzialmente autotradurre delle stringhe che si dovranno solo convalidare, semplificando e velocizzando di molto il lavoro di traduzione.

I maggiori software utilizzati sono in KDE **Lokalize**, in GNOME **Poedit** oppure si può provare **Virtaal**. Esistono comunque moltissimi altri software per fare questo.

E tutti i file .po di Fedora dove stanno?

Il team di localizzazione ed il team di internazionalizzazione da molto tempo si appoggiano alla piattaforma *Transifex* per immagazzinare, creare, distribuire e tradurre la maggior parte dei file .po necessari agli sviluppatori di Fedora.

Transifex.com è una piattaforma web di localizzazione.

È nato come progetto del Google Summer of Code per risolvere alcuni problemi del Fedora Project inerenti alla localizzazione, appunto.

Il progetto è poi cresciuto diventando una società a sé stante che offre i propri servizi in modo gratuito agli sviluppatori di progetti di software libero.

Le ultime versioni di Transifex permettono di tradurre direttamente online i file .po e forniscono anche degli strumenti di convalidazione delle stringhe tradotte da parte di alcuni iscritti che hanno il ruolo di revisori.

I file .po si possono scaricare e tradurre in locale per poi caricarli successivamente.

È anche possibile scaricare da Transifex dei database di traduzione in modo che il proprio programma di traduzione locale possa funzionare al meglio e aiutare il più possibile il traduttore.

Ultimamente alcuni sviluppatori o proprietari di pacchetti di software in Fedora si stanno appoggiando ad una seconda piattaforma di traduzione online che è <http://zanata.org> questo progetto, simile a Transifex, per certi versi, ed è sponsorizzata da RedHat in persona.

Non permette però il download dei file .po o .pot.

I file devono essere tradotti solamente online, in questo modo Zanata è in grado di creare un enorme database di traduzione di aiuto per i propri traduttori.

Ora che si è capito meglio cosa sta dietro alle quinte di un processo di traduzione di un software in Fedora non resta che entrare a far parte del team Italiano di traduzione:

http://fedoraproject.org/wiki/L10N_Italian_Team_Nuovi_traduttori.



Figura 2: pagina FOL del team traduzione

Intervista a: Antonio Trande

Gabriele Trombini (mailga@fedoraonline.it)

Le interviste di Folio - Antonio Trande

Questa volta abbiamo rivolto le nostre 10 domande ad Antonio Trande, che ci ha fornito risposte interessanti.



Chi è Antonio Trande?

classe 1980, tecnico bio-chimico ed autodidatta in informatica.

La sua vita professionale ha poco a che fare con Linux (di cui si occupa dal 2006) e l'informatica in generale. Gli anni di studio scolastico erano rivolti alla Chimica e alla Microbiologia; dalla maturità in poi ha lavorato prevalentemente nella produzione, nel controllo qualità ed in parte anche nella ricerca in più industrie farmaceutiche in Italia.

Usa Fedora e sue derivate dal 2006; dal Dicembre 2010 è entrato a far parte dei collaboratori italiani del fedoraproject, in particolare tra gli ambasciatori e tra i traduttori; rispettivamente si occupano della divulgazione e della traduzione software/documentale. Inoltre è impegnato nella pacchettizzazione di RPM per il team di MOC, un player audio da console molto potente per sistemi LINUX/UNIX (Download [<http://moc.daper.net/download>]).

Il suo blog:
<http://fedora-os.org>

FOL: 1) Come è il tuo desktop, come preferisci averlo?

Antonio: Pulito, funzionale, veloce. Non serve altro. Il PC è uno strumento che deve aiutare l'uomo, prima di essere (ormai) anche un "gadget" da mostrare ad altri con colori, luci, numeri ed effetti speciali.

Utilizzo prevalentemente XFCE e Gnome; sebbene proprio le ultime versioni di Gnome si allontanano dalla concezione classica del Desk-

top così come lo abbiamo conosciuto fino ad ora, ritengo sia utile studiarlo e capirlo permettendoci di mantenere libera la mente da certe restrizioni che non hanno ragione di esistere. In ogni caso facciamo sempre in tempo a tornare sui nostri passi, questo è il bello del software libero.

FOL: 2) Quale è il programma/funzione che ti ha impressionato di più?

Antonio: Indubbiamente il Terminale. Prima che iniziassi ad utilizzare sistemi GNU/Linux non avevo la più pallida idea di cosa fosse questo potente strumento, eppure si utilizza anche in altri sistemi operativi proprietari maggiormente noti. Non solo ti permette di capire come lavorano certe operazioni ma, una volta scoperto in tutte le sue funzioni, risulta anche sicuro, veloce, adattabile e potente.

FOL: 3) Quale è stata, se c'è stata, la delusione più cocente che ti ha dato Fedora?

Antonio: Come posso essere deluso da un programma. Le delusioni sono ben altre ... A parte le battute, per adesso sono contento di poterla usare a mio piacimento.

FOL: 4) De Icaza ha diagnosticato la morte del Desktop su Linux, visto l'impegno profuso su Gnome3 dal Project, tu cosa ne pensi?

Antonio: Non entrerei troppo nel gergo tecnico perché non ho le competenze per farlo. La frase che più mi ha colpito è stata: "L'ecosistema che ha preso vita con Apple OSX è semplicemente impossibile da ottenere con Linux oggi"; mi chiedo io sia questo l'obbiettivo oggi perché se così fosse non penso che sia lo stesso del mondo FOSS. Tuttavia sono d'accordo con lui quando dice che le "posizioni di potere" cercate

da certe distro rischiano di compromettere seriamente gli sforzi di tutti; l'incompatibilità tra distro porterebbe grande dispendio di energia da parte degli sviluppatori terzi e paradossalmente avremmo l'effetto opposto a quello cercato e cioè l'abbandono del supporto ai sistemi GNU/Linux. Il primo esempio che mi viene in mente è Unity; tempo fa nella mailing list di sviluppo del progetto alcuni ragazzi hanno presentato Unity pacchettizzato per Fedora, bene diversi pacchetti core di Gnome venivano letteralmente rimpiazzati da quelli adattati per Unity; due shell di Gnome non basate sugli stessi componenti. Questo non è accettabile. Quindi si dovremmo metterci d'accordo almeno sul mantenimento di una base comune su cui ognuno può lavorarci, sempre e comunque nel rispetto degli ideali del software libero.

FOL: 5) Quale è la principale qualità che deve avere un Ambassador?

Antonio: A mio parere è la tenacia. L'ambasciatore (specialmente in Italia) di un progetto internazionale nato negli USA è una figura border line principalmente per i seguenti motivi: 1) Non è un lavoro nel senso comune del termine, si è spinti dalla volontà di "fare qualcosa per gli altri" ma nel nostro bel paese si parte dalla convinzione che se non si ha un ritorno economico allora è un qualcosa di poco serio, poco affidabile e da sottovalutare a prescindere da tutto. 2) Usare termini nuovi, spesso in inglese, come opensource oppure project oppure chiamare un sistema operativo fedora (nome di origine russa poco usato in Italia) oppure semplicemente dire di usare altri programmi per un uso comune che non si chiamano Windows Media Player o Nero o Internet Explorer porta i tuoi interlocutori a perdere subito l'orientamento. 3) Parlare di software gratuiti equivale erroneamente a parlare di software scadenti. L'ambasciatore deve essere paziente, altruista e fermo nelle sue convinzioni.

FOL: 6) È facile collaborare con il Project?

Antonio: Non sempre purtroppo. L'ostacolo è principalmente farsi capire al meglio con una lingua che non è la tua da persone che a loro volta

non usano l'inglese come lingua madre. Culture diverse (stiamo parlando a livello mondiale) devono fondersi in un unico progetto usando prevalentemente strumenti di comunicazione poco "umani" come mail, chat, blog; è già difficile capirsi parlando di persona, figuriamoci dietro ad una tastiera in un'altra lingua.

"...se un software è costoso allora è valido, se non posso permettermi di pagare una licenza allora mi procuro lo stesso software "valido" per vie meno legali. Per vincere queste radicate convinzioni serve lavorare sui ragazzi già nelle scuole, menti giovani sono più adatte ad acquisire una realtà non-prefissata."

Antonio Trande

FOL: 7) Il Fedora Project è ben organizzato o noti delle lacune, quali?

Antonio: Sono entrato direttamente nel Fedora Project da pochissimi anni; avendo, il Fedora Project, un raggio d'azione molto ampio, ogni ambassador non può conoscere le sfumature di ogni paese partecipante (ricordo che il Progetto è presente in tutto il mondo) di conseguenza non posso avere un quadro completo; tuttavia credo di poter dire che in Italia è poco presente nel "mondo reale": ad esempio una piccola rivista come Folio nelle edicole non sarebbe male, distribuirla gratuitamente nelle scuole e nelle Università non sarebbe una cattiva idea. Se poi ci vogliamo aggiungere anche qualche banner pubblicitario in "siti web chiave" (che non siano forum tecnici o blog tematici) allora sarebbe ben evidenziato.

FOL: 8) L'Italia è un paese difficile per promuovere Fedora?

Antonio: L'Italia è un paese difficile per tante cose; non è facile convincere una persona che si può uscire da schemi prefissati da anni di imposizione di un mercato basato sulla proprietà del software. Come dicevo prima, se un soft-

ware è costoso allora è valido, se non posso permettermi di pagare una licenza allora mi procuro lo stesso software "valido" per vie meno legali. Per vincere queste radicate convinzioni serve lavorare sui ragazzi già nelle scuole, menti giovani sono più adatte ad acquisire una realtà non-prefissata.

FOL: 9) In che direzione dovrebbe andare il Project?

Antonio: A mio parere la chiave è la scuola; a partire dalle scuole medie fino alla maturità con corsi gratuiti sul software libero in generale,

scuole: noi ambasciatori come pionieri del software libero con il benessere del Fedoraproject e degli Istituti scolastici.

FOL: 10) Il Fedora Project avrà un termine? Se sì a cosa sarà dovuto?

Antonio: Non posso sapere una cosa del genere. Mi preoccuperei se il software libero avesse una fine perché allora significherebbe che anche il Fedora Project non avrà motivo di esistere.

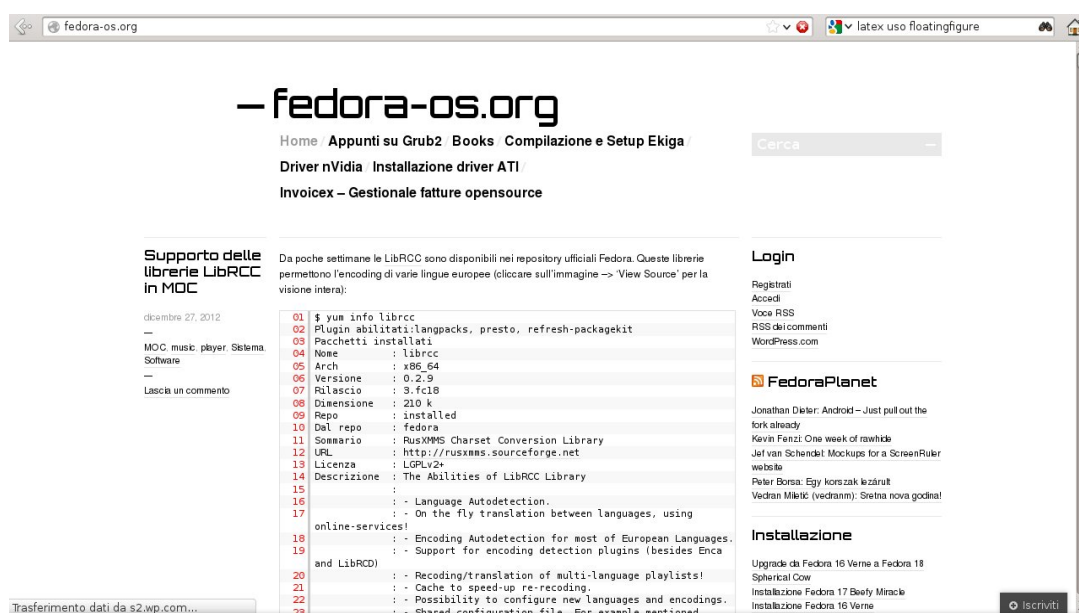


Figura 3: il sito di Antonio Trande

su cosa significhi entrare in questo mondo per la propria formazione personale, per lo sviluppo delle proprie capacità d'interazione con un codice sorgente, per la libertà di poterne usare uno (programma) per i propri bisogni e non per le manie di conquista del mercato di una software house che fa del closed source la propria fonte di ricchezza rinnovabile. Probabilmente attirerò le antipatie di molti ma penso che i Linux Day siano ormai superati, utili solo come ritrovo annuale dei "soliti noti" cioè gente che conosce GNU/Linux da anni; chi è al di fuori sembrerebbe un pesce fuor d'acqua capitato lì per caso. Quindi investire di più nella formazione nelle

FUDCON -Paris

Robert Mayr (robyduck@fedoraonline.it)

Cosa succede ad un FUDCon? Ve lo raccontiamo per filo e per segno.

Il FUDCon è uno degli eventi più importanti di ogni singola area del mondo e si svolge una volta l'anno. Per l'area Euro, nominata EMEA, la *'Fedora Users and Developers Conference'* si è svolta a Parigi dal 13 al 15 ottobre 2012.

Durante le giornate del FUDCon si alternano presentazioni su argomenti specifici da parte di contributors o sviluppatori con workshops o hackfests, in cui si lavora su iniziative e progetti particolari. La partecipazione è sempre aperta a tutti.



Figura 4: la sede del FUDCon 2012 è stata Parigi

Due filoni che quest'anno hanno contraddistinto il FUDCon sono stati l'architettura ARM e Openshift, piattaforma di cloud computing offerta da RedHat.

Ma un passo alla volta, di seguito vi racconto la mia esperienza personale.

Prima volta a un evento Fedora di un certo tipo, prima volta che incontro gli amici del Fedoraproject che fino a quel momento conoscevo soltanto in modo virtuale e attraverso i loro nick e purtroppo sono anche l'unico che dall'Italia ha deciso di partecipare al 'FUDCon Paris 2012'.

Partito a notte fonda arrivo a Parigi alle 8:00 di mattino, tempo di raggiungere il centro passando per l'Arc de Triomphe riesco a trovare facilmente l'Eurosites George V Business Center, la location della prima giornata, che contemporaneamente ospita anche l'Open Forum, al quale partecipa tutto il mondo Open Source.

Mi registro e comincio con la famosa operazione nick -> faccia -> nome reale e dopo un po', scoraggiato dai pessimi risultati mi accomodo nella Conference Hall, in cui la Leader del Progetto Fedora, **Robyn Bergeron**, sta aprendo il FUDCon facendo una lunga introduzione su Fedora ai presenti in aula.

Visto che la prima tattica non ha funzionato decido di andare all'attacco e mi presento a ogni essere

umano che porta al collo il laccio di Fedora, stavolta le cose vanno meglio e in pochi minuti mi trovo a chiacchierare con **Pierre-Yves Chibon (pingou)** e **Kèvin Raymond (shaiton)**.

Piano piano riesco a conoscere anche tutti gli altri e sono davvero onorato di potermi confrontare con personaggi come **Tom Callaway (spot)**, **Christoph Wickert (cwickert)**, **Jiri Eischmann (sesivany)** o **Remi Collet (remi)**, non solo su argomenti inerenti alla nostra distribuzione preferita, ma anche su temi di tutt'altro tipo.

Dopo un pranzo 'al sacco' cominciano i talk su varie tematiche, e ognuno dei circa 80 partecipanti prende posto in quelli che ritiene personalmente più interessanti.

Io decido di ascoltare subito **Marek Jelen** sull'argomento Openshift, e non sapendo quasi nulla sul progetto sono stato ripagato, davvero un talk interessante in cui ci siamo scambiati anche le immagini ISO di Openshift, basato su Fedora 16.

Si susseguono altri talk, come quello di Christoph Wickert sul concetto di Leader in una struttura che non ha una gerarchia ben definita oppure quello sull'infrastruttura da parte di Pierre-Yves Chibon.

Travolto da tutte queste informazioni decido di darmi il colpo finale, inserendomi anche nella sessione pratica di Remi Collet sul packaging, e lui, essendo un guru nel mondo della pacchettizzazione, ha saputo trasmettere un bel po' della sua lunga esperienza.

Il pomeriggio volgeva al termine e siccome l'amico Kèvin Raymond si era offerto di ospitarmi a casa sua per la notte, mi metto d'accordo per la serata, in cui comunque ci saremo rivisti tutti.

Essendo lui l'organizzatore dell'evento aveva anche altri compiti prima di cena, per cui mi sono aggregato a **Fabian Affolter (fab)** e **Gerold Kassube (geroldka)**, due ragazzi che come me avevano già un certo languorino... Ovviamente ci perdiamo in mezzo a Parigi e solo dopo



Figura 5: un momento del talk di Tom Callaway (spot)



Figura 6: Fabian Affolter

un bel piatto di pasta e grazie al segnale GPS sul mio telefonino ritroviamo la strada per il FUDPub, in cui rivediamo tutti i partecipanti per la 'vera' cena, French Pizza e tanta tanta birra :)

Sono a tavola con la squadra ungherese e devo dire che non si finisce mai di conoscere gente nuova, **Zoltan Hoppa (zoltanh7211)** è davvero simpatico e anche i 'vicini' di tavolo **Peter Robinson** e **Dennis Gilmore** non sono da meno.

A notte inoltrata prendo il treno con Kèvin e ci dirigiamo verso casa sua, dove esausti crolliamo nel sonno.

Per dormire abbiamo solo poche ore, perchè la seconda giornata ci attende e si svolge nella Cité des Sciences, un complesso enorme in cui si svolgono anche molte attività educative per le famiglie con bambini.

E' la giornata dei barcamp e ce ne sono davvero tanti, impossibile vederli tutti.

Nella mattinata ascolto Tom Callaway che ci presenta l'esperienza utente da un punto di vista Desktop, veramente interessante e con alcuni spunti su cui si è discusso molto, per esempio il futuro dell'attuale interfaccia grafica.

Sono curioso a cosa portano le sue riflessioni e se ci saranno delle prese di posizione da parte del Fedoraproject.

Naturalmente non potevo mancare al barcamp su 'websites' tenuto da Kèvin, in cui ha presentato quella che sarà probabilmente la futura struttura delle pagine web, e che semplificherebbe di molto la gestione delle singole pagine.

Infine, per concludere la mattinata, torno anche dall'Infrastruttura, in cui Pierre-Yves parla di Cloud Computing legato all'Infrastruttura, un altro progetto che faciliterebbe molte cose anche ai ragazzi della QA (quality assurance) per testare le varie release in ambienti diversi o per personalizzare macchine virtuali. Dopo un buffet abbondante e lo scambio di gadget Fedora la giornata procede per me con il DE Xfce da parte di Cristoph e un barcamp più difficile da seguire, il kernel visto da Josh Boyer, in cui spiega in modo semplice la struttura del 'cuore' del sistema e come svilupparlo.

Infine Fabian ci invita per un barcamp molto intenso sul gruppo Ambassador, in cui vengono nominati nuovi mentor per la zona EMEA e si discute sulla possibilità di definire 'inattivi' gli Ambassador che attualmente non sono più presenti o reperibili.

Una discussione davvero costruttiva e interessante, in cui emergono anche molte richieste per rendere i processi più snelli ed efficaci.

Con la seconda giornata per me il FUDCon termina e faccio ritorno a casa, ma non senza promettermi di esserci anche nel 2013.



Figura 7: il banner del FUDCon Paris



Figura 8: Un saluto dal FUDCon

Linux Day 2012 - FAD Milano

Gabriele Trombini (mailga@fedoraonline.it)

Linux Day - FAD Milano

Come abbiamo pensato di svolgere il FAD all'interno del Linux Day a Milano.

Cominciamo con il precisare che il FAD inteso come evento ambassador, non è il medesimo evento che viene definito nella pagina del wiki del Fedora Project <https://fedoraproject.org/wiki/FAD>, quale, invece, è stato quello che si è svolto in Germania un mesetto fa ed al quale abbiamo partecipato (e che sarà oggetto di cronaca nel prossimo numero).



Figura 9: Linux Day 2012 al Politecnico di Milano

Innanzitutto il Linux Day di Ottobre 2012 ci ha fornito l'occasione di poter avere la possibilità di inserirsi all'interno delle conferenze e dei meeting che il LUG del Politecnico di Milano stava preparando. Quando **Marina Latini** me ne ha parlato, abbiamo subito pensato che avremmo potuto approfittare delle quattro ore stabilite per offrire una panoramica della contribuzione nel Project; quattro ore sono davvero poche, ma in fondo ci ha dato la possibilità di allargare la nostra visione dell'evento considerando la cosa da un punto di vista non della distro bensì dell'utente.

Nei talk, normalmente, si cerca di approfondire e dare una visione a 360 gradi dell'argomento, basandosi più sull'ego del relatore che sulla noia che interviene dopo un po' di tempo.

Un relatore sicuramente competente, tende a sciorinare nozioni e logiche a scapito della dinamicità dell'intervento, correndo il rischio di scivolare lentamente verso il tedio monotematico.

Non avendo l'aspirazione di esaurire alcun argomento all'interno delle quattro ore (cosa impossibile, dal nostro punto di vista, data la complessità di ciascun argomento scelto), abbiamo stabilito di velocizzare le presentazioni in modo da suscitare quantomeno la curiosità dei presenti con il fine ultimo di farci porre le domande in un secondo momento o negli spazi appositi all'interno di forum, meeting e canali IRC.

Di sicuro gli argomenti avevano un appeal eccezionale (pacchettizzazione, traduzioni, kernel, infrastruttura, website, design, virtual machines) ed i nomi di ragguardevole livello per Fedora in Italia (Andrea Veri, Federico Vaga, Samuel Iuri, Massimiliano Bugni, Gianluca Sforna, Mario Santagiuliana, Robert Mayr oltre, naturalmente a Marina Latini).

Inoltre, su idea di Gianluca, abbiamo pensato di mantenere parallelamente sessioni pratiche di packaging e di traduzione, che si sarebbero svolte in un'aula separata.

Personalmente ritengo che la formula adottata sia stata davvero entusiasmante dimostrando come si possa accedere ai diversi gradi di partecipazione al Project da parte dei contributors e, ritengo, che l'esperienza sia da ripetere.



Figura 10: Marina Latini (@deneb)



Sarà però da affinare la fattibilità di questo genere di incontri in un evento a se stante, avulso dai Linux Day, i quali, a mio avviso, cominciano ad avvertire il logorio del tempo.

Non possiamo esimerci dall'inserire anche una cronologia dell'evento, che ha visto la presenza costante di circa 30 ascoltatori:

- come ovvio, il primo intervento ha visto Marina Latini impegnata nella presentazione del Fedora Project e dei gruppi di interesse. Sono state sottolineate le attività con particolare attinenza a quelle che sarebbero state oggetto della sessione;
- Gianluca Sforza ha trattato il packaging spiegando come costruire il proprio pacchetto in accordo con le linee guida di Fedora e quali sono gli strumenti necessari. Ha raccomandato l'installazione degli strumenti di sviluppo dopodichè ha proseguito illustrando le voci, ed il significato, presenti all'interno delle sezioni del file `.spec`, soffermandosi sull'importanza di avere uno spec file ben fatto, comprensivo delle dipendenze. Una volta affinato il file spec Gianluca ha cominciato il test di compilazione aiutando coloro che si sono trovati nei guai e spiegando gli errori. Dopo il suo intervento Gianluca si è spostato nell'aula riservata alla sessione pratica;
- Robert Mayr ha illustrato il gruppo websites, precisando le tipologie di software in uso nel team, le conoscenze di base che si deve avere e quali strumenti (a cominciare git) siano particolarmente efficaci. Ha inoltre spiegato il sistema di ticketing necessario per trovare un coordinamento equilibrato;
- il lavoro del team infrastructure è stato trattato da Andrea Veri con un viaggio all'interno del loro flusso di lavoro, tangenziale a quello websites e del software usato;
- il gruppo design è stato presentato da Samuel Iuri, che ha spiegato le linee guida e le differenze tra un logo Fedora ed uno presunto tale. I colori sono molto importanti per la comunità Fedora, infatti l'uso di quelli ufficiali impone una palette approvata dal Fedora Project. Ha anche illustrato una scelta tra i più comuni software per la realizzazione di logo, banner, header e quant'altro sia possibile immaginare;



Figura 11: C'ero anche io Gabriele Trombini (@mailga)

- uno dei talk più importanti di questo evento era legato al gruppo localizzazione. Mario Santagiuliana si è particolarmente soffermato sull'importanza del lavoro del I10n team.

Dal suo intervento risulta ben chiaro che se noi possiamo usare il sistema operativo nella nostra lingua, lo dobbiamo a questo gruppo che è in grado di tradurre dall'inglese.

Ogni stato ha il proprio gruppo ed ogni gruppo ha molto lavoro sulle proprie spalle anche se sono aiutati da un software di sistema che gli consente di tradurre molto più facilmente che a mano;

- la spiegazione della compilazione del kernel secondo le modalità Fedora è stata fatta da Federico Vaga (che ha anche una certa esperienza sui moduli kernel), il quale ha esposto i concetti del kernel ed ha specificato l'approccio allo stesso che l'utente deve avere. E' partito dal download e si è focalizzato principalmente sul file `.config` e sulle `patches`. Infine ha spiegato come compilare un kernel con le patch usando gli strumenti forniti da Fedora;
- l'ultimo argomento era strettamente legato al Linux Day, dedicato al free software nella piccola e media impresa, ed era effettivamente di difficile trattazione. Non volevamo parlare di un uso server di Fedora, così abbiamo chiesto a Massimiliano Bugni di dimostrare la funzionalità di Fedora in contesto lavorativo. Ne è venuta fuori una ottima panoramica della virtualizzazione e sulla creazione di propri livecd.



Figura 12: Questo è un imperativo!

Luminance HDR

Marina Latini (deneb@fedoraproject.org)

Un software per la gestione di immagini HDR e HDRI che si è guadagnato il riconoscimento di progetto del mese di Sourceforge

Luminance HDR è un software open source, rilasciato secondo i termini della licenza GNU GPL, pensato per fornire un supporto completo alla realizzazione di immagini HDR o HDRI (High dynamic range imaging).

Installazione da riga di comando

L'installazione su Fedora è molto più semplice perché da Fedora 17, Luminance HDR è già incluso nei repository ufficiali.

Per installare il software occorrerà digitare da terminale:

```
# yum install luminance-hdr
```

Alla schermata successiva sarà visualizzato l'elenco delle dipendenze, la dimensione totale del download, la dimensione totale dell'installazione ed una conferma per proseguire con l'installazione a cui occorrerà rispondere "s" e digitare invio.

A questo punto basterà attendere e luminance sarà pronto all'uso.

Dipendenze risolte

```
=====
```

Package	Arch	Versione	Repository	Dim.
Installazione:				
luminance-hdr	i686	2.2.1-4.fc17	updates	3.3 M
Installazioni per dipendenze:				
fftw-libs-single	i686	3.3.1-3.fc17	updates	456 k
gsl	i686	1.15-4.fc17	fedora	824 k
qt	i686	1:4.8.2-7.fc17	updates	4.4 M
qt-mobility	i686	1.2.0-10.20110922.fc17	fedora	2.8 M
qt-settings	noarch	4.8-19.fc17	updates	26 k
qt-x11	i686	1:4.8.2-7.fc17	updates	12 M
qtwebkit	i686	2.2.2-1.fc17	updates	6.6 M

Riepilogo della transazione

```
=====
```

Install 1 Pacchetto (+7 Pacchetti dipendenti)

Dimensione totale del download: 31 M

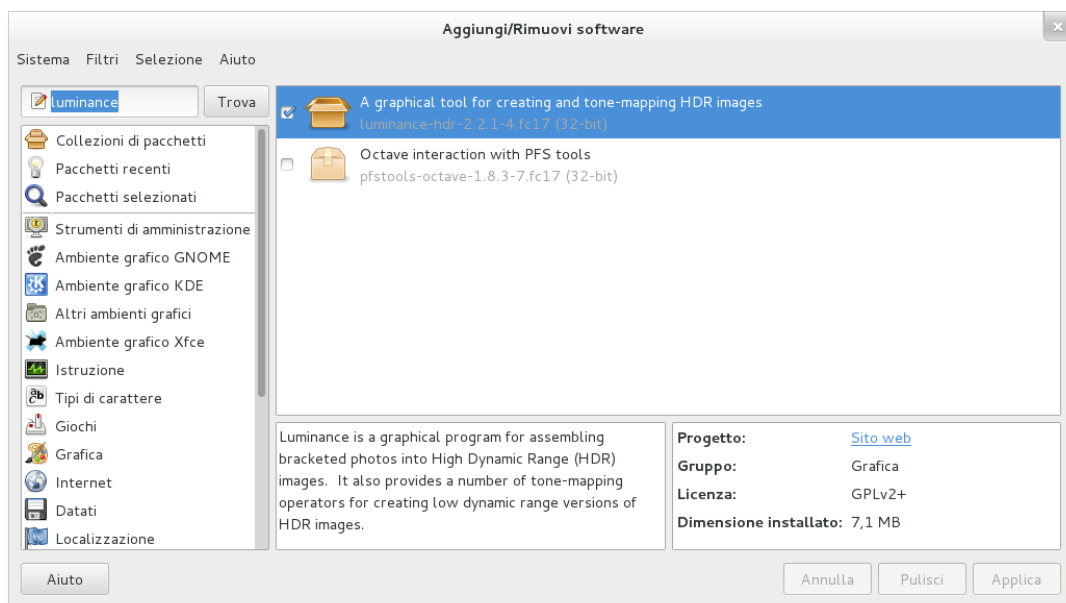
Dimensione installata: 95 M

Procedere [s/N]: s

Installazione dal gestore dei pacchetti

L'installazione dal gestore dei pacchetti è altrettanto semplice, poiché, come dicevamo, il software è già presente nei repository ufficiali.

L'operazione da eseguire sarà cercare il pacchetto Luminance, selezionarlo dall'elenco ed applicare la modifica effettuata.



Formati di immagini supportate

Luminance supporta molti formati di immagini, sia per l'HDR sia per LDR.

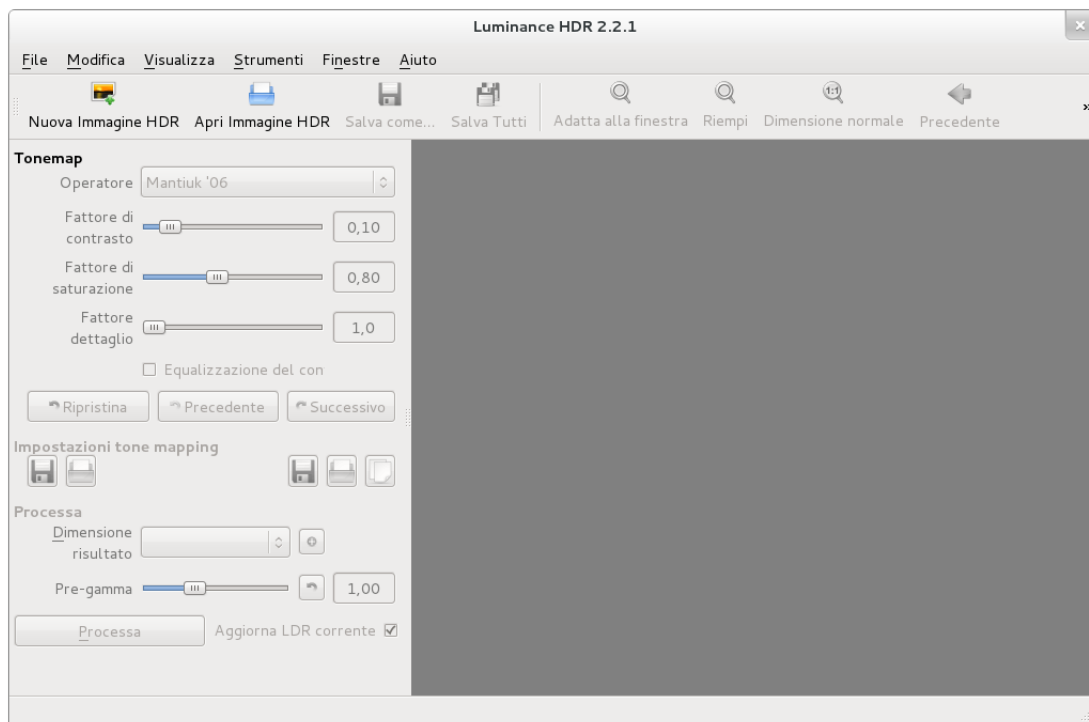
- HDR:
 - OpenEXR (exr);
 - Radiance RGBE (hdr);
 - Tiff 16bit, 32bit float e LogLuv (tiff);
 - Raw;
 - PFS (pfs).
- LDR:
 - JPEG;
 - PNG;
 - PPM;
 - PBM;
 - TIFF (8 bit).

Un esempio pratico

Luminance ha un ottimo wiki in cui trovare guide e manuali: <http://tinyurl.com/LuminanceHDR-Wiki>

A questo punto, dopo aver installato Luminance HDR non rimane che scoprire le funzionalità di questa applicazione con un esempio pratico.

Per prima cosa avviare l'applicazione, si troverà l'interfaccia grafica di seguito riportata

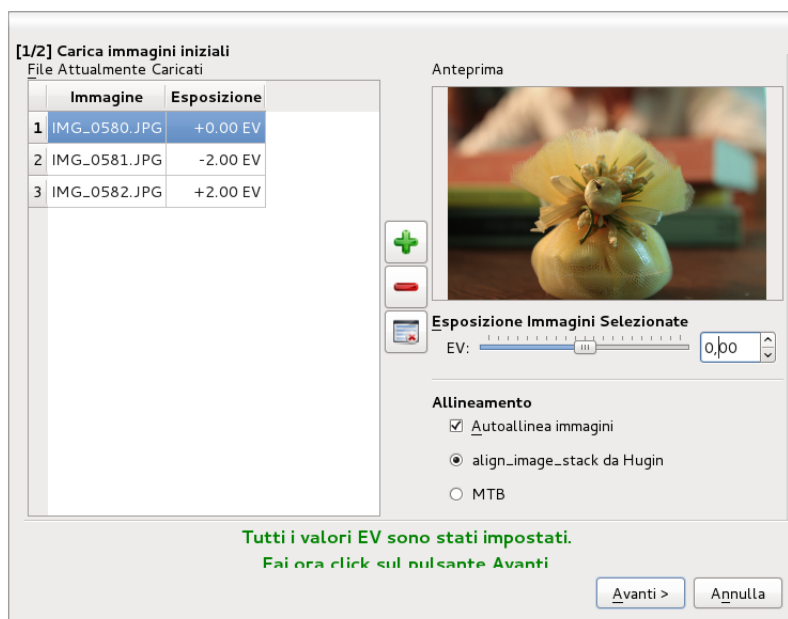


Selezionando la creazione di una nuova immagine HDR si aprirà una procedura guidata per la scelta delle immagini da elaborare.



Quello di cui si avrà bisogno sono almeno tre immagini scattate con esposizioni differenti. Occorre fare molta attenzione, le immagini dovrebbero differire solo per l'esposizione quindi occorre munirsi di cavalletto ed impostare l'autoscatto per ridurre vibrazioni, movimenti e disallineamenti indesiderati.

Dalla schermata successiva selezionare le foto impostando la corretta esposizione.

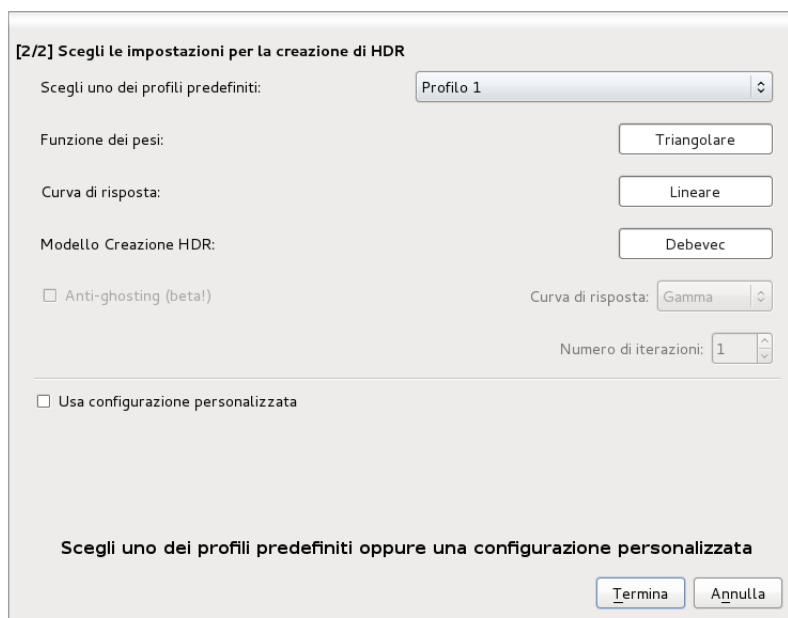


La schermata successiva permetterà di allineare manualmente le immagini ed applicare delle correzioni grazie all'“anti-ghosting”.

Sconsiglio di utilizzare questa ultima proprietà perché attualmente è in stadio beta.

In ogni caso, se le immagini sono state scattate correttamente, si potrà semplicemente andare alla schermata successiva senza dover effettuare modifiche.

Anche alla schermata seguente è possibile semplicemente andare avanti utilizzando il “profilo 1”



A questo punto si è pronti per il tone mapping; per farlo si dovrà scegliere tra una serie d'algoritmi di enhancement, chiamati Operatori.

In questa fase si potrà fare tutti gli esperimenti e modifiche che si preferiscono per ottenere il risultato desiderato.

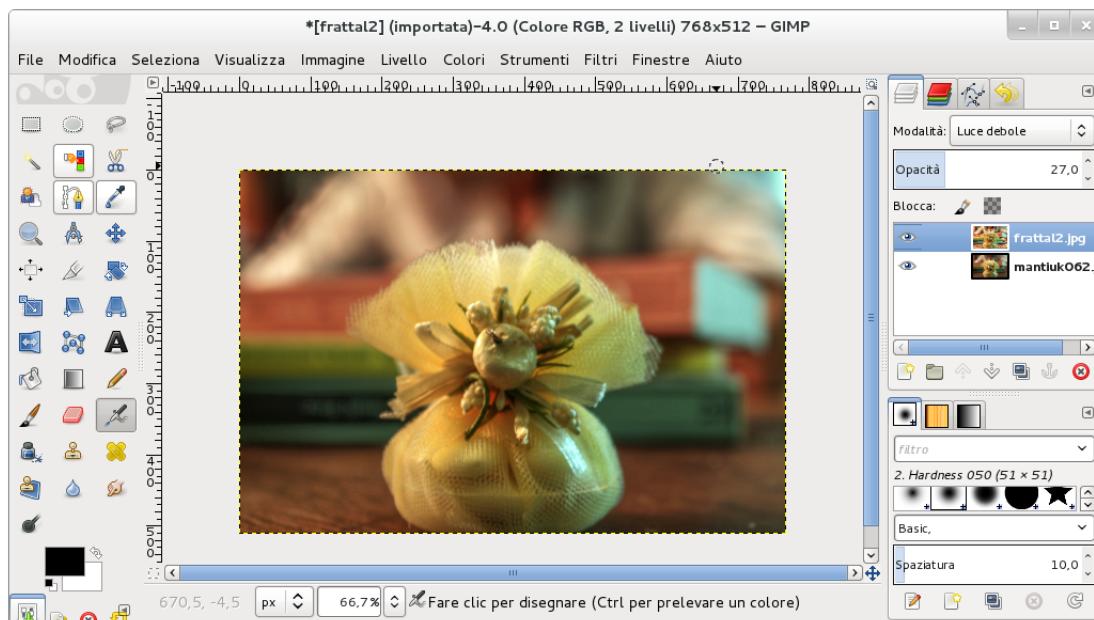
Personalmete preferisco utilizzare gli algoritmi Fattal e Mantiuk'06 per poi ricombinare le due immagini

ottenute con GIMP.

Di solito utilizzo l'immagine ottenuta con il Frattal; sarà la HDR pura mentre quella ottenuta dal Mantiuk servirà a migliorare le ombre e le luci dell'immagine finale.

Ottenute le due immagini non rimane che rielaborarle con GIMP.

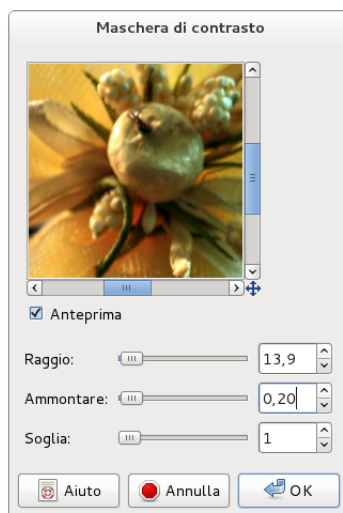
Aprire le due immagini con GIMP, inserendo quella ottenuta con l'algoritmo frattal al livello superiore, con modalità di sovrapposizione Luce debole ed opacità compresa tra il 25% ed il 40%.



L'immagine è già pronta ma volendo si può applicare una maschera di contrasto per migliorarla ancora. Per prima cosa occorre fondere i due livelli per poi inserire la maschera dal menu

Filtri-> Miglioramento-> Maschera di contrasto.

Questi sono i settaggi della maschera che uso solitamente ma ovviamente è possibile riadattarli alle proprie esigenze.



L'immagine è quasi pronta; manca solo qualche piccolo aggiustamento per diminuire il disturbo presente e per migliorare ulteriormente il contrasto e la luminosità.

Per prima cosa occorre duplicare il livello con modalità sovrapposta ed opacità intorno al 15% ed applicare una sfocatura gaussiana con raggio 4 (Filtri-> Sfocature-> Gaussiana).



Le ultime modifiche possono riguardare il contrasto e la luminosità o le curve di colore ma queste ovviamente dipendono dall'immagine che si sta elaborando.



Urban Terror

Alberto Chiusole (bebo.sudo@gmail.com)

collaborazione con Giulio Erler (juliux.pigface@gmail.com)

Uno "sparatutto" per gli amanti del gaming

UrbanTerror è uno sparatutto adrenalinico nato nell'ormai lontano 2000, attualmente sviluppato da Frozen Sand.



Figura 13: Urban Terror, uno sparatutto adrenalinico

È basato su una versione completamente open source del motore grafico del vecchio Quake3, ricordato grazie anche ai giochi da esso derivati (OpenArena, Soldier of Fortune 2 o ancora Tremulous, World of Padman, ecc..).

Il gameplay si basa sull'azione e sulla velocità di reazione.

Siamo infatti di fronte ad un gioco in cui la freddezza e l'agilità di polso contano più della strategia e del numero di giocatori.

Un gamer esperto può facilmente abbattere manciate di principianti armati fino ai denti, ma non abili con mouse e tastiera.

Anche un giocatore alle prime armi noterà immediatamente che, grazie ad uno sprint con l'apposito pulsante,

te, acquisterà una notevole velocità di corsa, ma diminuirà la "stamina", meglio comprensibile come "energia", "fiato".

Questo rende il gioco più realistico rispetto ad altri FPS (FirstPersonShooter) poiché il giocatore dovrà rallentare o fermarsi appunto, per riprendere fiato: anche il calare della vita del personaggio incide negativamente sulla stamina.

Strettamente collegato a questo discorso è il danno localizzato, un'altra caratteristica di UrbanTerror introdotta con la versione 3.0: in sostanza, quando si viene feriti in una parte del corpo, questo provoca danni specifici.

Ad esempio, se veniamo colpiti alle gambe, noteremo subito un rallentamento nella velocità di corsa, se siamo sprovvisti del giubbotto in kevlar i danni al torace saranno più gravi e porteranno ad una morte più veloce, se veniamo colpiti alle braccia perderemo stabilità nell'uso dell'arma.

I colpi alla testa provocano morte istantanea, se non siamo in possesso dell'elmetto che aumenta - di poco - la resistenza ai cosiddetti "Headshot".

Dopo uno scontro, c'è anche il rischio di morire per dissanguamento se non riceviamo una cura adeguata. Esiste inoltre il rinculo, che mina la stabilità della mira dopo un certo numero di colpi.

Armi diverse poi, richiedono tempi diversi di ricarica.

Ogni giocatore può dotarsi di un numero fisso di equipaggiamento, selezionabile dopo la rinascita o raccogliendo oggetti da terra, in numero limitato.

Non troveremo mai, ad esempio, giocatori con due armi primarie contemporaneamente.

Un altro punto a favore di Urban Terror è la varietà delle modalità di gioco, tra cui troviamo:

- **Team Deathmatch** = la più classica delle modalità, in cui due squadre si affrontano in un duello. La fazione che totalizza più punti vince il match.
- **Capture The Flag** = lo scopo del gioco è quello di rubare la bandiera avversaria e portarla sulla propria per fare punti.
- **Team Survivor** = duello all'ultimo sangue. Il giocatore che sopravvive dopo gli scontri, decreta la vittoria per la propria squadra.
- **Capture And Hold** = in questa modalità la mappa è disseminata di bandiere fisse che inizialmente sono neutre (di colore grigio). Se un giocatore giunge su uno di questi punti, lo cattura e ne cambia

il colore in quello della propria squadra. Il controllo per un certo periodo di tempo decreterà l'assegnazione di un punto.

- **Bomb Mode** = la partita è suddivisa in round. Ad ogni round una squadra deve piazzare e difendere una bomba in una tra due locazioni a scelta, mentre l'altra squadra deve cercare di impedirne la collocazione o tentare di disinnescarla. A metà partita i compiti delle squadre si invertiranno.
- **Follow The Leader** = anche qui il match si svolge in più turni. Le due squadre possiedono una bandiera fissa e un giocatore che funge da "Leader della squadra". Ad ogni round le fazioni devono proteggere il proprio comandante in modo da permettergli di giungere sulla bandiera avversaria. Contemporaneamente, occorre impedire al team avversario di fare lo stesso. Il round termina non appena un Leader arriva alla bandiera o una squadra viene annientata. Un giocatore che viene ucciso deve aspettare l'inizio del round successivo per rientrare in gioco.
- **Free For All** = un'altra modalità classica. In questo tipo di gioco, ogni giocatore gioca per se stesso: non sono presenti squadre, è uno scontro "tutti contro tutti". Chi accumula più uccisioni, sarà decretato vincitore.

Le tipologie più diffuse sono le prime tre.

È tuttavia consigliato, ai "novizi", di guadagnare esperienza e consapevolezza in server Free For All.

Il gioco di squadra è, almeno inizialmente, più complicato ma gratificante ed esaltante.

Arriviamo finalmente a parlare dell'armamentario: in UrbanTerror, a differenza di altri videogiochi, tutti i giocatori possono possedere qualsiasi arma da subito, senza essere costretti a cercarle per la mappa (es. Unreal Tournament). È possibile equipaggiarsi con un'arma primaria, una pistola, un coltello (obbligatorio in ogni caso) e altri tre slot con altri equipaggiamenti a discrezione, occupabili con un'arma secondaria, giubbotti ed elmetti in kevlar, granate e molto altro.

L'elenco delle armi soddisferà tutti i gusti: si va da fucili d'assalto (il famigerato Kalashnikov, il potente Negev, ecc..), a fucili di precisione, per poi passare a discrete armi secondarie da combattimento, e molte altre da scoprire giocando.

Prendendo in considerazione il lato tecnico del titolo in esame, notiamo subito che questo è un gioco neppure confrontabile con altri "collegli" dello stesso genere, soprattutto sotto l'aspetto della grafica.

I dettagli sono pochi, le armi sono rozze, i personaggi sono elementari.

La mancanza di risorse di team che programmano giochi Open Source si fa sentire, Urban Terror non riesce certo ad alimentare una concorrenza ad Halo o ad altri grandi nomi.

Il rovescio della medaglia a nostro favore però, è la possibilità per nulla scontata di giocare su macchine poco performanti o con schede grafiche non all'avanguardia.

È stato da noi testato su un vecchio computer con processore Athlon XP, equipaggiato con poco più di 500 megabyte di ram e una mediocre scheda video.

Non troveremo colonne sonore e canzoncine di accompagnamento molto raffinate, in compenso però gli effetti audio sono molti, sia ambientali - come finestre che si rompono se infrante, il cambio del caricatore, rumori di sottofondo con macchine, treni, navi ecc. - che fisici - come il respiro di altri personaggi nelle vicinanze, il rumore di passi o spasmi vari.

Durante il gioco capita inoltre di sentire semplici frasi dei personaggi, registrate in inglese, purtroppo unica lingua del gioco.

Per quanto riguarda gli scenari di gioco, è possibile lottare in centinaia e centinaia di mappe, che vanno dai classici contesti urbani con case diroccate o grattacieli, a scenari in "stile seconda guerra mondiale". Sono presenti depositi petroliferi, fattorie, castelli medievali e centinaia di altre ambientazioni per tutti i gusti, molto ben realizzate.

In alcuni casi ci si potrà addirittura scontrare in mappe con bassa gravità.

Nota positiva è anche la presenza dei cosiddetti Fun Stuff, dettagli estetici applicabili al proprio protagonista, come passamontagna, maschere da ninja, corna, maschere antigas eccetera.

Oltre a correre e saltare, in UrbanTerror possiamo anche scivolare (in gergo PowerSlide) per guadagnare ancora un po' di velocità o addirittura aggrapparci ai cornicioni dei muri.

Troviamo poi la possibilità di saltare i muri (WallJumping).



Figura 14: Una scena d'azione

Gli utenti Fedora possono trovare il gioco nel repository standard.
Nel momento in cui scriviamo, la versione 4.1.1 è facilmente installabile tramite yum:

yum install urbanterror

Il lanciatore si preoccupa di scaricare l'archivio completo e di configurare il gioco per la distro: mentre giochiamo non dovremo preoccuparci di scaricare mappe manualmente perché è compito dell'ottimo autodownloader integrato.



Figura 15: tutte le armi disponibili fin da subito!

Pur avendo più di dieci anni sulle spalle, UrbanTerror è tuttora un videogame apprezzabile ed indicato soprattutto a chi vuole divertirsi senza tante pretese.

La varietà del gameplay e delle ambientazioni consentono un'esperienza di gioco avvincente e sicuramente non noiosa.

Le giustificabili insufficienze – relative e soprattutto non gravi – al comparto tecnico, sono sopperite dal ritmo incalzante e dalla coinvolgente atmosfera che questo titolo riesce a suscitare.

Inoltre, le richieste hardware consentono anche a giocatori occasionali di passare qualche ora di frizzante adrenalina.

Il tutto è arricchito dall'eccellente autodownloader che semplifica e ottimizza l'installazione su un sistema Fedora.

VANTAGGI:

- + varietà di mappe e modalità;
- + autodownloader integrato;
- + eseguibile su macchine obsolete.

SVANTAGGI:

- grafica appena sufficiente;
- realismo ok, ma non alla pari di titoli commerciali;
- colonna sonora grezza.

VOTI:

- * Grafica = 6
- * Sonoro = 6,5
- * Giocabilità = 8,5
- * Longevità = 8
- * Multiplayer = 8,5

Media = 7,5

I voti non vanno intesi come sentenze o verità assolute.

Cercheremo, all'interno dei nostri articoli, di avere un occhio il più possibile obiettivo.

Certamente però, non siamo ne' giornalisti professionisti, ne' esperti del settore, quindi le nostre sensazioni potrebbero essere frutto di esperienze personali o gusti soggettivi.

Drupal

Robert Mayr (robyduck@fedoraonline.it)

Analisi soggettiva di un CMS

Lo sviluppo dei CMS (Content Management System) negli ultimi 10 anni ha contribuito a una certa facilità nella creazione di siti web dinamici, scritti in PHP con l'utilizzo di un database.

Essi sono lo strumento più adatto per chi non ha delle conoscenze approfondite di PHP e HTML, perché mettono a disposizione dell'utente una discreta base di un sito web, che mediamente si può integrare facilmente con appositi moduli o temi grafici.

In questo articolo voglio descrivere meglio le mie sensazioni nell'utilizzo di Drupal, un CMS forse tra i più chiacchierati negli ultimi anni.

Per i più impazienti posso dire da subito che Drupal è effettivamente uno dei migliori CMS in circolazione, ma non è adatto per tutti gli utenti e non si addice indistintamente a tutti gli utilizzi.

Come sempre va valutato bene l'utilizzo finale del CMS, le caratteristiche necessarie per il sito da creare e la familiarità dell'utente stesso nell'uso di un Content Management System.

Nel nostro caso specifico, la valutazione era stata fatta tenendo conto dei seguenti aspetti:

- Codice xhtml strict 1.0;
- Massima robustezza e sicurezza possibile;
- Facilità e flessibilità di inserimento di moduli creati ad hoc;
- Predisposizione a Html 5;
- Documentazione e comunità, moduli disponibili.

Questi 5 aspetti hanno eliminato moltissimi dei CMS in giro, infatti non è così semplice trovare chi soddisfa tutti questi criteri.

D'altra parte, però, rimanevano comunque alcuni tra i CMS più conosciuti, tra cui Joomla, ezPublish, oppure e107.

Sicuramente Drupal è conosciuto per essere un po' più robusto rispetto agli altri per via della sua costruzione, dall'altra parte questo aspetto rende un po' meno facile l'utilizzo del CMS stesso.

Infatti, nonostante abbia utilizzato in passato moltissimo Xoops, Joomla forse risulta essere il più immediato nell'uso quotidiano.

Drupal invece è stato, soprattutto nella fase iniziale un mondo quasi tutto nuovo.

Da qui arriva anche una delle maggiori critiche di tutta la comunità di Drupal, perché la versione 7 è strutturata in un modo completamente diverso, sia rispetto agli altri CMS conosciuti da me fino a quel momento, sia rispetto alla versione 6.

1. I temi e moduli inclusi nel core e quelli aggiunti successivamente risiedono in posti diversi;
2. I file css sono moltissimi e sono suddivisi per tipo e non più per template o modulo;
3. La gestione di javascript risulta in un primo momento fuorviante;
4. Il menu di amministrazione è poco intuitivo, la gestione di moduli e contenuti segue una logica completamente diversa rispetto ai due CMS citati prima;
5. Javascript: la versione fornita purtroppo è già vecchiotta, spesso questo gioca dei brutti scherzetti se si utilizzano degli elementi esterni nella pagina, che richiedono versioni più



Figura 16: Un CMS versatile

aggiornate per funzionare al meglio. Inoltre, per imparare al meglio Drupal, ho scelto di farmi le “ossa” sulla versione 6, che è abbastanza in linea con quanto visto su altri CMS, per poi trasportare, e di conseguenza capire più velocemente, la gestione sulla versione successiva.

6. Altrimenti forse ci avrei messo il doppio del tempo...

Per chi ora si chiede quali sono allora i punti a favore e perché la valutazione finale è comunque molto positiva, eccoli:

- L'impostazione del “file system” e della gestione dei moduli comporta una maggiore robustezza, in quanto tutto viene separato per singola funzione;
- La documentazione fornita è ottima e il supporto della comunità è pressoché totale;
- I moduli disponibili sono moltissimi;
- Drupal permette una programmazione di moduli nuovi con una facilità come nessun altro CMS;
- Drupal utilizza innoDB per la scrittura dei dati nel database, con conseguente aumento delle prestazioni dello stesso;

Inoltre Drupal ha un modulo molto potente se usato nel modo corretto, ovvero la tassonomia: sono delle categorie, all'interno delle quali si possono raggruppare infiniti termini, utilizzando dei cosiddetti vocabolari.

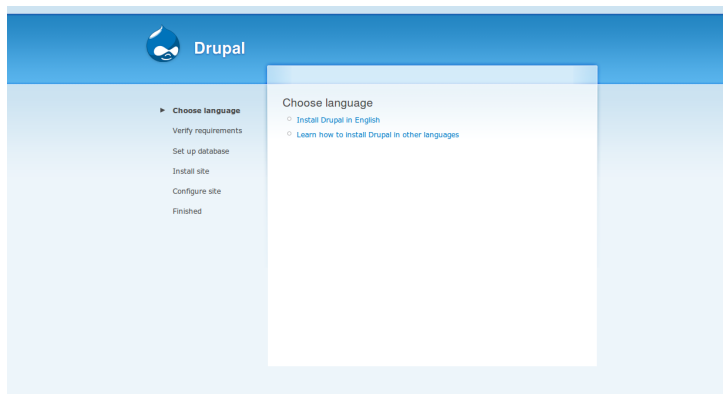


Figura 17: Installazione di Drupal

Il vero vantaggio di questo sistema gerarchico è la possibilità di gestirlo indipendentemente dai contenuti, quindi con la massima libertà di modifica.

Inoltre, il modulo views, un vero e proprio ambiente per la creazione ad hoc di una determinata pagina o di un singolo blocco, può raggruppare o meno alcuni contenuti specifici definiti precedentemente nella tassonomia. Più difficile da spiegare che da vedere, ed è per questo che finisco questo mio breve “excursus” ripetendo quan-

to detto all'inizio.

Drupal è potentissimo e allo stesso tempo flessibile, ma non è di immediata comprensione per l'utente principiante.

Terminale

Gabriele Trombini (mailga@fedoraonline.it)

(seconda parte) - Consolle virtuale e terminale - la base dei sistemi GNU/Linux

utente, root e root

Nel numero scorso abbiamo cominciato ad avvicinarci al terminale, illustrandone le potenzialità e introducendo la differenziazione tra utente "user" ed utente amministratore, cioè "root". Riepilogando, all'apertura del terminale, il simbolo del dollaro (\$), identifica l'utente come user:

```
[gabri@192-168-1-2 ~] $
```

mentre digitando `su` ci si può autenticare come amministratore (previo inserimento della password corretta che abbiamo assegnato a root):

```
[gabri@192-168-1-2 ~] $ su
Password:
[gabri@192-168-1-2 gabri] #
```

Vi è una ulteriore distinzione da specificare *perchè con il su si acquisiscono i poteri di root ma l'ambiente in cui si opera è ancora quello dell'utente.*

Cosa significa tutto ciò? Partiamo dall'esempio visivo, facendo queste tre operazioni distinte da terminale, partendo sempre da user:

- primo esempio:

```
[gabri@192-168-1-2 ~] $ echo $PATH
/usr/lib64/qt-3.3/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/home/gabri/.local/bin:/home/gabri/bin
```

- secondo esempio

```
[gabri@192-168-1-2 ~] $ su
Password:
[root@192-168-1-2 gabri] # echo $PATH
/usr/lib64/qt-3.3/bin:/usr/lib64/ccache:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/home/gabri/.local/bin:/home/gabri/bin
```

- terzo esempio

```
[gabri@192-168-1-2 ~] $ su -
Password:
[root@192-168-1-2 ~] # echo $PATH
/usr/lib64/qt-3.3/bin:/usr/lib64/ccache:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

Vediamo in cosa si differenziano queste tre operazioni:

1. nel primo esempio:

- a il prompt contiene *gabri*;
- b il prompt contiene la tilde *~*;
- c il prompt contiene il simbolo del dollaro *\$*;
- d l'output contiene anche il path ai binari sotto */usr/* e */home/utente/* oltre che a */bin/*;

2. nel secondo esempio:

- a il prompt dopo il login contiene *root*;
- b il prompt dopo il login contiene *gabri*;
- c il prompt dopo il login contiene il simbolo del cancelletto *#*;
- d l'output dopo il login contiene anche il path ai binari sotto */usr/* e */home/utente/* oltre che a */bin/*;

3. nell'ultimo esempio:

- a il prompt dopo il login contiene *root*;
- b il prompt dopo il login contiene la tilde *~*;
- c il prompt dopo il login contiene il simbolo del cancelletto *#*;
- d l'output dopo il login contiene anche il path ai binari sotto */root/*.

Si può vedere chiaramente il passaggio da utente a root tra il *punto 1.a* ed il *punto 1.b*, il passaggio dalla directory di lavoro dell'utente root tra il *punto 2.b* e il *punto 3.b*, il cancelletto che differisce l'utente root dallo user tra il *punto 3.a* ed il *punto 3.b* ed infine la variazione dell'ambiente di lavoro tra il *punto 2.d* ed il *punto 3.d* tra i due diversi login di root.

Risulta quindi evidente che la Shell ci mette in condizione di sapere in ogni momento chi siamo e in che ambiente operiamo.

Si noti, inoltre, che nel passaggio tra root del secondo esempio e root del terzo esempio, non è richiesta la password, in quanto si è già amministratori.

Questo è importante perché è la base dei sistemi Linux, che suddivide in modo preciso e definito ciò che un utente può fare da ciò che non può fare.

L'utente **root** (loggandosi con il *su -*), acquisendo quindi l'ambiente di root, ha in mano il sistema e può disporre di tutti i poteri per l'amministrazione del sistema operativo e, se usato in modo improprio, potrebbe causare danni irreversibili.

Generalmente, per ovvie questioni di sicurezza, all'utente root è inibito l'accesso all'interfaccia grafica e ad alcune GUI ed è bene considerare attentamente di lavorare come user. Abbiamo visto quindi come passare da utente a root, ma in un sistema multiuser è possibile effettuare uno switch anche da un utente all'altro, come spiegato in:

```
[gabri@192-168-1-2 ~] $ man su
```

shell bash? e basta?

Fino ad ora abbiamo parlato solo di shell bash (acronimo di *bourne again shell*) default in Fedora, ma ne esistono diversi altri tipi come, ad esempio:

- shell dash
- shell ksh
- shell tcsh
- shell zsh

tutte presenti nei repository Fedora e ciascuna con una propria peculiarità.

Non staremo qui ad esaminare le potenzialità per ciascuna di esse, però è importante sapere che esistono e sono utilizzabili tanto quanto la bash.

Per avere più informazioni, basta digitare, da terminale:

```
[gabri@192-168-1-2 ~] $ yum info "nome shell"
```

Se verranno installati diversi tipi di shell, sarà possibile recuperarne l'elenco con il comando:

```
[gabri@192-168-1-2 ~] $ cat /etc/shells
```

Per passare da un tipo di shell ad un'altra, ovviamente tra quelle installate, basta digitare il comando al prompt prendendolo dall'output di `/etc/shells`:

```
[gabri@192-168-1-2 ~] $ sh  
sh-4.2
```

non è necessario digitare tutto il percorso `/bin/sh` perchè abbiamo visto che nell'ambiente utente è presente il path a `/bin`, se non ci fosse stato avremmo dovuto scriverlo.

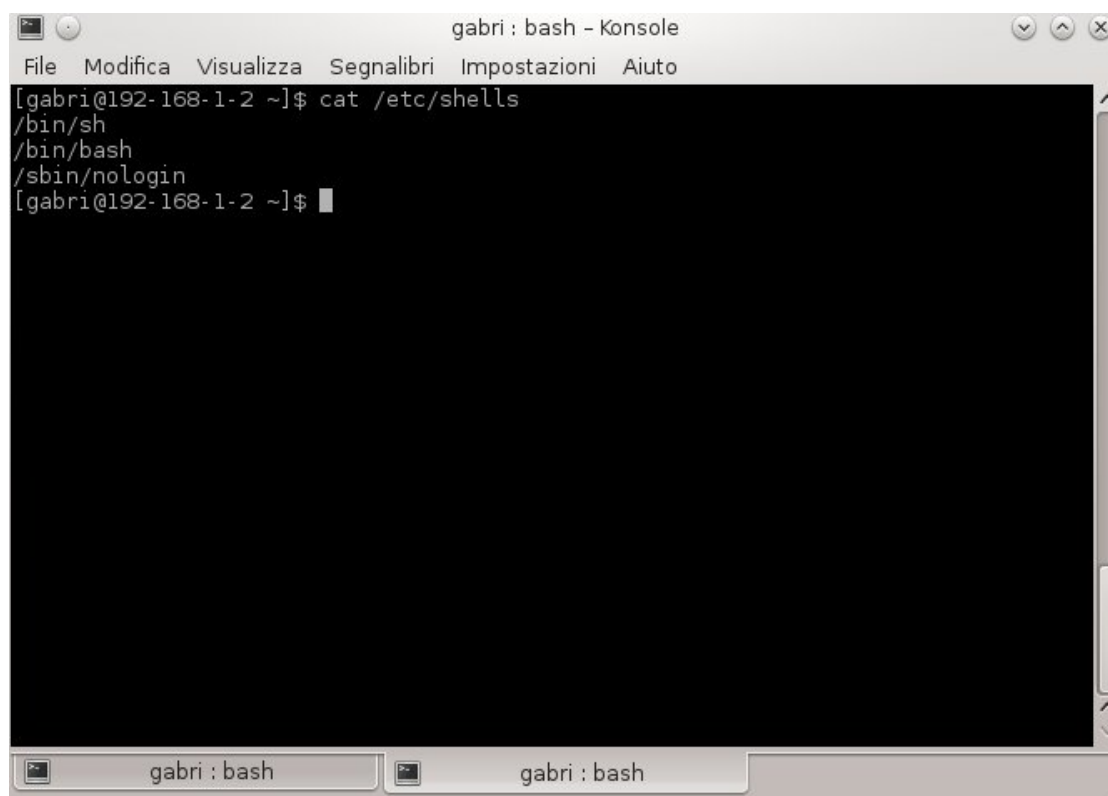


Figura 18: shell in una installazione standard di Fedora

guardare nel filesystem

A questo punto abbiamo le conoscenze di base per cominciare a "girovagare" all'interno del filesystem e capire come eseguire i medesimi compiti dei filemanager grafici.

La shell permette di avere più informazioni dalle directory e dai file proprio per la sua duttilità, essendo anche in grado di combinare più comandi e filtrare gli output.

comando ls

questo comando permette di visualizzare l'elenco dei files e directory; infatti digitando:

```
[gabri@192-168-1-2 ~] $ ls
```

si ottiene la lista dei files e le directory nel punto del filesystem dove ci troviamo, ed i colori ci aiutano a distinguere gli uni dalle altre.

Le opzioni che il comando ls offre sono davvero parecchie, basta dare uno sguardo a:

```
[gabri@192-168-1-2 ~] $ man ls
```

ci limitiamo quindi ad una panoramica delle opzioni più rilevanti e più usate.

Innanzitutto nel filesystem Linux esistono dei file e delle cartelle che vengono nascosti dalla vista corrente, per motivi di convenienza, di configurazione o di sicurezza.

ls permette di elencare quelli visibili; teniamo presente che per nascondere un file in linux basta anteporre al nome un `.` (file = file visibile | `.file` = file nascosto), mentre per poter comprendere nell'elenco anche un file nascosto occorre utilizzare l'opzione `"-a"`:

```
[gabri@192-168-1-2 ~] $ ls -a
```

In entrambi i casi, le informazioni restituite sono un elenco di files e directory che però hanno delle proprie caratteristiche e proprietà, visibili utilizzando l'opzione `"-l"`

```
[gabri@192-168-1-2 ~] $ ls -l
```

per non visualizzare i files/directory nascoste e combinando le due opzioni per vedere anche quelli nascosti:

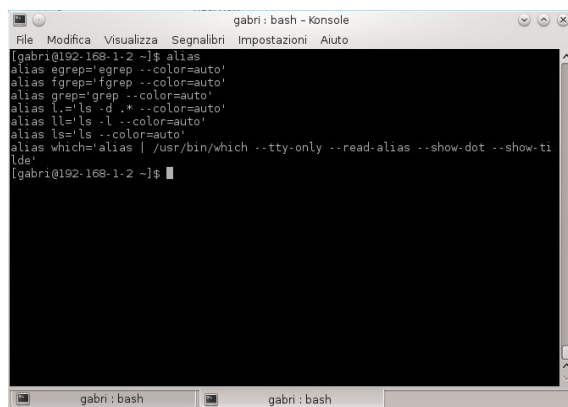
```
[gabri@192-168-1-2 ~] $ ls -al
```

E' possibile anche verificare il contenuto di tutte le directory ricorsivamente a partire da quella in uso aggiungendo l'opzione `"-R"`:

```
[gabri@192-168-1-2 ~] $ ll -aR
```

Ovviamente è possibile ricercare anche se non ci si ricorda interamente il nome del file, utilizzando o il punto interrogativo ("`?`") qualora non ci si ricordasse un carattere del file, oppure il classico asterisco ("`*`") se non ci si ricordasse una parte del nome.

(fine seconda parte)



```
gabri: bash - Konsole
File Modifica Visualizza Segnalibri Impostazioni Aiuto
[gabri@192-168-1-2 ~]$ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -d * --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-ti
lde'
[gabri@192-168-1-2 ~]$
```

in fedora è possibile fare ricorso all'alias preimpostato per il comando `ls -l` che diventa `ll`, utilizzabile con le altre opzioni. Gli alias disponibili sono visibili digitando a terminale `alias`

Servizi di sistema (demoni)

Gabriele Trombini (mailga@fedoraonline.it)

(ultima parte) - I servizi in Fedora

In Fedora è stato ormai completato il passaggio dai servizi **SysVinit** a **Systemd** (una panoramica delle differenziazioni all'indirizzo http://fedoraproject.org/wiki/SysVinit_to_Systemd_Cheatsheet).

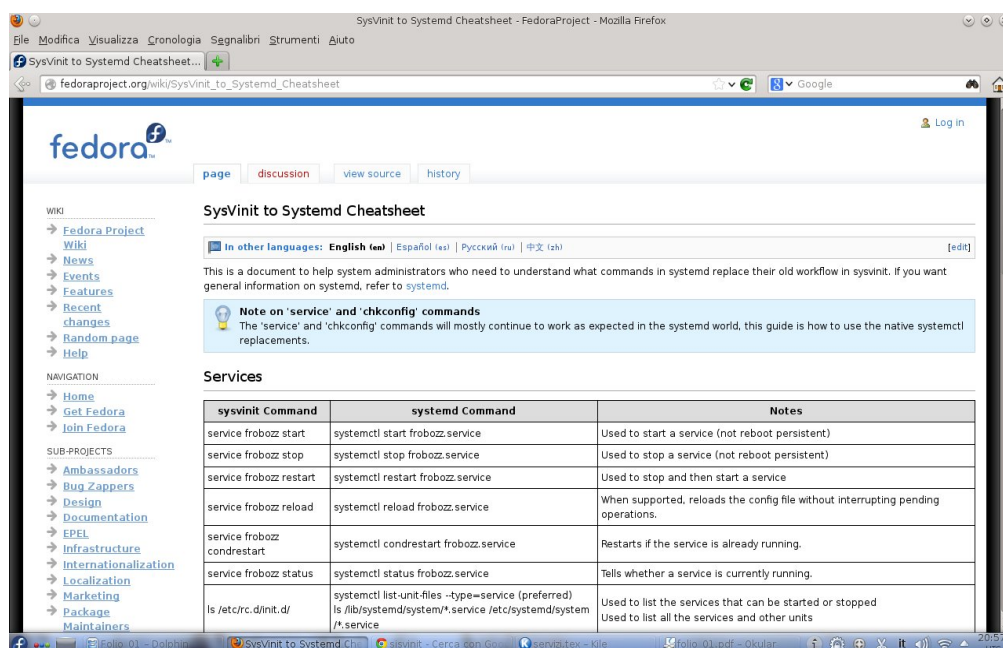


Figura 19: la pagina del Wiki dedicata alla differenza di chiamate tra SysVinit e Systemd

In Systemd possiamo creare principalmente due tipi di chiamate ai servizi:

- **mediante script;**
- **diretto.**

Nel primo caso (**script**) occorrono due tipi di file, uno script eseguibile ed un file di testo.

Lo script è solo da posizionare all'interno della directory `/usr/lib/systemd/scripts/nome`, sarà poi il "nome.servizio" che verrà inserito in `/usr/lib/systemd/system/nome.service`, sul quale ci si sofferma in questo articolo, ad attivarlo.

La sintassi del file `.service` segue delle regole ben precise:

```
[Unit]
Description=mio servizio

[Service]
Type=oneshot
ExecStart=/usr/lib/systemd/scripts/mio_script

[Install]
WantedBy=multi-user.target
```

In esempio è riportato un servizio molto semplice ma è quel che serve per capirne la logica. Come è possibile intuire, il grosso del lavoro viene lasciato fare allo script "*mio_script*" inserito nel path `"/usr/lib/systemd/scripts/".`
 Nel secondo caso (**diretto**), per cose molto semplici, è sufficiente il file "*nome.service*", che non farà riferimento alcuno agli script ma la cui sintassi è la stessa vista nell'esempio precedente:

```
[Unit]
Description=mio servizio

[Service]
Type=oneshot
ExecStart=/bin/sh ll / > /root/servizioavviato.txt

[Install]
WantedBy=multi-user.target
```

Se si volesse quindi scrivere un *file.service* (i file di script seguono altre definizioni e non sono oggetto di analisi in questo articolo) dobbiamo stare attenti alla sintassi e devono cominciare con una sezione, perciò diamo uno sguardo ai campi:

[Unit]

Description= una breve descrizione del servizio e non deve superare gli 80 caratteri;
Documentation= man:nome.service(8) man:nome.conf(5) <http://www.nome.org/docs>;
 questo tag è supportato a partire da fedora 17 e consente di accedere alla documentazione, si può usare `man:manpage info:infofile` qualora essa fosse compresa nel pacchetto.
 Se il file è in solo testo si può usare il tag `file://path/to/file` ed, infine, se avessimo un sito web di riferimento la sintassi è `http://www.nome.org/docs`.
 Possono essere inseriti diversi campi, separati da spazi.

[Service]

Type= questo tag identifica la tipologia del servizio, per i servizi D-Bus in questa riga dovrebbe essere inserito "dbus", per servizi tradizionali si può usare "forking", per servizi che non ne hanno altri collegati o a supporto "simple" e per riferirsi a servizi di script semplici "oneshot" (vedere comunque <http://Opointer.de/public/systemd-man/systemd.service.html>).
 Nel caso il servizio sia di tipo "simple", essendo il default è possibile ometterlo.
BusName= Se il tipo (il Type dell'opzione di cui sopra) è "dbus" è obbligatorio metterlo, ma si può omettere se non prende il nome da un bus D-Bus.
ExecStart= è sempre obbligatorio essendo l'unica opzione indispensabile; questo tag richiede la stringa con la quale si attiva il demone con tutte le opzioni necessarie.
ExecReload= dovrebbe essere specificato per tutti quei servizi che supportano il ricaricamento.
 Se non è supportato si può omettere.
 Si raccomanda di inserire codice che ricarica la configurazione in modo sincrono.

[Install]

WantedBy= qui occorre inserire quelli che in SysVinit si chiamavano runlevel (in systemd, cambiando la logica, vengono istituiti i .target) e si raccomanda di inserire **graphical.target** (servizi relativi alla GUI) o **multi-user.target** (per tutto il resto).
 Questo servizio verrà avviato a seconda del tipo di avvio.
 Per informazioni più approfondite, riferirsi alla pagina <http://fedoraproject.org/wiki/Packaging:Systemd> del Fedora Project ed alla pagina **man** di `systemd.service`.



Per caricare una variabile di sistema nella sezione **[service]**, occorre inserire l'opzione **EnvironmentFile=** richiamando quindi il file che contiene le variabili che si intendono utilizzare ed alle quali ci si può riferire utilizzando la sintassi `$variabile` (per una parola) e `!$variabile` (per il valore della variabile separata da spazi) inserendola poi come opzione nel comando di ExecStart.

Per evitare che ExecStart generi un errore qualora non trovasse la variabile, occorrerà inserire un `"-"` (trattino) nella linea EnvironmentFile=.

```
[Service]
Type=forking
EnvironmentFile=-/etc/sysconfig/httpd
ExecStart=/usr/sbin/httpd $OPTIONS
ExecReload=/usr/sbin/httpd $OPTIONS -k restart
```



Attenzione:

Qualsiasi file `.service` in `/etc/systemd/system` sovrascrive quelli in `/lib/systemd/system` per cui prestare attenzione qualora si modifichi un file (meglio spostarlo in `/etc/`)

Per quanto riguarda Fedora, **non è consentito usare richieste di dipendenze nella sezione [Unit] con i tag Requires e/o Wants, bensì occorre ordinare le dipendenze utilizzando Before= e After=.**

Ciò consente a Systemd di ordinare l'avvio qualora due servizi siano richiesti contemporaneamente ma che non necessariamente avvia uno se trova già avviato l'altro.

Wants e Requires sono usati solo nel caso di una dipendenza obbligatoria, comunque correlati all'ordinamento.

Systemd consente tre forme di attivazione:

- **hardware** solo quando un servizio è installato ma si avvia solo se un certo tipo di hardware è installato, generalmente udev si occupa di attivare il servizio;
- **socket** si attiva solo quando un servizio consente a systemd di rimanere in ascolto di connessioni e, quando systemd riceve una connessione da un socket il servizio viene avviato. E' necessario un `.socket` file in `/lib/systemd/system` che comunichi a systemd di rimanere in ascolto su un dato socket e cosa fare quando riceve una connessione. Fedora non utilizza questa feature;
- **dbus** per consentire lo startup di un servizio e dell'utenza D-bus. Per questa operazione è essenziale D-bus sia attivato e che le richieste ad esso siano inviate a systemd, cosicché ci si possa riferire ad una singola istanza del servizio anche qualora venisse attivato sia dall'avvio che, successivamente, dall'utilizzo del sistema operativo.

Come è stato già visto nel numero precedente, per fare in modo che venga avviato al boot (**In Fedora 18 è possibile omettere il .service**):

```
# systemctl enable nome.service
```

E' inoltre possibile passare delle istanze systemd anche al kernel al momento del boot, eccone alcune:

- **systemd.unit=** per avviare temporaneamente il sistema secondo il valore impostato (desumibile da `man systemd.special`);
- **rd.systemd.unit=** per avviare temporaneamente la ramdisk iniziale secondo il valore impostato (desumibile da `man systemd.special`);
- **systemd.show_status=** valore booleano, se vero mostra un breve aggiornamento dello stato dei servizi mentre il sistema parte. Se tra le opzioni del kernel c'è `quite`, viene impostato automaticamente a falso;
- **systemd.setenv=** accetta una stringa sotto forma di `VARIABILE=VALORE`. Può essere usata (anche più volte) per impostare variabili di ambiente per il processo di inizializzazione e di tutti i processi figli già al boot. Se esse non sono rilevabili, azzera la variabile di default che verrebbe impostata dalla ramdisk iniziale del sistema;

- **quiet** se inserito come opzione al kernel spegne tutti gli output al boot. Il kernel ha già attiva questa opzione perchè ha il log disabilitato, passando questa opzione si disabilita sia l'output del kernel che del sistema;
- **emergency** effettua il boot in modalità emergenza;
- **single, s, S, 1** effettua il boot in modalità ripristino;
- **2, 3, 4, 5** effettua il boot secondo il vecchio sistema SysV dei runlevel;
- **locale.LANG=, locale.LANGUAGE=, locale.LC_CTYPE=, locale.LC_NUMERIC=, locale.LC_TIME=, locale.LC_COLLATE=, locale.LC_MONETARY=, locale.LC_MESSAGES=, locale.LC_PAPER=, locale.LC_NAME=, locale.LC_ADDRESS=, locale.LC_TELEPHONE=, locale.LC_MEASUREMENT=, locale.LC_IDENTIFICATION=** imposta la localizzazione del sistema, vedere man locale.conf(5) and locale(7).

Per informazioni più complete, da terminale digitare **man kernel-command-line**.

A questo punto, avendo approfondito la conoscenza di systemd, ancora ben lontana da essere completa, è possibile capire anche la logica di lavoro qualora si volesse cambiare il target di avvio del sistema (quelli che in SysVinit erano definiti "runlevel").

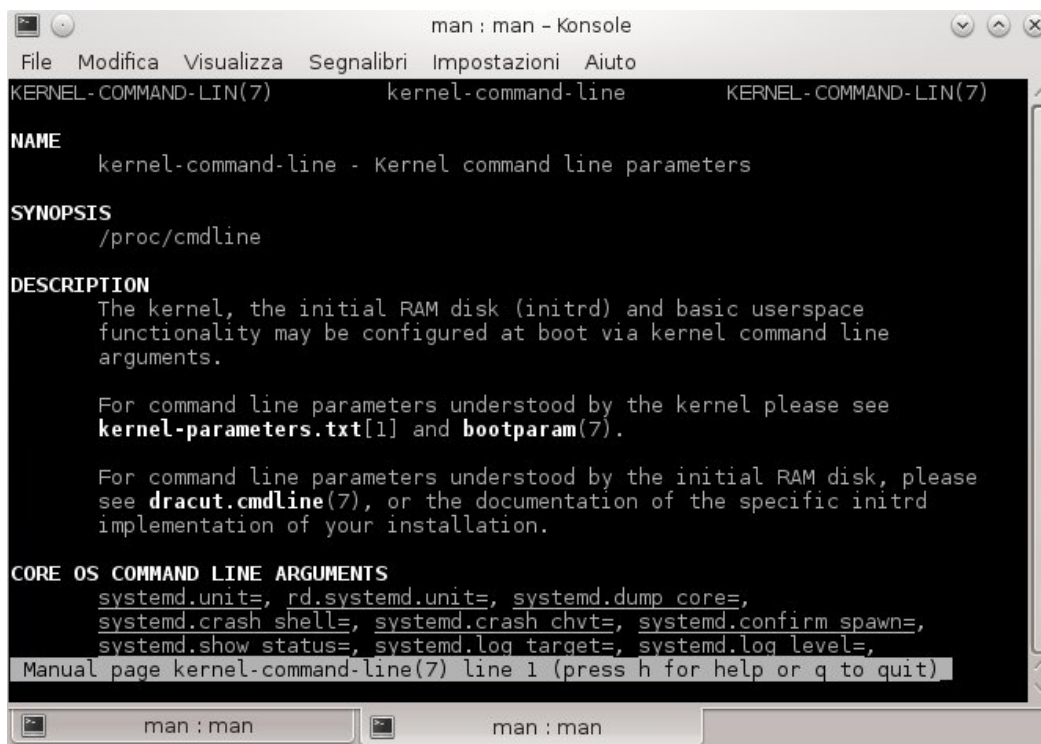


Figura 20: La pagine man (nell'immagine quella di kernel-command-line) sono una fonte di notizie inesauribili

```
$ ln -sf /usr/lib/systemd/system/multi-user.target /etc/systemd/system/default.target
```

Questo comando permette di stabilire di default l'avvio con il **multi user target** (cioè il sistema senza l'interfaccia grafica, quello che era il runlevel 3).

```
$ ln -sf /usr/lib/systemd/system/graphical.target /etc/systemd/system/default.target
```

Viceversa questo comando permette di stabilire di default l'avvio con il **grafical target** (cioè il sistema con l'interfaccia grafica, quello che era il runlevel 5).

E' necessario cancellare i link simbolici prima di cambiare il runlevel predefinito.

Creazione di RPM

Gabriele Trombini (mailga@fedoraonline.it)

(seconda parte) - costruzione di un RPM

Abbiamo visto, quindi, che l'albero della *rpmbuild* creata con il comando:

```
$ rpmddev--setuptree
```

è costituito dalle directory:

- BUILD
- RPMS
- SOURCES
- SPECS
- RPMS

In questo capitolo cominceremo quindi a popolare queste directory, cominciando dalla **SPECS** e dalla **SOURCES**.

Esse sono, infatti, quelle dove l'intervento dell'utente sarà fondamentale. Per poter costruire il proprio pacchetto RPM occorre assolutamente cominciare dai sorgenti, che normalmente troviamo in formato compresso *.tar.gz* o *.tar.gz2*, perciò il primo passo è quello di scaricarli e posizionarli all'interno della directory SOURCES.

Allo stesso modo dovremo inserire qui sia le eventuali patch che vorremo includere sia, nel caso in cui avessimo una icona da associare al programma, il file di icona in forma xpm o gif (chiaramente quest'ultimo ha una sua utilità solo nel caso di installer grafici e di software prodotti per un uso all'interno del sistema grafico).

Avendo, quindi, preparato il necessario per poter procedere alla creazione del pacchetto, andremo a creare il file **.spec** che costituisce il punto focale del sistema RPM.

il file .spec

Innanzitutto le informazioni relative al file si possono trovare a questi indirizzi:

- http://fedoraproject.org/wiki/How_to_create_an_RPM_package/it#Panoramica_SPEC_file;
- http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/ch-specfiles.html;
- <http://www.rpm.org/max-rpm/s1-rpm-build-creating-spec-file.html>.

In esso sono contenute tutte le direttive che un file rpm deve possedere per essere sia funzionale al sistema che conforme alle specifiche del **Fedora Project** ed è da questo file che il compilatore preleva le istruzioni per la costruzione.

Per avere un punto di partenza già valido per un semplice pacchetto, agire da terminale:

```
$ cd /rpmbuild/SPECS
$ rpmddev--newspec prova
prova.spec created; type minimal, rpm version >= 4.10.
```

Andando quindi a vedere nella directory, troviamo il file spec minimale, creato da modello:

```
$ ls
prova.spec
```

che, dando una occhiata al suo interno, risulta essere:

```

$ cat prova.spec
Name: prova
Version:
Release: 1%{?dist}
Summary:

License:
URL:
Source0:

BuildRequires:
Requires:

%description

%prep
%setup -q

%build
%configure
make %{?_smp_mflags}

%install
rm -rf $RPM_BUILD_ROOT
%make_install

%files
%doc

%changelog

```

Quella appena vista è una configurazione di base che deve essere completata secondo una sintassi ben definita ed è costituito da diverse sezioni, tra le quali:

- preamble;
- prep;
- build;
- check;
- install;
- script;
- verify;
- file;
- changelog.

Tip:

tutte le sezioni comandi iniziano con un simbolo di percentuale (%).
Se il compilatore dovesse quindi trovare nei commenti il simbolo percentuale con la sezione (ad esempio %prep), restituirà un errore.
Per evitarlo è sufficiente inserire un altro simbolo % davanti al precedente, così:

%%prep

Pur essendocene altre utili a seconda dei casi e delle necessità, queste saranno quelle che esamineremo.

In tutto il file è possibile inserire commenti esplicativi per la lettura e comprensione del file e dei passaggi in esso contenuti che non devono entrare nel processo di compilazione; per farlo occorre mettere un simbolo del cancelletto (#) all'inizio di ogni riga.

Introduciamo anche il concetto di *“macro”* (che tratteremo in un capitolo a parte) perchè ne vedremo alcune nel corso della stesura del file .spec.

Una macro definisce un valore statico a cui fare riferimento e ciascuna di esse (che sia impostata dal sistema o dall'utente), offre un vantaggio in termini di rinvio a tale valore all'interno di specfile complessi. E' di uso comune dichiarare quelle personali (non di sistema) in testa al file usando la dichiarazione **“%define”** ma non è una necessità legata alle specifiche; infatti ciò può avvenire ovunque nel documento.

Come detto in precedenza, nella gerarchia del filesystem della buildroot il file `.spec` deve essere inserito all'interno della directory SPECS.

preamble

Questa sezione è predisposta per contenere tutte quelle informazioni che poi verranno estratte dalle query di ricerca e informazioni sul pacchetto (si può trovare un indice sintetico all'interno del sito rpm.org <http://www.rpm.org/max-rpm/s1-rpm-specref-preamble.html>).

La specifica **NVR (Name – Version – Release)** permetterà la comparazione tra nomi, versioni e release, perciò è bene che chi si accinge a compilare un pacchetto Fedora tenga ben presente questa logica. Analizziamo perciò la sintassi e le voci principali che fanno parte del preambolo.

Name: "nome-del-pacchetto"

Questo sarà il nome del pacchetto che verrà utilizzato nell'etichetta e nel nomefile ed in base ad esso si effettuerà la comparazione tra nomi.

Il nome del pacchetto non deve contenere spazi (es. Name: mio-pacchetto).

Version: "#.#.#"

In questa linea si inserirà, ovviamente, la versione del software ed anch'essa verrà inclusa nell'etichetta e nel nomefile.

Si deve utilizzare una sintassi che prevede una numerazione e, invece del trattino usato per il nome, dei punti per le separazioni (es. Version: 1.1.2).

Anche in questo caso non sono ammessi spazi e verranno filtrati per il controllo delle versioni.

Release: "1"

La release equivale al numero di volte in cui il software è stato pacchettizzato, praticamente è un contatore delle modifiche fatte al file `.spec` (es. Release: 1)

Come per le precedenti definizioni, anche questa numerazione viene inserita nell'etichetta e nel filename e verrà considerata per le ricerche sul pacchetto.

License: "tipo di licenza"

Questa direttiva è usata per le informazioni di licenza (es. License: GPLv2+ and GFDL), che va a sostituire la voce copyright e che per Fedora devono seguire quanto riportato alla pagina wiki <https://fedoraproject.org/wiki/Packaging:License>

Group: "Gruppo/Sezione"

Inserendo il gruppo e relativa sezione separate da uno slash in questa informazione (es. Applications/Editors) permetterà di trovare il proprio pacchetto tra i gruppi disponibili in Fedora qualora vengano fatte delle ricerche per tipologia di applicazione.

L'elenco è possibile trovarlo in sono elencati in `/usr/share/doc/rpm-X.X/GROUPS` (dove x.x è la versione presente nel proprio sistema dell'utility "rpm").

Anche se questo tag è presente nel modello, **rpm in Fedora 18 non considera questa sezione.**

Source: "url"

Questa particolare informazione può essere ripetuta per ciascun sorgente presente nella directory SOURCE, numerando sequenzialmente ciascun input, partendo da 0:

Source0: "http://www.url/al/file1.tar.gz"
Source1: "http://www.url/al/file2.tar.gz"

Essa ha due utilità; fornire le informazioni su dove è contenuto il sorgente e capire quale è il nome che si può trovare nella directory SOURCE, che *deve* corrispondere.

Da questa riga, rpm preleverà il solo nome del sorgente, perciò pur scrivendo quel che si vuole nel corpo del testo, questa sarà l'unica informazione che verrà considerata.

```

Scrivania : bash - Konsole
File Modifica Visualizza Segnalibri Impostazioni Aiuto
%define glib2_version 2.28.0
%define pygo_version 2.90
%define desktop_file_utils_version 0.9
%define gtksourceview_version 3.0.0
%define enchant_version 1.2.0
%define isocodes_version 0.35
%define libpeas_version 0.7.3
%define zeitgeist_version 0.3.2

Summary:      Text editor for the GNOME desktop
Name:         gedit
Epoch:       2
Version:      3.6.2
Release:      1%{?dist}
License:      GPLv2+ and GFDL
Group:        Applications/Editors
#VCS: git:git://git.gnome.org/gedit
Source0:      http://download.gnome.org/sources/gedit/3.6/gedit-%{version}.tar
.xz

URL:          http://projects.gnome.org/gedit/

Requires(post):      desktop-file-utils >= %{desktop_file_utils_version}
Requires(postun):    desktop-file-utils >= %{desktop_file_utils_version}
:

```

Figura 21: L'header (preambolo) dello specfile di gedit

BuidRequires: "nome-pacchetto o file"

In questo caso, si forniscono informazioni a riguardo dei requisiti indispensabili alla compilazione del pacchetto e si può inserire sia una lista di pacchetti che di files in formato query di richiesta, cioè senza versione nè estensione. Può essere ripetuta per ciascun file/pacchetto richiesto, senza una numerazione:

BuidRequires: "nome-pacchetto o file 1"
BuidRequires: "nome-pacchetto o file 2"

Requires: *"nome-pacchetto o file"*

A differenza del tag precedente, questa direttiva non indica il pacchetto/file o i pacchetti/files necessari alla compilazione, bensì i pacchetti/files necessari al funzionamento del programma installato. Può essere ripetuta, senza una numerazione:

Requires: *"nome-pacchetto o file 1"*
Requires: *"nome-pacchetto o file 2"*

Rpm è, inoltre, in grado di rilevare alcune dipendenze richieste, in alcuni casi, infatti, non può, perciò è sempre bene elencarne la lista.

Patch: *"nome-file.patch"*

Questa particolare informazione può essere ripetuta per ciascuna patch presente nella directory SOURCE, numerando sequenzialmente ciascun input, partendo da 0:

Patch0: *"nome-file-1.patch"*
Patch1: *"nome-file-2.patch"*

BuildRoot: *"path/di/build"*

Utile per poter definire una directory di build differente per testare la compilazione ed installazione. Anche se questo tag è presente nel modello, **rpm in Fedora 18 non considera questa sezione.**

URL: *"url"*

A differenza dell'url indicato sopra, qui viene indicata l'url per la documentazione del pacchetto (es. <http://projects.gnome.org/gedit/>).

Distribution: *"nome-distribuzione"*

Specifica per quale distribuzione è stato eseguito il package (es. Distribution: Fedora); volendo è possibile omettere la voce.

Icon: *"nome_icona.ext"*

Definisce il nome dell'icona assegnata al pacchetto, deve essere XPM o GIF (.xpm o .gif) ed inserita, come detto, nella directory SOURCES.

Summary: *"testo"*

La direttiva summary non è altro che una breve descrizione del pacchetto che deve stare su una riga (non più di 50 caratteri).

Excludearch: "archX, archX"

Esplicita le architetture escluse.

Exclusivearch: "archX, archX"

Al contrario del precedente identifica le architetture comprese nella compilazione.

BuildArch: "archX, archX"

Sezione da aggiungere invece di Excludearch e Exclusivearch per definire la piattaforma sulla quale è possibile installare il pacchetto (es. BuildArch: noarch).

Excludeos: "nomeos"

Esclude un sistema operativo (es. Excludeos: windows).

Exclusiveos: "nomeos"

Tag per definire un sistema operativo specificato (es. Exclusiveos: linux).

Sempre nel preambolo si inseriscono altre informazioni un po' più approfondite, ad esempio:

% Description: "testo"

La direttiva description inizia con un segno di percentuale e, a differenza del sommario, sta su più di una riga (ciascuna deve essere al massimo di 79 caratteri) e permette di descrivere molto particolarmente il pacchetto.

Occorre prestare però attenzione alla formattazione perchè diversi tools che lavorano sui files rpm la interpretano diversamente; i salti di interlinea sono visti come paragrafi diversi e iniziando la descrizione con spazi bianchi il testo verrà considerato pre-formatto.



Attenzione:

non tutti tag appartenenti all'header sono obbligatori, non tutti sono conformi alle specifiche del Fedora Project e non tutti vengono considerati da rpm.

(fine seconda parte)

Kernel

Giuseppe Delvecchio (virus@fedoraonline.it)

(Ultima parte) Kernel - approfondimento del numero precedente

Proviamo ad approfondire alcuni concetti base del kernel linux:

1. multitasking;
2. gestione memoria;
3. kernel space / user space;
4. multiuser;
5. tutto è un file.

Multitasking

Quante volte avete ascoltato un brano musicale mentre leggevate una pagina web sulla vostra macchina?

L'impressione che la macchina esegua contemporaneamente entrambe le operazioni è una illusione, essa è dovuta alla capacità del sistema di essere estremamente veloce nell'eseguire le operazioni in successione, in modo tale da non manifestare interruzioni.

Il kernel linux è nato come sistema multitasking (task = funzione/compito) cioè il kernel esegue più compiti (processi, programmi) nello stesso tempo, distribuendo opportunamente le risorse ed eseguendo ora un processo ora l'altro.

*Il componente kernel che si occupa di eseguire il cambio di esecuzione è lo **scheduler** (schedule = pianificare).*

Le politiche per il cambio di esecuzione devono essere efficienti, efficaci, e devono tener conto che alcuni processi potrebbero rimanere in attesa di dati provenienti da dispositivi più lenti. Da qui la necessità di uno scheduler che ottimizzi al meglio le risorse disponibili.

Esistono vari metodi di multitasking, utilizzati in base al comportamento che si vuole ottenere dal sistema. In linux, come in tutti i sistemi più moderni, è stata scelta la modalità preemptive (prelazione), ovvero la possibilità da parte del kernel di interrompere l'esecuzione di un processo per liberare risorse a prescindere dalla volontà del processo stesso.

Gestione memoria

La gestione della memoria in un elaboratore è una delle operazioni più delicate che vengono eseguite dal kernel, ogni processo deve avere a disposizione memoria per le sue istruzioni e per i suoi dati.

I livelli di memoria in un elaboratore, in ordine di velocità, sono grosso modo:

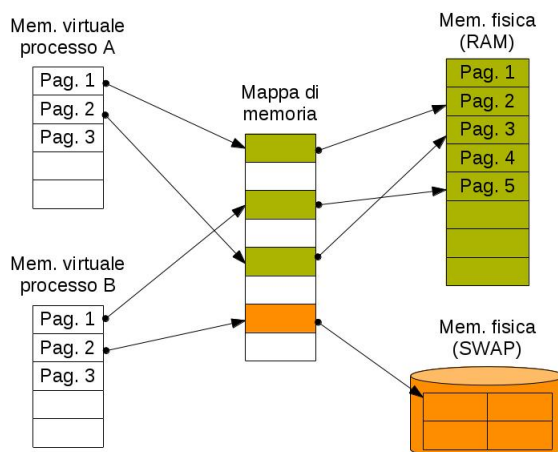
- a = registri del processore, memoria in quantità ridottissima ma ad altissima velocità, su di essi vengono eseguite le istruzioni elementari;
- b = cache L1, più ampia, ad alta velocità ma di alto costo, che conserva i dati/istruzioni che presumibilmente saranno processati a breve, la sua presenza riduce il tempo di latenza della comunicazione tra la cpu e la ram, tra esse a cascata ci possono essere altre cache, L2 / L3 di capacità più ampie ma con velocità leggermente inferiori. Le cache sono "innestate" direttamente sul processore, il loro ordine di grandezza attuale è il Megabyte;
- c = ram, random access memory, di media capacità, velocità ridotta rispetto alla cache di costo più contenuto, l'ordine di grandezza attuale è il Gigabyte. Le memorie fin qui descritte sono volatili, cioè non sono in grado di mantenere il loro contenuto in assenza di energia elettrica;
- d = memorie di massa, quali dischi rigidi o dispositivi analoghi di elevata capacità e costo relativamente basso ma con velocità di comunicazione ancora più ridotta, il loro ordine di grandezza attuale è il Terabyte.

Dato che le istruzioni e i dati di un processo vengono caricati dai dischi fino ai registri della cpu, si comprende come deve essere efficiente e sicuro il trasferimento dei dati e la gestione dello spazio disponibile.

Ci sono vari modi di gestione ed organizzazione della memoria, quello che viene utilizzato usualmente nel kernel linux è quello della memoria virtuale.

Ad ogni processo viene assegnata una memoria con indirizzi progressivi fittizi, che non corrispondono alla reale posizione dei dati nella ram.

Il kernel divide la memoria virtuale di ciascun processo in pagine di dimensione fissa, ogni pagina virtuale è associata ad una pagina della memoria reale, questa associazione è contenuta in una mappa detta page table. Uno schema aiuterà a comprendere meglio questo problema:



Naturalmente una stessa pagina di memoria reale può essere "puntata" da molte pagine di memoria virtuale, questo è, per esempio, il caso delle librerie condivise.

Poiché la capacità della memoria fisica è inferiore alla memoria virtuale, il kernel provvede a mantenere nella ram solo le pagine che servono ai processi in esecuzione.

Quando la ram non è sufficiente il kernel provvede a trasferire le pagine meno utilizzate nell'area di swap appositamente creata su di un dispositivo di massa, operazione che ovviamente rallenta l'esecuzione dei processi vista la bassa velocità di questi dispositivi.

Quando un processo tenta, erroneamente, di accedere ad una pagina non più esistente avviene una cosa particolare: il kernel invia al processo un segnale che ne causa l'interruzione, il temibile signal **11 - SIGSEGV**.

Vi chiederete del perchè di questa organizzazione complessa: gestendo la memoria in questo modo il multitasking è più efficiente.

Resta comunque il fatto che il kernel linux esegue queste operazioni con grande rapidità ed in maniera totalmente trasparente ai processi.

Da quanto detto si comprende che il kernel tende ad occupare quanta più ram disponibile per essere più efficiente e rendere il sistema più reattivo possibile.

Può comunque capitare che la ram non risulti sufficiente per far fronte alle esigenze di tutti i processi attivi.

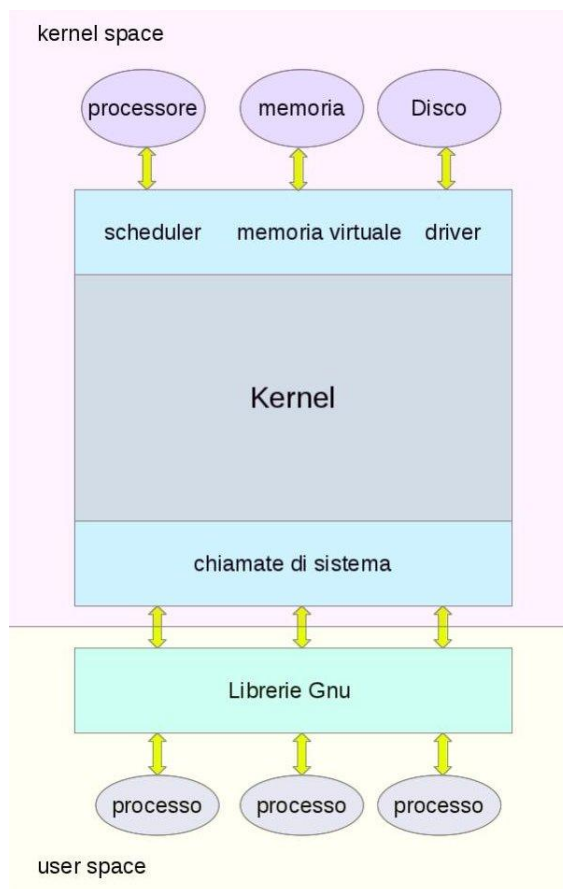
Questo può accadere per due ragioni principali:

- * un memory leak, ovvero una occupazione abnorme (e solitamente erronea) da parte di qualche processo;
- * inadeguatezza delle risorse a disposizione rispetto alle esigenze del software utilizzato.

La conseguenza di questo fenomeno è un uso eccessivo dell'area di swap, col disco che lavora in modo frenetico ed il sistema fortemente rallentato o addirittura bloccato.

Kernel space / user space

A questo punto è necessario uno schema che riassume la situazione:



Il settore in colore rosa delimita lo spazio kernel, cioè tutte le operazioni e i dati necessari al funzionamento del kernel, il settore in colore giallo delimita lo spazio utente cioè tutto ciò che non è kernel.

I due ambienti sono separati e non sono possibili "commistioni" o interferenze.

Il kernel è l'unico processo che accede direttamente alle risorse hardware, gli altri processi girano in un ambiente virtuale e possono accedere alle risorse solo tramite la mediazione ed il coordinamento del kernel.

La comunicazione tra i due spazi kernel/utente avviene mediante due interfacce:

- * Le chiamate di sistema (Syscall) dal lato kernel,
- * Le librerie di funzioni (glibc del sistema GNU) dal lato user.

Fate attenzione che quando si dice user/utente di sistema, non necessariamente, si intende un umano, con i suoi problemi e le sue necessità, ma un qualunque processo non kernel.

Facciamo un esempio:

Un processo, dal nome "cpio", lanciato da un servizio, vuole copiare un file da un disco ad un altro.

"Cpio" è in user space e "corre" ignaro verso le istruzioni di copia file, non appena ci arriva, la corrispondente libreria glibc fa la chiamata di sistema "syscopio" che indica al kernel le operazioni da eseguire.

Le operazioni vengono prese in carico dal kernel, il processo "cpio" non sa cosa stia succedendo, non deve saperlo, il kernel esegue queste indicazioni senza conoscerne lo scopo, non sa che quello che sta facendo è una copia.

Una volta eseguita l'operazione il kernel e la corrispondente "syscopio" inviano un segnale di "tutto fatto" tramite la libreria di funzione a "cpio", il quale, soddisfatto, termina il suo compito, segnalando al servizio che lo ha lanciato che tutto è andato bene.

In effetti il kernel linux, da solo, in una macchina è insufficiente, non possiede infatti gli strumenti di amministrazione del sistema ed i programmi di utilità, strumenti che vengono forniti dal progetto GNU e dalle sue librerie di funzioni.

È questo il motivo per cui è più corretto parlare di sistema GNU/Linux.

L'uno senza l'altro sarebbero inutili.



Figura 22: Logo del progetto GNU

Multuser

Il sistema Linux è multiutente, ovvero è in grado di far utilizzare la macchina a più utenti contemporaneamente.

Per questo fatto sono previsti meccanismi di permessi e di protezioni che impediscono ad utenti diversi e ai loro processi di interferire e/o danneggiarsi tra loro.

Esistono vari livelli di protezione che pongono ben precisi limiti di operatività agli utenti.

Esiste un unico utente che non ha limiti e per cui le protezioni sono disattivate: root.

È importante quindi un uso attento e consapevole dell'utente root: abusare dei suoi poteri può causare danni irreversibili al sistema.

Tutto è un file

In un sistema Linux tutto è considerato un file: *"everything is a file"*, ovvero i dati, gli input/output da dispositivo qualunque esso sia, con l'esclusione delle interfacce di rete, è organizzato come un file.

I file gestiti dal kernel sono di sette tipi:

- a = regular file, sono i file che contengono dati, quelli che vengono comunemente intesi come file. Il kernel non ha alcun supporto per la tipizzazione dei file, né per le estensioni, infatti si fa uso del cosiddetto magic number per distinguerli, volendo essere più chiari, il kernel non distingue un file contenente dati immagine da un file contenente testo;
- b = directory, file che contiene puntatori ai file che devono apparire contenuti in esso;
- c = symbolic link, file che contiene il riferimento al percorso di un file;
- d = char device, file che identifica un dispositivo a caratteri, ovvero un file che è in grado di comunicare byte per byte con un determinato dispositivo, ad esempio una porta parallela;
- e = block device, dispositivo a blocchi, tipo di file che comunica per blocchi predeterminati di byte, un tipico utilizzo è la comunicazione con un disco rigido;
- f = fifo, file di comunicazione software unidirezionale, usati ad esempio per "compensare" flussi di dati attraverso i buffer (memorie di transito);
- g = socket, file di comunicazione software bidirezionale tra processi per la condivisione dati.

L'insieme di tutti i file presenti nel sistema è organizzato in un'unica struttura gerarchica che si ramifica da una directory principale, detta root ("radice" o "/"). È possibile visualizzare il primo livello di tale struttura tramite il comando:

```
$ tree -L 1 /
```

"il tutto è un file" è organizzato dal kernel in maniera tale che un processo utente utilizza sempre la stessa interfaccia qualunque sia il tipo di file, uno strato software intermedio del kernel detto *"Virtual File System"* gestirà poi il tipo di filesystem da utilizzare e le operazioni da compiere.

Riprendo l'esempio precedente.

Un processo, dal nome "copio", lanciato da un servizio, vuole copiare un file da un disco ad un altro.

Quando il processo "copio" incontra le istruzioni di lettura del file dal primo disco e le istruzioni di scrittura sul secondo disco non si preoccupa minimamente del tipo di filesystem di entrambi.

Sarà il kernel e il "Virtual File System" ad occuparsi di rendere i dati coerenti con filesystem utilizzato.

Wake On Lan

Giuseppe Delvecchio (virus@fedoraonline.it)

Vediamo come accedere legalmente ad un computer in remoto

Vi è mai capitato di aver dimenticato un documento importante ed indispensabile in ufficio e che non potete accedere dalla rete esterna al vostro bel server perchè il vostro computer è spento? Il comando wol è quello che vi occorre.

Wol è l'acronimo di *Wake On Lan*, ovvero *risveglio da remoto di un computer spento*.

Materiale occorrente:

- computer rigorosamente spento ma collegato sia alla rete elettrica che network;
- router acceso collegato via ethernet al computer spento;
- conoscenza dell'ip pubblico del vostro router o nome dell'host/dominio;
- se la vostra rete ha un ip pubblico dinamico dovete attivare un servizio del tipo DynDNS per bypassare questo problema.

Sul computer da risvegliare:

Per prima cosa occorre fare partire il computer spento ed accedere al BIOS, dovete attivare l'opzione di wake on lan sulla vostra scheda di rete, di solito troverete questa opzione nel settore *Power* alla voce *PCI Power ON* oppure *Power On : Lan*, comunque qualcosa del genere.

Quasi tutte le schede madri per desktop hanno questa opzione attivabile, nei portatili di solito non è presente.

Fate partire il vostro sistema Fedora 17 e date il comando:

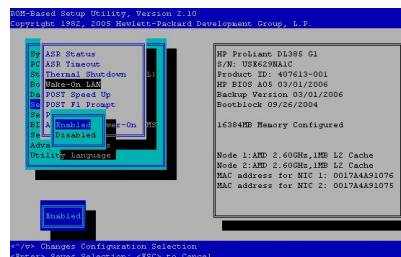


Figura 23: funzione WOL nel BIOS HP

```
# ethtool eth0
```

se eth0 è l'interfaccia di rete, se non conoscete il suo nome date:

```
# ifconfig
```

ed individuate la scheda attiva, controllate l'output:

```
Supports Wake-on: pumbg Wake-on: d
```

in Supports Wake-on ci deve essere la lettera *g*, mentre in Wake-on la *d* il cui significato è: disabilitato; ora attivatelo:

```
# ethtool -s eth0 wol g
```

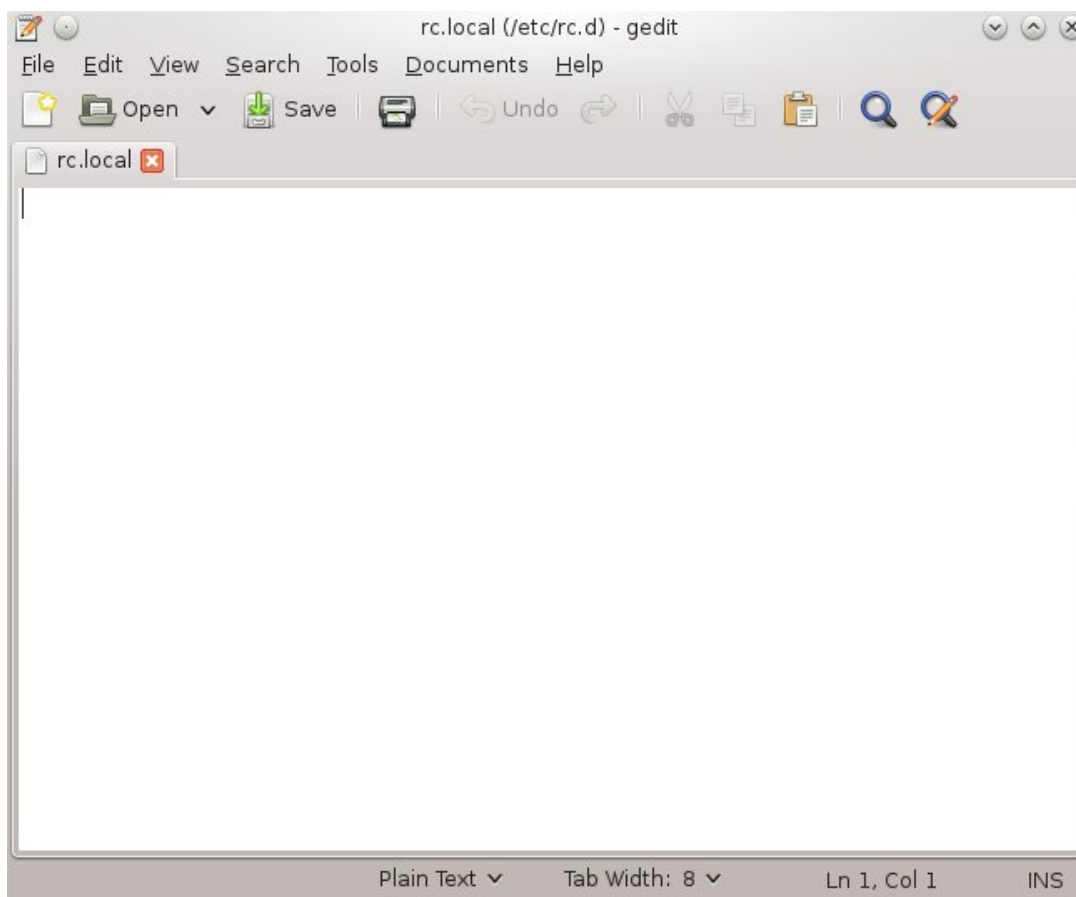
ricontrollate `ethtool eth0` e verificate che Wake-on riporti la lettera *g*.

Ora però c'è un problema, dopo il boot successivo allo spegnimento questa impostazione viene persa (per sicurezza), se vogliamo renderla definitiva occorre modificare le opzioni dei file di configurazione della rete: purtroppo editare `/etc/sysconfig/network-scripts/ifcfg-eth0` ed aggiungere `ETH-TOOL_OPTS=wol g` non funziona in F17 (non ho indagato ancora sul perchè) ed allora occorre procedere diversamente:

```
# gedit /etc/rc.d/rc.local
```

inserite:

```
#!/bin/bash  
ethtool -s p4p1 wol g
```



salvate chiudete, date:

```
# chmod +x /etc/rc.d/rc.local
```

```
# systemctl enable rc-local.service
```

per rendere la cosa definitiva.
infine date il comando:

```
# ip link
```

appuntatevi il mac-address della scheda di rete, supponiamo sia **AA:BB:CC:DD:EE:FF** e spegnete il computer.
passiamo ora alla:

Configurazione del router

Aprirete la pagina di configurazione del vostro router (*ad esempio 192.168.1.1*), andate nella pagina relativa al firewall/route nella sezione NAT aggiungete il forwarding della porta 40000 protocollo udp verso la broadcast 192.168.1.254, poi nella pagina degli accessi per la gestione attivate il server telnet (c'è sicuramente).

Useremo telnet perchè molti router non consentono le operazioni che stiamo per fare da interfaccia web. Aprirete un terminale e date il comando:

```
$ telnet 192.168.1.1
```

date la password di gestione del router, ammesso che l'abbiate impostata, e date il comando telnet:

```
> ip arp status
```

accertatevi che l'output contenga una linea del genere:

```
192.168.1.255 10 Mb Ethernet 0 ff:ff:ff:ff:ff:ff 43 NULL
```

se non è presente, date il comando:

```
> ip arp add 192.168.1.255 ether ff:ff:ff:ff:ff:ff
```

poi date comunque il comando:

```
> ip arp add 192.168.1.254 ether ff:ff:ff:ff:ff:ff
```

quest'ultimo comando serve ad attivare il broadcasting arp ricontrollate l'output di *ip arp status* e vedete se le linee inserite sono a posto.

potete uscire dal server telnet con un ovvio:

```
> exit
```

Fatto questo ritornate sulla pagina di configurazione del router e disattivate il server telnet; è molto pericoloso tenere un server telnet acceso, in quanto per sua definizione è privo di qualunque controllo di sicurezza.

Appuntatevi l'ip pubblico del router oppure il nome dell' host/dominio.

Non spegnete il router, se lo fate l'indirizzamento arp verrà perso e dovrete ripetere le operazioni sopra esposte.

Accensione da remoto - WOL - Wake On Lan

Utilizzate un computer collegato ad una qualsiasi rete esterna alla vostra, installate da root:

```
# yum install wol
```

ed infine date il comando risolutivo:

```
$ wol -i mio.ufficio.it AA:BB:CC:DD:EE:FF
```

dove **mio.ufficio.it** è il nome dell' host/dominio o l'ip pubblico e **AA:BB:CC:DD:EE:FF** è il mac-address della scheda di rete del computer da accendere.

il computer si accenderà. Attendete il tempo necessario che il vostro computer termini lo startup ed accedete con un sorriso al vostro server.

Comando divertente

Terminiamo con un comando piuttosto divertente, date da root un:

```
# yum install xcowsay fortune-mod
```

e poi da utente:

```
$ xcowsay -think -t 15 -cow-size=large 'fortune -s'
```

ecco cosa potrebbe uscire raffigurato sul vostro desktop.



Buon Divertimento.

Fedora 18

Robert Mayr (robyduck@fedoraonline.it)

Uno sguardo all'interno delle features approvate per la Spherical Cow

La versione 18 della nostra distro preferita porta il nome *Spherical Cow*, ma per adesso ha ricordato più la mucca pazza che quella sferica, viste le disavventure vissute in tutta la sua fase pre-rilascio.

Come spesso accade, però, le cose nate con difficoltà poi a volte si rivelano dei piccoli gioiellini, e Fedora 18 dovrebbe rientrare proprio tra queste.

Tra le novità che hanno dato più grattacapo agli sviluppatori, causando un ritardo delle versioni di rilascio mai visto fino ad ora, ci sono l'installer completamente riscritto e il Secure Boot, componente di Windows 8 che dovrebbe inibire il caricamento di software malevoli durante la fase di boot.

Ma andiamo con ordine e vediamo le più importanti features punto per punto:

Anaconda New UI

Il famoso installer che tutti abbiamo conosciuto e apprezzato negli ultimi anni per la sua linearità è stato mandato in pensione.

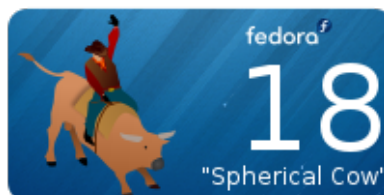
Al suo posto troviamo la versione 2, completamente riscritta da zero in pochissimi mesi e purtroppo causa principale di tutti i ritardi di F18.

Ed è proprio il suo pezzo forte, una migliore gestione delle partizioni, che ha creato più difficoltà.

Inoltre sarà possibile gestire più facilmente i mirror e la connessione rete già durante l'installazione.

Inutile dire che l'interfaccia grafica è completamente diversa e che il codice si è snellito di parecchio.

Questo comporta molti vantaggi anche per l'installazione in se, che dovrebbe essere più veloce rispetto a prima e dare all'utente più flessibilità in caso di sistemi multi-boot.



Secure Boot

Altra novità importante, ma sulla quale ancora adesso le discussioni non mancano, e la sensazione è che ci saranno ulteriori sviluppi nei prossimi mesi.

Come ben noto, Windows 8 introduce attraverso il firmware UEFI un sistema di boot verificato, che rende possibile l'avvio della macchina soltanto se il software che viene avviato durante il boot abbia una firma digitale Microsoft.

Fedora e Ubuntu da tempo lavorano a una soluzione per poter avviare il proprio sistema operativo, seguendo strade diverse.

Fedora ha chiesto e ottenuto la firma digitale Microsoft, mentre Ubuntu sta sviluppando le proprie firme.

Entrambe non sembrano soddisfare la Linux Foundation, che ha già annunciato la possibilità di un pre-bootloader per aggirare il problema dell'avvio di SO non firmati.

Vedremo come evolve, fatto sta che su Fedora 18 è possibile mantenere attivo il Secure Boot su macchine certificate Windows 8.

FedUp

L'introduzione del nuovo installer ha costretto gli sviluppatori anche a pensare a un nuovo strumento di aggiornamento.

FedUp sostituisce in tutto e per tutto il vecchio PreUpgrade, che deve essere utilizzato per aggiornamenti fino a Fedora 17, mentre un aggiornamento da F17 -> F18 sarà possibile soltanto con questo nuovo tool.

Gnome 3.6

Eccoci alla versione 3.6, un notevole passo in avanti è stato fatto e gli utenti Gnome saranno felicissimi della stabilità e dei feature contenuti in questa quarta versione di Gnome Shell.

KDE Plasma Workspaces 4.9

Include il Plasma Desktop e le aree di lavoro per i netbook, oltre ovviamente all'ultimo aggiornamento delle applicazioni e del Desktop KDE.

Xfce 4.10

La nuova versione di Xfce introduce innumerevoli bugfix e miglioramenti rispetto alla versione precedente.

DNF

Versione sperimentale di un nuovo package maintainer, che insieme alla libreria Hawkey e libsolv dovrebbe portare a una gestione pacchetti più snella e veloce.

RPM 4.10

La versione RPM viene portata alla 4.10.

256 color terminals

Di default i colori a disposizione dei terminali non avranno più 8 ma 256 colori.

Eucalyptus

Il software per il cloud computing per un'infrastruttura privata come il Cloud come servizio.

Firewalld

Sarà firewalld il servizio firewall di default in F18, al posto di iptables.

MATE Desktop

Sarà possibile, almeno abbinandolo a un altro Desktop Environment, utilizzare il DE basato su Gnome 2.x.

NetworkManager Hotspots

Viene introdotta la funzionalità Access Point in versione semplificata, per correggere problemi di compatibilità e fornire una più semplice funzionalità.

OpenStack

Viene aggiornato all'ultima release, nome in codice Folsom.

oVirt engine 3.1

L'ultima versione include la GUI e alcune nuove funzionalità.

PowerPC ppc64p7 subarch support

Viene introdotto il supporto per l'architettura secondaria ottimizzata di PPC64 Power 7.

Python 3.3

Importante aggiornamento all'ultima versione di Python.

Apache 2.4

Fedora 18 arriva con il webserver in versione 2.4 che introduce moltissime novità. I file di configurazione di Apache 2.2 potrebbero non più funzionare a causa di questo aggiornamento.

Samba 4

Nuovissima versione appena rilasciata di Samba, le novità con il passaggio a questa nuova major release sono molte.

SELinux Systemd Access Control

Systemd avrà accesso a SELinux per controllare se un processo ha i permessi di gestire un file unit.

/tmp on tmpfs

Ora un tmpfs di default viene montata in /tmp. Tutte le novità introdotte da Spherical Cow sono disponibili alla pagina ufficiale: <http://fedoraproject.org/wiki/Releases/18/FeatureList>.

Una cosa la sappiamo già adesso. Fedora 18, una volta installata, è tra le più performanti e stabili versioni degli ultimi 2 anni e darà ai suoi utilizzatori tante soddisfazioni.



Come spesso accade, però, le cose nate con difficoltà poi a volte si rivelano dei piccoli gioiellini, e Fedora 18 dovrebbe rientrare proprio tra queste

Rober Mayr (robyduck@fedoraonline.it)

FOLLie

FedoraOnLine

FedoraOnLine - Ecco le miglior FOLLie!

Diamo vita ad una nuova rubrica, direttamente da **FedoraOnLine** e chi pensava che il portale non potesse divenire fonte di divertimento, **rimarra di stucco!**

1

Fedora 17 - Programmi	Montare una ISO
Pivello	Domani 12:00:00
 <p>Utente in difficoltà Da: Loti Registrato: 13-13-2013 Messaggi: 113</p>	<p>Buongiorno, sono un utente in difficoltà con il montaggio di un file immagine iso, come si esegue da linea di comando? Grazie in anticipo.</p>
Carlo	Domani 12:01:00
 <p>Maestro Da: Milano Registrato: 01-01-2003 Messaggi: 1313</p>	<pre>mkdir /media/prova mount -o loop /percorso/file_immagine /media/prova C.</pre>
Pivello	Domani 12:04:00
 <p>Utente in difficoltà Da: Loti Registrato: 13-13-2013 Messaggi: 114</p>	<p>Ho un problema, al comando: <code>C.</code> ho avuto come risposta dal sistema: <code>bash: C.: comando non trovato...</code> che faccio?</p>
Carlo	Domani 12:07:00
 <p>Maestro Da: Milano Registrato: 01-01-2003 Messaggi: 1314</p>	

Fedora 17 - Programmi	Cercare un file
Pivello	Dopodomani 17:45:00
 <p>Utente in difficoltà Da: Loti Registrato: 14-15-2016 Messaggi: 16</p>	<p>Buongiorno a tutti, sto cercando di usare il comando find, il file che cerco è "gatto" ma spesso mi risponde: find: '/tmp/systemd-namespace-AK15yv': Permesso negato ripetendo il messaggio tante volte. è possibile evitare che stampi questi messaggi?</p>
Collaboratore	Dopodomani 17:49:00
 <p>Maestro Da: Milano Registrato: 01-01-1986 Messaggi: 2530</p>	<pre>find / -name gatto 2>/dev/null</pre>
Pivello	Dopodomani 18:05:00
 <p>Utente in difficoltà Da: Loti Registrato: 14-15-2016 Messaggi: 17</p>	<p>Ma che comando mi hai fatto lanciare? ora il mio disco non funziona più!!</p>
Collaboratore	Dopodomani 18:06:00
 <p>Maestro Da: Milano Registrato: 01-01-1986 Messaggi: 2531</p>	<p>Puoi riportare con precisione quali comandi hai dato al sistema?</p>
Pivello	Dopodomani 18:05:00
 <p>Utente in difficoltà Da: Loti Registrato: 14-15-2016 Messaggi: 18</p>	<p>Beh, mi sono documentato, ho approfondito la cosa e ho modificato il comando, ho dato da root: <pre>find / -name gatto 2>/dev/sda</pre> ho sbagliato qualcosa?</p>
Collaboratore	Dopodomani 18:06:00
 <p>Maestro Da: Milano Registrato: 01-01-1986 Messaggi: 2532</p>	

2

FOLIO - IL WEBMAGAZINE DI FEDORAONLINE ([HTTP://WWW.FEDORAONLINE.IT](http://www.fedoraonline.it))

RIVISTA NON PROFESSIONALE TEMATICA E LIBERA CREATA DAI
LETTORI DI FEDORAONLINE, SCARICABILE E DAI CONTENUTI
FORNITI DAGLI UTENTI FEDORA IN ITALIA E NEL MONDO.



Folio ed i suoi contenuti sono distribuiti con licenza creative commons, reperibile a questo link:
<http://creativecommons.org/licenses/by-sa/3.0/it/>

Questa rivista segue le linee guida dettate dal Fedora Project reperibile all'indirizzo http://fedoraproject.org/wiki/Legal:Trademark_guidelines:

Publications

"It is permissible to use the Fedora trademarks in the title and content of a publication, provided that: the use is clearly in reference to the Fedora Project or its software the use does not imply sponsorship or endorsement by Red Hat or the Fedora Project Proper trademark symbols are used in connection with the Fedora Trademarks and the trademark attribution statement must appear as explained in Proper Trademark Use."
Fedora® and the Infinity design logo are trademarks of Red Hat, Inc.
