

folio

Il Magazine di fedoraOnline

- Introduzione al fedora Project

- Intervista: Ambassador Fedora

- FedoraOnline: i motivi del cambiamento

- Uno sguardo al terminale

- Fedora 17: le Features

... 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

Collaboratori:

Andrea VERRI

a verri@fedoraproject.org

Interventi:

Gianluca SFORNA

giallu@fedoraproject.org

Grafica:

Cristian POZZESSERE

cristianpozzessere@gmail.com

IN QUESTO NUMERO

Editoriale (pag.1)

Spiegati i motivi di questo Webmagazine, fortemente voluto.

di Gabriele Trombini

Il Fedora Project (pag.3)

Cos'è il Fedora Project? Ecco cosa, dove, come e perché!

di Gabriele Trombini

Intervista a: Gianluca Sfor- na (pag.7)

Uno degli Ambassador tra i più attivi risponde alle nostre domande.

di Gabriele Trombini

Fedora Inside (pag.10)

Presentazione di Andrea Veri; la nostra guida all'interno del progetto Fedora.

di Gabriele Trombini

FedoraOnLine (pag.11)

Cronistoria e retroscena del passaggio da FOL 1 a FOL 2.

di Gabriele Trombini

Primi passi (pag.14)

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

di Gabriele Trombini

Sistema base (pag.19)

Panoramica sulla gestione dei servizi (prima parte).

di Gabriele Trombini

Sistema base (pag.23)

Dracut, cosa è ed a cosa serve.

di Gabriele Trombini

Sistema avanzato (pag.26)

La creazione di un rpm, struttura di base (prima parte).

di Gabriele Trombini

Sistema avanzato (pag.29)

Introduzione al Kernel Fedora (prima parte).

di Giuseppe Delvecchio

Sistema e sicurezza (pag.32)

Cosa è e come funziona il keyserver Fedora, generazione di chiavi gpg.

di Andrea Veri

Notizie (pag.35)

Sveliamo le features della prossima release: Fedora 17 (Beefy Miracle).

di Robert Mayr

Editoriale

Gabriele Trombini (mailga@fedoraonline.it)

Folio - il primo numero e le nostre aspettative

Carissimi lettori,
era da parecchio tempo che avevamo in
cantiere l'idea di redigere una rivista in
formato PDF che parlasse di tematiche
al di fuori di quelle trattate all'interno
del forum tecnico.

Il progetto, sempre rimandato soprattutto per mancanza di tempo, ha ripreso il vigore necessario quando si è stabilito che era il momento di gettare il cuore oltre l'ostacolo.

Di certo non potevamo lasciarci sfuggire l'occasione fornita dal restyling del sito, per cui eccoci a presentare il primo numero (numero pilota) di questa nuova iniziativa di FedoraOnLine e del suo staff.

E' sicuramente un layout non definitivo, da affinare e collaudare nel tempo; una prova, quindi, per verificare la centralità del progetto rispetto alle aspettative dell'utente.

Non vogliamo che diventi una sovrapposizione di FedoraOnLine, pertanto ciò che tratteremo in questa sede saranno argomenti, talvolta tecnici, che descriveranno

il funzionamento di programmi, del sistema e di tutte quelle particolarità che nel forum non vengono approfondite per ovvie ragioni di tempo e spazio, ma che riteniamo essere di importanza rilevante per gli utenti.



Figura 1: il nuovo portale FedoraOnLine

Verranno anche portate avanti e sviluppate tematiche connesse a Fedora, al Fedora Project ed a quant'altro ruoti intorno ai sistemi GNU/Linux con l'intenzione di accorciare la distanza che intercorre tra gli utenti ed il Fedora Project, scrivendo diffusamente sulle qualità e, perché no?, dei difetti della struttura organizzativa, conoscendo anche da vicino chi è impegnato nelle varie attività ad esso correlate.

Ma non solo! FedoraOnLine, tramite questa rivista, vuole cercare di ridurre quelle difficoltà che i nuovi utenti riscontrano nell'avvicinamento a GNU/linux, creare-

si sia per oggettive complessità tecniche sia per stereotipi che non hanno senso di esistere; in sostanza proveremo a descrivere una visione a 360 gradi dell'open source e del free software.

Non abbiamo mire editoriali, non siamo professionisti della carta stampata, nè vogliamo esserlo, ma questa iniziativa è rivolta a coloro, soprattutto, che hanno voglia di guardare e capire cosa ci lega ad uno sviluppatore dell'Illinois, ad un grafico dell'India e a chissà quante altre persone.

Basta un gesto del mouse o premere un carattere sulla tastiera affinchè i *linuxiani* si pongano delle domande.

Già qualche anno fa eravamo riusciti a trasformare una idea in qualcosa di buono e di utile, scrivendo un libro sulla release 9 di Fedora, che fornisse un supporto per l'avvicinamento al nostro sistema operativo.



Figura 2: Fedora 9, il libro

Ovviamente siamo del tutto aperti a con-

sigli, critiche e a correzioni da parte di chi legge ed il riferimento a cui inviare tutto ciò che vi dovesse venire in mente è redazione@fedoraonline.it.

In questo primo numero cominceremo ad introdurre il Fedora Project, descrivendone gli obiettivi e parlandone un po' con chi ne è addentro, parleremo dei motivi e dei retroscena che ci hanno portato al restyling del nostro sito, vedremo come avvicinarci alla shell Linux, approfondiremo software, illustreremo i servizi di Fedora, daremo una prima occhiata al kernel e cominceremo a studiare il sistema di packaging di Fedora, cioè gli rpm.

Ovviamente non potremo fare a meno di gettare uno sguardo sul futuro prossimo di Fedora, alle features annunciate per la release numero 17 la cui uscita è prevista a Maggio 2012, come da roadmap che si può trovare alla pagina <http://fedoraproject.org/wiki/Schedule>.

Niente altro da aggiungere, al momento, specificando, però, che l'impostazione grafica della rivista sarà soggetta a miglioramenti e che per il momento la cadenza per la pubblicazione dei numeri successivi a questo è ancora lontana dall'essere definita.

Il successo di questa iniziativa dipende dai nostri lettori, vi aspettiamo.

Fedora Project

Gabriele Trombini (mailga@fedoraonline.it)

Il Fedora Project - comprensione di un grande fenomeno legato al software libero

Credo che tutti noi ci siamo chiesti almeno una volta quale lavoro stava dietro al nostro sistema operativo. Quali fossero i mezzi utilizzati e quante persone erano impegnate nel suo sviluppo.

Tutto è riconducibile al Fedora Project (<http://fedoraproject.org/>) che si può definire un contenitore per le idee che vengono portate avanti da singoli utenti che condividono il proprio lavoro.



Figura 3: Il sito del Fedora Project

Chiunque voglia impegnarsi attivamente nel progetto potrebbe trovare un po' difficoltoso avventurarsi all'interno del suo sito web in quanto la sua organizzazione è molto capillare.

Vediamo, in questo articolo, di chiarire un po' le idee degli utenti.

Innanzi tutto è bene precisare i termini del legame tra Red Hat ed il Fedora Project, come spiegato all'indirizzo web <http://it.redhat.com/resource/library/articles/relationship-between-fedora-and-rhel>, dove ben si definisce che sono due entità separate, la prima con l'obiettivo dell'IT, la seconda a vantaggio della comunità:

"Le dimensioni e l'esperienza della comunità di Fedora, rendono Fedora l'incubatrice e il terreno di prova ideale per le funzioni che, in seguito, verranno integrate in Red Hat Enterprise Linux. Per soddisfare i requisiti di qualità e affidabilità che rendono Red Hat Enterprise Linux la scelta ideale per le applicazioni di importanza cruciale, i test e processi di controllo di qualità di Red Hat Enterprise Linux sono separati e distinti da quelli di Fedora."

Fedora: sviluppo rapido della tecnologia più recente.

La comunità di Fedora è composta da migliaia di utenti, collaboratori e sostenitori, i quali interagiscono tramite i forum

online, le mailing list ed i wiki per sostenersi l'un l'altro. Con uno sviluppo e un ciclo di distribuzione molto rapidi, Fedora fornisce la tecnologia più recente sulle piattaforme hardware attuali.

Red Hat Enterprise Linux: piattaforma open source stabile e supportata.

Se si sceglie di eseguire Red Hat Enterprise Linux, si entra in stretto contatto con il principale fornitore di soluzioni open source. Non solo si ottiene una piattaforma affidabile e stabile, con un ciclo di vita di 10 anni, ma si può godere anche dei vantaggi offerti da organizzazioni internazionali di progettazione, consulenza e assistenza. Una sottoscrizione Red Hat Enterprise garantisce l'accesso a software e manutenzione di alta qualità, oltre alle informazioni e all'assistenza che si estendono al ciclo di vita e all'architettura dell'intera infrastruttura.”

Nella pagina del sito Fedora <http://fedoraproject.org/sponsors> si specifica che la casa madre mette a disposizione del progetto infrastrutture, personale e finanziamenti:

“Red Hat, Inc. è lo sponsor principale per il Fedora Project. Red Hat fornisce al Fedora project una varietà di risorse, che includono il supporto dei dipendenti a tempo pieno, infrastrutture hardware e di banda, finanziamento degli eventi e consulenza legale.”

Ricordiamoci bene, però, che Red Hat non è Fedora.

Il Fedora Project è il contenitore ideale dove, chiunque di noi, può sostenere, integrare, migliorare, creare i progetti che diventeranno parte integrante dell'idea che sta alla base di tutto questo.

Non è possibile scindere la totalità del progetto dal sistema operativo perché esso è l'emanazione ultima di quello che è stato creato; la sua punta di diamante.

Prima di essere un sistema operativo, Fedora è un crocevia di libero pensiero. È riduttivo, infatti, qualificare il Fedora Project come un sistema operativo; alle spalle di esso c'è una comunità internazionale che sostiene i settori di sviluppo.

Le attività del progetto sono molteplici, dallo sviluppo di applicazioni, allo sviluppo di rami del kernel, da quello del design a quello della internazionalizzazione, da quello del marketing a quello del web e altro ancora.

La distanza tra l'utilizzatore ed il progetto, non è poi così lontana, anzi, è a portata di mano! Una semplice digitazione di www.fedoraproject.org sulla barra degli indirizzi e si è già dentro, immersi nelle possibilità che ti può dare.

I valori alla base del lavoro di tutte le

persone coinvolte, sono quattro:

-  libertà
-  comunità
-  condivisione
-  innovazione

e ciascun contributo deve rispettare questi valori.

In questo articolo vorrei soffermarmi sul concetto di comunità, estesa al mondo intero, entro la quale ciascuno di noi può (come detto) dare il proprio contributo.

I mezzi a disposizione degli utenti sono davvero molti e molto potenti, a cominciare proprio da una area di interesse ben definita.

Dividiamo pure in macroaree all'interno delle quali è possibile trovare il proprio spazio specifico e ben dettagliato:

-  redattore
-  disegnatore
-  comunicatore
-  sviluppatore del sistema operativo
-  traduttore
-  sviluppatore web o amministratore

Prima di andare ad analizzare questi ruoli, è bene specificare il ruolo del FAS (Fedora Account System), a cui ci si deve iscrivere, per poter iniziare la propria attività in favore di Fedora.

Il FAS, raggiungibile all'indirizzo <https://admin.fedoraproject.org/accounts/user/new> è il sistema per poter richiedere l'associazione ad un gruppo, gestito dagli amministratori. Infatti gli iscritti sono tutti gli appartenenti ai gruppi di lavoro ed è all'interno di esso che è possibile sottoscrivere la domanda di ammissione.

Avere un account FAS valido ci permette di avere uno spazio personale nel wiki all'interno del quale è possibile descrivere le proprie esperienze ed i propri contributi.

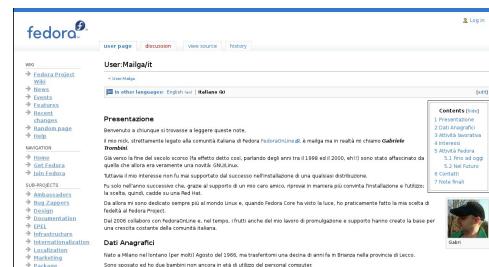


Figura 4: pagina wiki Mailga

Nella pagina web <https://fedoraproject.org/wiki/Projects> è possibile avere una lista dettagliata e le spiegazioni dei vari progetti.



Il ruolo di redattore è particolarmente adatto a coloro che hanno particolare facilità di scrittura. In questa area trovano spazio sia i collaboratori che scrivono i contributi, sia coloro che li traducono nelle varie lingue.



In questo gruppo potrà trovare una attività congeniale chi è particolarmente abile nelle arti grafiche.



Questa macrosezione è pensata per le persone che hanno attitudini alla comunicazione, sia essa in pubblico che attraverso attività di marketing.



I gruppi appartenenti a questa categoria sono fondamentali per la riuscita del sistema operativo. Le persone con conoscenze sulla programmazione potrebbero partecipare alle attività degli sviluppatori.

Un'altra sezione veramente importante per la diffusione del sistema nei vari paesi



del mondo è proprio questa. Chiunque possa aiutare nella traduzione di tutta la documentazione Fedora nella propria lingua madre, avendo conoscenze dell'inglese, troverà ampio spazio.



Sviluppatore web/sysadmin

Infine, chi è addentro al linguaggio web, allo sviluppo di applicazioni web e/o all'amministrazione dei sistemi Linux può davvero risultare fondamentale per l'infrastruttura del Fedora Project.

Tutto quanto gira intorno e all'interno del progetto è nato nello spirito open e free, che abbiamo visto essere tra i fondamenti delle attività ad esso legate.

Ed è proprio in questo che si deve partire per comprendere il fenomeno, perchè chi non intende lasciare libera una comunità di proporre, di decidere e anche di sbagliare, non potrà sperare di ottenere risultati apprezzabili.

Intervista a: Gianluca Sfora

Gabriele Trombini (mailga@fedoraonline.it)

Le interviste di Folio - incontriamo Gianluca Sfora

Abbiamo chiesto al nostro amico Gianluca di rispondere a 10 domande, sperando anche di metterlo in difficoltà. La qualità delle risposte non ci ha affatto sorpreso.



Chi è Gianluca Sfora? (@giallu)
classe 1972, programmatore e sysadmin.
Inizia a lavorare stabilmente su Linux nel 1999 con Red Hat Linux 6.0; nel 2003 diventa prima utente e poi contributor del progetto Fedora.
Oggi è maintainer di diversi pacchetti RPM ospitati tra repository ufficiali (Fedora, RPM-Fusion) e privati, traduttore ed ambassador del progetto.
Il suo blog:
<http://morefedora.blogspot.com>

FOL: 1) chi è il Fedora Project?

Giallu: Il progetto non è nessuno in particolare, ma in effetti si può dire che tutti i partecipanti al progetto in qualche modo sono Fedora: infatti anche il più piccolo dei contributi permette al progetto e agli altri partecipanti di crescere e migliorarsi, quindi il progetto nel suo

complesso si può sicuramente considerare “figlio” di ogni contributor. Non a caso ho partecipato ad EuroPython 2011 con questo badge:



FOL: 2) gli ambassador promuovono il progetto o il sistema operativo?

Giallu: Direi entrambi: gli Ambassador promuovono senz’altro il progetto e i suoi valori, ma è chiaro che la distribuzione era e rimane il prodotto principale e più visibile quindi trovo difficile che si possa scindere i due aspetti.

FOL: 3) sbaglia qualcosa, il Fedora Project? Se sbaglia, in che modo?

Giallu: Un progetto così grande sicuramente ha sia punti di forza che debolezze. E’ però praticamente impossibile dare un giudizio assoluto, visto che tutto dipen-

de da cosa si usa come metro per definirne il successo. Se per dire mi aspetto dal progetto una distribuzione che posso installare su un server e supportare per una decina di anni, troverò che si stanno compiendo molti errori. Lo stesso se mi aspetto una distribuzione “rolling”, che includa sempre e comunque solo le ultime versioni di ogni software. Da notare che gli obiettivi del progetto non sono “statici” ma sono stati più volte sviluppati, raffinati e chiariti; ad oggi, se esaminiamo la pagina del wiki che li elenca (<https://fedoraproject.org/wiki/Objectives>) direi che il progetto nel suo complesso è decisamente sulla buona strada.

FOL: 4) cosa rubereste a Debian, a Slackware ed a gentoo?

Giallu: Premesso che dopo tanti anni di uso esclusivo di Fedora (e derivate) non conosco bene la situazione dei nostri “cugini”, devo dire che di Debian ammire l’organizzazione complessiva che riesce a tenere un gran numero di utenti e sviluppatori anche a dispetto dei loro ingombranti vicini di casa.

FOL: 5) non vi pare che i linux day e i Fudcon siano eventi riservati ad una ristretta utenza, in fondo si riferiscono sempre agli stessi ambienti?

Giallu: Credo che i due eventi siano profondamente diversi: se i Linux Day si ri-

volgono ad una ampia platea essendo un evento divulgativo per definizione (anche se poi sono tanti i partecipanti già introdotti alla filosofia del software libero), il FUDCon mi sembra più un evento del fare e per questo sicuramente ricco di utenti avanzati e sviluppatori che approfittano della occasione di aggregazione per accelerare quelle attività di sviluppo a volte rallentate dalla distanza fisica tra i contributor. Concordo comunque sul fatto che siano eventi di “nicchia”, visto che comunque ancora oggi, a dispetto di tutti i progressi fatti, Linux continua ad essere il settore “Altro” vicino a “Windows” e “MacOSX” dei diagrammi a torta.

FOL: 6) dove pensate che si possa essere più incisivi nella promozione?

Giallu: Direi un po’ ovunque: scuole, università, LUG, eventi, aziende pubbliche private etc. Ad oggi mi pare evidente che altre distribuzioni siano riuscite ad attrarre un grosso numero di utenti (che è comunque un bene visto che per lo più provengono da altri sistemi operativi). Se consideriamo che Fedora si rivolge ad un pubblico più consapevole la situazione non è drammatica, ma siccome penso sia più difficile “convertire” un utente piuttosto che acquisirlo la prima volta, spero che lo sforzo di marketing sul progetto sia sempre più efficace.

FOL: 7) perchè l’utente dovrebbe usa-

re fedora e non altre distro? solo per il free software, non vi pare un po' troppo idealistico?

Giallu: Il software libero l'utente lo trova in tutte le distro; non tutte però ti garantiscono che la totalità dei contenuti siano effettivamente ridistribuibili; per dire, se sei una azienda ed hai intenzione di produrre e vendere un prodotto hardware o software basato su Linux, Fedora può essere una ottima scelta proprio per questo motivo. La altre motivazioni si trovano nella metodologia di sviluppo: collaborazione con gli upstream (i progetti su cui si basa la distribuzione tipo kernel, GNOME, KDE, etc), aggiornamenti rapidi (siamo quasi sempre i primi a fornire le ultime versioni disponibili), totalità degli strumenti usati disponibili (per cui chiunque può farsi la "sua" distribuzione derivata) L'utente comune tende invece a dare valore ad altre caratteristiche, ad esempio che i suoi mp3 o DVD si possano riprodurre semplicemente cliccandoci sopra (per inciso, questa cosa funziona proprio in questo modo in Fedora, previa attivazione dei repository RPMFusion) oppure che la esistenza di driver proprietari venga notificata automatizzandone l'installazione. Purtroppo questo a volte si scontra con il fatto che Fedora punta alla produzione e integrazione del migliore e più recente software libero: per esempio, molto spesso nel passato i driver ATI disponibili erano incompatibili con il server X o con il

kernel distribuiti in quel momento da Fedora, e solo con mesi di ritardo AMD ha provveduto a fornire driver funzionanti; in queste condizioni non c'è molto che il progetto possa fare, se non lavorare ancora di più per rendere inutili tali driver, sostituiti dalle loro controparti open.

FOL: 8) ritieni di aver fatto tutto il possibile per promuovere Fedora?

Giallu: Sicuramente no: il lavoro non manca mai, ma il tempo e l'energia a volte non sono facili da trovare. Per questo penso sia sempre prioritario trovare e motivare nuovi contributor, così che la riserva di idee e manodopera per nuove iniziative sia sempre alimentata.

FOL: 9) Fedora avrà un ritorno dall'utenza italiana?

Giallu: Sicuramente. La barriera alla partecipazione al progetto è sempre più bassa e chiunque, a prescindere dalle proprie capacità tecniche, può dare un contributo. Basta guardarsi un po' intorno per scoprire il team più adatto alle proprie inclinazioni e capacità ed iniziare!

FOL: 10) Come siamo noi italiani rispetto alla totalità del progetto?

Giallu: Ci sono team ben più organizzati, ma devo dire che gli Italiani nel progetto non mancano a tutti i livelli. Se solo trovassimo il modo di aumentarne la coesione credo che saremmo tra i gruppi più importanti del progetto...

Fedora Inside

Gabriele Trombini (mailga@fedoraonline.it)

Una collaborazione speciale - Andrea Veri

Fin da questa prima uscita e con grande piacere, possiamo avvalerci della collaborazione di un amico che opera nell'infrastruttura del Fedora Project e in posizioni prestigiose all'interno del panorama GNU/Linux.

Visto la natura delle argomentazioni che Andrea tratterà all'interno di Folio (principalmente sicurezza e sistemi, talvolta spiegherà dinamiche viste dall'interno del progetto) la presenza dei suoi contributi, ovviamente, non potrà avere il carattere della continuità.

Andrea Veri è un ventiduenne studente della facoltà di Giurisprudenza dell'Università di Udine.

Da sempre appassionato di informatica, iniziò nel 2006 a contribuire nei campi della documentazione e del packaging di applicazioni in Ubuntu per poi spostarsi successivamente in Debian, dove ottenne il titolo di sviluppatore ufficiale con i relativi accessi all'archivio principale della distribuzione.

Dal 2009, iniziò la sua collaborazione con il progetto GNOME nel quale ricopre al-

cune posizioni di carattere infrastrutturale, nello specifico, si occupa di amministrare e gestire il server LDAP di GNOME e presiede la GNOME Membership & Elections Committee, la quale si occupa di valutare, accettare o rifiutare le richieste di appartenenza alla Fondazione GNOME e non solo: tale commissione è, inoltre, incaricata di organizzare le elezioni della Board of Directors a Giugno prima del GUADEC.

Dal 2010, Andrea si sposta, pur mantenendo attive le altre sue collaborazioni con Debian e GNOME, in Fedora.

I primi gruppi di lavoro furono relativi alle traduzioni in lingua italiana, al gruppo Insight ed a quello Infrastructure.

Seguì poi la creazione del sub-planet Just-Fedora e la successiva candidatura alla Board di Fedora.

Al momento, le sue attività si concentrano maggiormente nel packaging RPM e nella amministrazione di sistemi di alcune macchine utilizzate dall'infrastruttura di Fedora.

Il nuovo FedoraOnLine: motivazioni e curiosità

Gabriele Trombini (mailga@fedoraonline.it)

FOL 2 - il percorso pieno di imprevisti

Il nostro webmaster, come molti di voi certamente sapranno, è uno stakanovista del lavoro in FedoraOnLine, un vulcano di idee, ed è sempre pronto a valutare qualsiasi possibilità di crescita.

Da molto tempo, all'interno dello staff, si discuteva circa un possibile rinnovamento grafico e concettuale del sito. Le idee erano molte ma la focalizzazione degli strumenti atti a raggiungere lo scopo erano ben remoti dall'essere stabiliti.

La questione non si poneva solo in termini di layout, la rivisitazione avrebbe dovuto riguardare la totalità del sito: si trattava di scegliere il cms in sostituzione di Xoops, che ci avrebbe permesso di gestire in maniera più efficace le sezioni presenti.

Prese così forma l'obiettivo da raggiungere, che comprendeva sia il cms che dei gestori di contenuti (in primis il forum stesso e la sezione documentazione).

Il lavoro che si presentava era decisamen-

te superiore alle forze presenti nello staff, anche perchè il forum "vecchia versione" doveva avere continuità.

Pertanto, definite le zone di interesse e gli strumenti per la loro gestione (Drupal a livello di cms, fluxbb a livello di forum e mediawiki per la parte documentale), abbiamo cominciato a lavorare sulle varie sezioni, ricercando anche dei validi collaboratori per ottenere i risultati sotto gli occhi di tutti.

Virus, Trpost ed io avremmo continuato a gestire il forum insieme, mentre Robyduck, MarioS e Pagnolo avrebbero cominciato a vedere come mettere insieme tutti gli strumenti che erano stati scelti.

Era l'estate dell'anno scorso e Virus ci mise a disposizione un server git per poter mettere insieme il lavoro fatto in comune.

Il primo grosso problema che si presentò era il trasferimento del database di Fol versione 1.0 in quello che sarebbe dive-



Figura 5: Il nostro server git

nuto Fol 2.0, perchè non aveva senso perdere la nostra storia.

Il secondo grosso problema, che ci era sembrato insormontabile, inizialmente, era dato dal login unificato tra Drupal e Fluxbb (non aveva senso mantenerlo anche per Mediawiki, visto che solo i redattori iscritti in mailing list, possono fornire contributi).

In entrambi i casi il lavoro fu abbastanza duro e richiedemmo aiuto anche ai nostri cugini d'oltralpe di fedora-fr.org, i quali ci indirizzarono verso una soluzione con delle criticità troppo onerose per le nostre energie.

Nel frattempo Robyduck e Pagnolo si occupavano della sistemazione grafica, uniformandola nei diversi programmi coadiuvati dai collaboratori per la parte grafica, lo storico Tuzzer ed Ilnanny mentre il nostro MarioS ricercava un modo più semplice per poter fare il login unificato

e per il trasferimento dei dati da un database all'altro.

Tutto procedeva, a Folio era stata data una priorità ovviamente secondaria, se nonchè, a causa di scelte personali Trpost si distaccò dal nostro gruppo di lavoro e impegni lavorativi portavano lontano dal progetto anche Tuzzer.

Se contiamo il fatto che alcuni utenti, solitamente presenti e attivi in seno alle notizie ed al forum, che ci avevano dato disponibilità a prendere in mano un settore di interesse, a causa di problemi personali non hanno potuto anch'essi proseguire nella collaborazione, ci siamo trovati, ad un certo punto, in grossa difficoltà.

Riassumendo, eravamo bloccati sulla soluzione del login (il trasferimenti da un database all'altro non era più un problema), la gestione del forum aveva subito una defezione, il supporto grafico era dimezzato e alcuni collaboratori non potevano darci una mano.

Per di più bisognava cominciare a tracciare le linee per la redazione di Folio.

Il tutto faceva prendere una brutta piega al lavoro, che subiva, quindi, un rallentamento.

La svolta ci fu quando MarioS ha trovato il modo per il login unico (ricordiamo

l'importanza di avere un login nella home del sito e non doverlo ripetere qualora si entri nel forum) e con il mio spostamento sull'attività di Mediawiki (insieme a Robyduck, che si occupava di tutto, e di Zamingas).

Mediawiki aveva bisogno di essere impostato per quanto riguarda le due pagine di fondamentale importanza, “contribuisci” e “convenzioni di scrittura”, che furono fatte in breve tempo da me e da Robyduck, dopo aver stabilito che anzichè appoggiarci al server git, per la parte documentale sarebbe stato meglio creare online la sezione.

Passammo poi, con il contributo di ilnanny, MarioS stesso e Zamingas, a importare le guide dal vecchio sito al nuovo; operazione tediosa ma relativamente breve.

Facendo le somme, a quel punto capimmo che era cominciata la fase di discesa; Drupal era ad un passo dalle rifiniture finali, il forum era stato importato nel nuovo database di Fluxbb, Mediawiki era finito e lo script di login era praticamente terminato.

Ormai ce l'avevamo fatta, per cui mi sono buttato a capofitto su Folio, se non che, la mia decisione iniziale di redigerlo con Scribus faceva a pugni con quello che nel web viene solitamente fatto: cioè l'utilizzo di Latex.

Dovevo cominciare daccapo, leggere e studiare, studiare e capire un linguaggio non propriamente facile anche se redditizio in termini di resa.

A questo punto, ed è storia di oggi, siamo online, con dei ritorni positivi e con l'uscita di questo numero di Folio.



Figura 6: Il nostro forum

La missione è compiuta, siamo orgogliosi di aver messo a disposizione degli utenti uno strumento che possa dare risposta alle richieste.

E Virus? Ragazzi! Senza Virus che ha retto sulle proprie spalle il carico del forum senza creare disagio agli utenti, non ho difficoltà ad ammettere che probabilmente tutto questo non avrebbe potuto essere così.

That's all, folks!

Linea di comando

Gabriele Trombini(mailga@fedoraonline.it)

(Parte prima) Consolle virtuale e terminale - la base dei sistemi GNU/Linux

La linea di comando è sicuramente un ostacolo per chi si avvicina a Linux.

Se è vero che un utente smaliziato lavora, in molti casi, con la shell aperta, è altrettanto vero che molti nuovi utenti si trovano impreparati all'utilizzo di quello che è universalmente riconosciuto come uno strumento essenziale dei sistemi del pinguino.

Se vogliamo avvicinarci con la giusta sicurezza a questo strumento, dobbiamo tenere ben presente che:

- *la shell è entusiasmante;*
- *la shell è potente;*
- *la shell non accetta la troppa confidenza.*

Cominciamo con il fare chiarezza sui termini che normalmente vengono usati anche per definire la medesima cosa, ma che hanno significati differenti seppur minimi:

- *terminale.*

Il terminale è lo strumento con il

quale si comunica con il sistema (tastiera e monitor).

- *shell.*

La shell è l'interprete dei comandi immessi tramite terminale. In un sistema con interfaccia grafica attiva (server X), la shell è ospitata all'interno di un terminale emulato.

- *consolle.*

La consolle è l'interfaccia testuale vera e propria, che si attiva al di fuori del server grafico.

Nell'uso comune, per comodità, i tre termini vengono intesi come lo stesso strumento, ma in questo articolo faremo riferimento alla *shell*, presente in tutti di desktop environments ed in particolare alla *shell bash (bourne again shell)*, installata di default dal sistema e sicuramente la più utilizzata.

Esistono varie tipologie di shell, tutte con delle caratteristiche in comune, ma che hanno delle differenze sostanziali apprezzate dai vari utenti più avanzati.

E' possibile verificare l'elenco delle shell installate con il comando:

```
$ cat /etc/shells
```



Figura 7: emulatore di terminale in Gnome (gnome terminal)

La shell è entusiasmante

Per chi arriva da altri sistemi operativi, dove l'interfaccia testuale è pressoché assente, si troverà spaesato di fronte alla necessità (talvolta assoluta) di dover aprire una shell per poter operare al meglio.

Dopo una normale diffidenza iniziale, in breve tempo avrà modo di apprezzare questo strumento basilare del sistema.

L'utente Linux è sempre spinto dalla curiosità e dalla voglia di capire ed è per questo che troverà molto piacevole appoggiarsi alla linea di comando.

La shell è potente

A differenza delle interfacce grafiche, la shell permette una iterazione con il sistema molto elastica, arrivando in profondità fino ai processi vitali.

Tutto il sistema, con le dovute autorizzazioni, può essere modificato e analizzato per mezzo dei comandi e opzioni di programmi di utilità e software aggiuntivi.

Inoltre la shell è in grado di interpretare il proprio linguaggio di scripting (*bash scripting*) in cui la concatenazione di comandi e operatori può dare modo all'utente di automatizzare la propria attività.

La shell non accetta la troppa confidenza

Proprio per le qualità di cui sopra, corriamo il rischio di acquisire troppa confidenza con uno strumento che, a causa della sua caratteristica di iterazione anche a basso livello con il sistema, potrebbe indurci all'esecuzione di operazioni senza la giusta dose di attenzione.

Si può arrivare a causare danni irreversibili alla propria station causando, spesso, perdite di dati, costringendoci ad un difficile recupero.

Non esiste un equilibrio oggettivo per questi fattori, il tutto è riconducibile alle proprie caratteristiche personali; di certo non

```

gabri@notebook:~$ cat /proc/cpuinfo
processor       : 0
vendor_id      : AuthenticAMD
cpu family     : 15
model          : 104
model name     : AMD Athlon(tm) 64 X2 Dual-Core Processor TK-53
stepping        : 1
cpu MHz         : 800.000
cache size      : 256 KB
physical id    : 0
siblings        : 2
core id         : 0
cpu cores       : 2
apicid          : 0
initial apicid : 0
fpu             : yes
fpu_exception   : yes
cpuid level    : 1
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 c
lflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt rdtscp lm 3dnowext 3dnow rep_good no
pl extd_apicid pni cx16 lahf_lm cmp_legacy svm extapic cr8_legacy 3dnowprefetch lbrv
bogomips        : 1593.91

```

Figura 8: visualizzazione delle caratteristiche del processore in gnome-terminal

bisogna temere il terminale ma sicuramente occorre avvicinarsi con piena coscienza di ciò che ci si accinge a fare.

Quindi il terminale ci consente di capire come funziona la nostra workstation ma, soprattutto, entriamo in sintonia con essa perché ci permette di capire cosa stiamo facendo e cosa fa il sistema in risposta al nostro comando.

Chi ha acquisito una buona manualità sostiene che con il terminale si lavora molto velocemente; e non lo dicono solo i “vecchi” utilizzatori.

Aprendo la shell la riga che compare (*/gabri@notebook ~/\$*) è già fonte di alcune informazioni come il mio nome utente (gabri), il nome della mia macchina

(notebook), in quale directory sono posizionato (la tilde ~ indica che mi trovo all'interno della mia home, ovvero /home/gabri) ed infine il simbolo del dollaro (\$), da distinguere da quello del cancelletto (#) che sta a significare che ho acquisito le credenziali di root (quest'ultimo caso verrà trattato successivamente).

A questo punto il terminale è pronto ad accettare i comandi che vengono imparati da tastiera.

\$ man nomecomando

Utilizzando il comando qui sopra, appare all'interno del terminale virtuale il manuale di istruzione per l'utilizzo del comando dato come argomento; è sempre bene studiarlo per destreggiarsi nell'esecuzione.

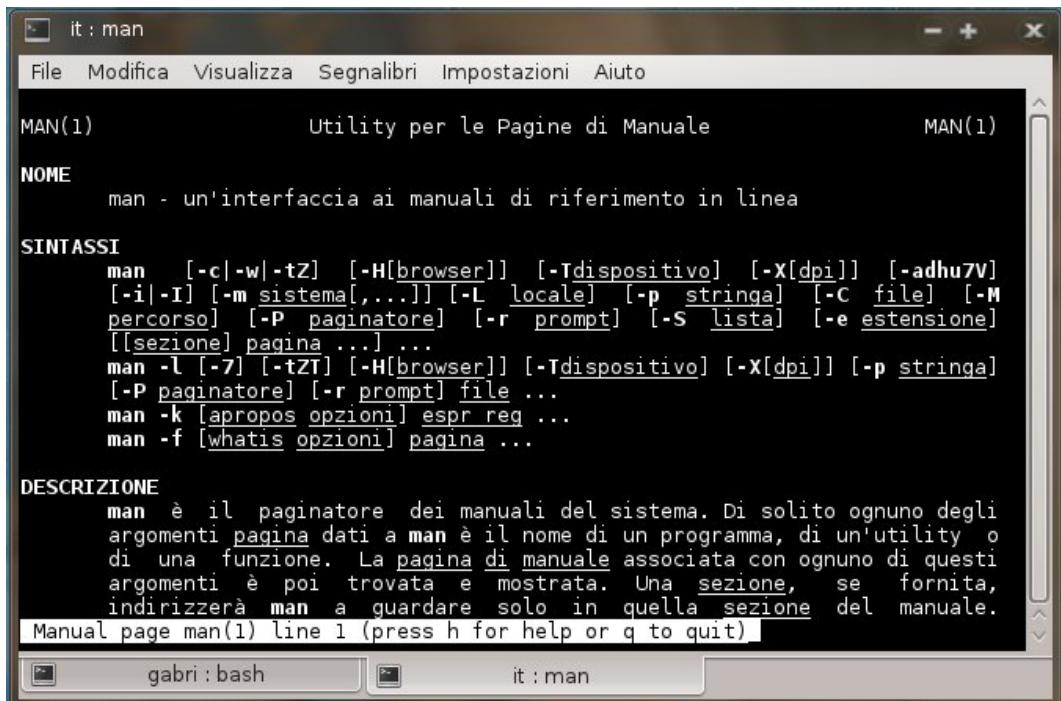


Figura 9: visualizzazione della pagina di manuale del comando *man*

cuzione secondo le nostre aspettative.

Già l'abituarsi alla lettura delle pagine *man* è un ottimo inizio, ma sicuramente il poter finemente configurare l'applicazione del comando mediante le opzioni rende molto interessante l'utilizzo della linea di comando.

Ovviamente il consiglio è quello di cimentarsi in esperimenti con comandi che non possono causare danni al sistema (come il comando *ls* oppure *cat* oppure lo stesso *man*), sebbene, utilizzando come utente e non come superuser, tali danni sarebbero circoscritti alla sola */home* dell'utente stesso.

La *Bash* è inoltre fornita di completamento automatico; ci basta scrivere le lettere di inizio del comando per avere, premendo due volte il tasto *tab*, l'elenco completo dei comandi che iniziano con quelle lettere che ho premuto.

E' possibile, inoltre, usare la *history* per ripercorrere ed eventualmente selezionare gli ultimi comandi dati mediante l'utilizzo dei tasti "freccia su" e "freccia giù".

Da linea di comando è possibile anche avviare i programmi, anche quelli grafici, semplicemente inserendo il nome dell'eseguibile prima dell'invio:

```
$ eseguibile
```

come in questo caso specifico in cui lanciamo il web browser *firefox*, senza opzioni:

```
$ firefox
```

L'avvio dell'applicazione da terminale è utile soprattutto per un primo debug nel caso di malfunzionamento del lanciatore grafico, che invece non darebbe alcuna informazione aggiuntiva (l'applicazione non partirebbe e basta), mentre l'errore che viene segnalato in output è fondamentale per poter circostanziare il problema.

Autenticandoci come utenti, abbiamo delle restrizioni per quanto concerne la modifica o anche solo la visualizzazione dei file di sistema, proprio per evitare che, accidentalmente o meno, possa venire compromessa l'operatività della nostra stazione di lavoro.

Per interagire appieno occorre pertanto avere le credenziali del *superuser* (utente *root* che invece gode dei più ampi permessi).

Per diventare utente root occorre digitare al prompt:

```
$ su
```

e ci verrà richiesta la password assegnata all'utente root che, una volta inserita, ci

permette di avere quei permessi necessari per l'amministrazione del sistema.

Da notare che, per motivi di sicurezza, alla digitazione della password di autenticazione, non si vedrà il cursore e che il prompt, come detto in precedenza, riporta il simbolo del cancelletto (#).



Figura 10: autenticazione di root

Eccoci autenticati come root nell'ambiente dell'utente (l'autenticazione con l'importazione dell'ambiente di root sarà oggetto di prossimi approfondimenti).

(fine prima parte)

Servizi di sistema (demoni)

Gabriele Trombini (mailga@fedoraonline.it)

(Parte prima) I servizi in Fedora - introduzione

Cominciamo a dare una occhiata al sistema di gestione dei servizi che recentemente ha cominciato a sostituire *SysVinit* ed *Upstart* nell'avvio e nella gestione dei servizi.

I collaboratori italiani del team di traduttori del Fedora Project hanno preparato la pagina wiki <https://fedoraproject.org/wiki/Features/systemd/it> dove sono ben spiegati i vantaggi di questo sistema di gestione della sessione e del sistema.

Facendo riferimento alla pagina del wiki <http://fedoraproject.org/wiki/Systemd>, dove troviamo dettagliatamente descritte le funzionalità, tra gli altri vantaggi notiamo che *systemd* velocizza l'avvio attivando processi in parallelo dando l'ascolto ai sockets prima di assegnare loro il servizio.

Il sistema *init* provvedeva a creare prima tutti i sockets e poi tutti i servizi, creando, in caso di dipendenza di un servizio rispetto ad un altro non ancora avviato, l'eventuale stallo del sistema o il mancato avvio del servizio stesso.

Inoltre esso si appoggia a D-Bus per l'avvio dei servizi on-demand, permette un recupero del sistema in caso di problemi (previo snapshot), si occupa del mount e dell'automount dei device (fstab può essere utilizzato come configurazione extra, indicando a *systemd* di monitorarlo), mantiene compatibilità con i sistemi precedenti e può essere configurato, praticamente, a piacimento.

Perciò, di fatto, a partire da Fedora 16 la nostra distribuzione ha introdotto in maniera massiva *systemd* per l'avvio e la gestione dei servizi.

Quali sono i vantaggi li abbiamo visti, seppur sommariamente (per approfondimenti dovremo studiarci le pagine web citate in precedenza), ma quello che più ci preme è la loro gestione.

Per facilitare il compito possiamo installare *systemadm* che ci fornisce una GUI con dei comandi minimali per l'avvio/arresto/riavvio dei servizi, per esaminarli e per effettuare lo snapshot di cui sopra:

レナート Wunschkonzert, Ponyhof und Abenteuerspielplatz پنارت

Fri, 30 Apr 2010

Rethinking PID 1

If you are well connected or good at reading between the lines you might already know what this blog post is about. But even then you may find this story interesting. So grab a cup of coffee, sit down, and read what's coming.

This blog story is long, so even though I can only recommend reading the long story, here's the one sentence summary: we are experimenting with a new init system and it is fun.

[Here's the code.](#) And here's the story:

Process Identifier 1

On every Unix system there is one process with the special process identifier 1. It is started by the kernel before all other processes and is the parent process for all those other processes that have nobody else to be child of. Due to that it can do a lot of stuff that other processes cannot do. And it is also responsible for some things that other processes are not responsible for, such as bringing up and maintaining userspace during boot.

Historically on Linux the software acting as PID 1 was the venerable sysvinit package, though it had been showing its age for quite a while. Many replacements have been suggested, only one of them really took off: [Upstart](#), which has by now found its way into all major distributions.

As mentioned, the central responsibility of an init system is to bring up userspace. And a good init system does that fast.

Figura 11: pagina del progetto (<http://0pointer.de/blog/projects/systemd.html>)

```
# yum install systemd-gtk
```

e per avviarlo:

```
$ systemadm
```

Inutile dire che utilizzando il terminale possiamo fruire dei vantaggi della velocità di esecuzione uniti alle opzioni dei comandi.

Il tool testuale è *systemctl*, il cui utilizzo completo può essere studiato nelle pagine man:

```
$ man systemctl
```

Questo comando permette di effettuare tutte le principali operazioni di cui un

utente potrebbe avere bisogno e la sintassi è molto semplice e, in qualche caso, ricorda l'utilizzo del comando “service”, con il quale mantiene una certa compatibilità prima di essere totalmente abbandonato in futuro.

- verificare i servizi

```
$ systemctl list-units --type=service
```

- verificare lo stato di un servizio

```
$ systemctl status nome.service
```

- informazioni approfondite sullo stato di un servizio attivo

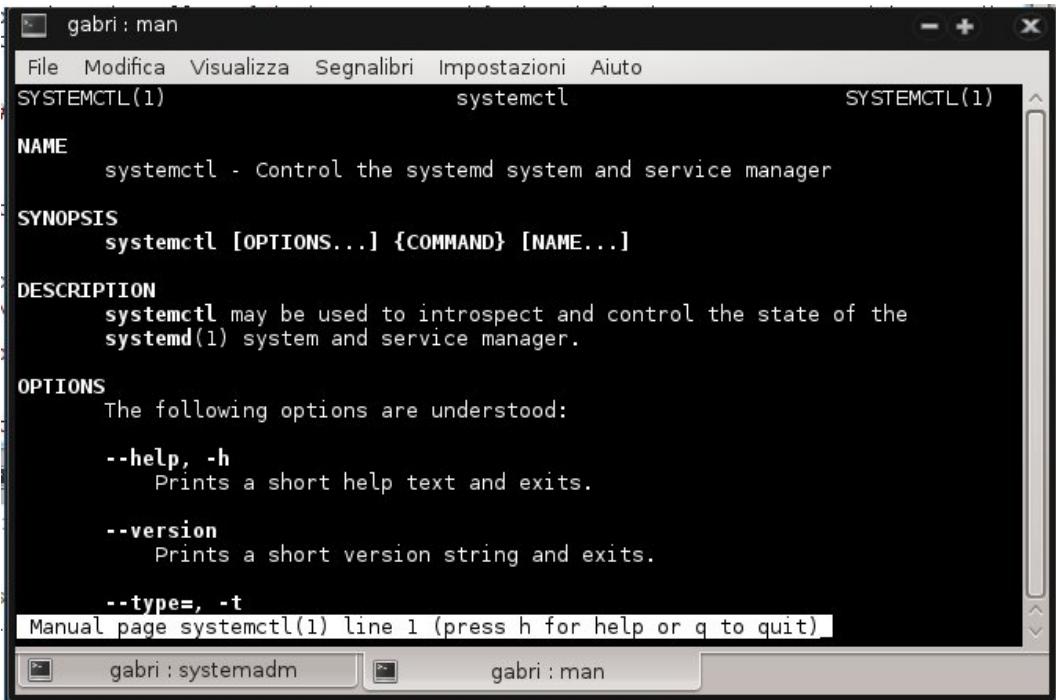


Figura 12: pagina del manuale di systemctl

```
$ systemctl is-active nome.service
```

- abilitare un servizio all'avvio

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

- disabilitare un servizio all'avvio

```
# systemctl disable nome.service
```

- avviare un servizio

```
# systemctl start nome.service
```

- fermare un servizio

```
# systemctl stop nome.service
```

- riavviare un servizio

```
# systemctl restart nome.service
```

- “uccidere” un servizio

```
# systemctl kill nome.service
```

Cercando di andare un po’ in profondità, possiamo vedere che le directory di riferimento di *systemd* sono fondamentalmente due:

- /etc/systemd
- /lib/systemd

Il contenuto della directory `/etc/systemd` ha la precedenza tra i due, ma entrambi contengono i servizi disponibili.

Ecco il file di configuraizone di mysqld:

```
[Unit]
Description=MySQL database server
After=syslog.target
After=network.target

[Service]
Type=forking
User=mysql
Group=mysql

ExecStartPre=/usr/libexec/mysqld-
prepare-db-dir
# Note: we set --basedir to prevent
probes that might trigger SELinux
alarms,
# per bug #547485
ExecStart=/usr/bin/mysqld_safe --no-watch
--basedir=/usr
ExecStartPost=/usr/libexec/mysqld-
wait-ready $MAINPID

# Give a reasonable amount of time for
the server to start up/shut down
TimeoutSec=300

# We rely on systemd, not mysqld_safe,
# to restart mysqld if it dies
Restart=always
[Unit] [Install]
```

WantedBy=multi-user.target

Qualora volessimo creare un servizio, il file di configurazione deve essere inserito all'interno di `/lib/systemd`, e poi collegato in `/etc/systemd` così che alla variazione del primo, corrisponda la variazione del secondo.

Per fare in modo che systemd venga a conoscenza del nuovo file:

```
# systemctl systemctl daemon-reload
```

Per abilitarlo all'avvio, basta quindi usare l'opzione `--enable` di cui abbiamo già parlato.

L'argomento merita un approfondimento maggiore (e per taluni aspetti ancora da scoprire) rispetto a quanto è possibile fare in un unico numero, tuttavia per un primo approccio al nuovo sistema di gestione dei servizi questo articolo può essere sufficiente.

Systemd verrà probabilmente ripreso nei prossimi numeri ad un livello di dettaglio più analitico, con esempi e con la chiamata in causa di *D-Bus*.

(fine prima parte)

Dracut

Gabriele Trombini (mailga@fedoraonline.it)

L'initramfs all'avvio di Fedora

Come sappiamo initrd e initramfs (per comodità diciamo che quest'ultimo è la versione attuale del precedente) non sono altro che dei file systems temporanei, che si collocano nella memoria, usati per caricare e montare il file system reale nel sistema.

Per compatibilità con un sistema generico, vengono inseriti al boot parecchi moduli di periferica e di filesystem ed il kernel, aumentando nelle dimensioni e nella quantità di hardware riconosciuto, impiega più tempo a caricarsi perciò si è reso necessario trovare soluzioni che mantengano un tempo ragionevole di start up.

A questo scopo è stata introdotta una fase iniziale di root file system in user space che si occupa di rilevare hardware, configurazioni, moduli prima di montare il file system root della macchina.

Difatti oggi la sequenza boot ordinariamente utilizzata è in due stages:

- lancio e caricamento kernel con tutti i suoi moduli built-in ovvero i moduli integrati che sono indispensabili al corretto funzionamento di una macchina base

- lancio di initramfs. Esso è di fatto un micro sistema in ram con le directory *bin - dev - etc - lib - tmp - var* etc... che provvede al caricamento non solo dei moduli atti al montaggio dei filesystem e quindi del filesystem root, cosa fondamentale, ma anche al montaggio di altri moduli quali ad esempio gli alsa, quelli video (se necessario), con eventuali regole udev su determinati dispositivi, o altri moduli o performance che l'utente (esperto) decide di caricare subito con la modifica di */etc/dracut.conf* con l'inserimento di regole specifiche in */etc/dracut.conf.d/*.

Dracut è quindi lo strumento che permette alla initial ramdisk di caricare solo quello che è necessario per l'avvio, creando una initramfs adatta al proprio sistema.

Tutto ciò avviene in automatico ma, qualora si avessero delle necessità particolari dovute ad installazioni successive di hardware o a moduli non interpretati dal sistema, è possibile utilizzare la linea di

comando per effettuare le operazioni di creazione, variazione, verifica e manutenzione della initramfs.

Il comando generico per poter creare un nuovo file system iniziale è:

```
# dracut /boot/initramfs-$(uname -r).img $(uname -r)
```

che viene utilizzato, ad esempio, dopo l'installazione dei driver grafici per poterli inserire già nel file system iniziale permettendo una corretta visualizzazione.

Come sempre in questi casi occorre sempre leggere il manuale dell'applicazione:

```
$ man dracut
```

all'interno del quale possiamo reperire le istruzioni e le opzioni per creare una immagine ramdisk iniziale integrando o escludendo moduli, filesystems, file di configurazione (compreso quello di default di dracut che è */etc/dracut.conf*), partizioni e altro.

E' anche previsto un file di log, conseguente all'utilizzo di dracut, che trova posto all'interno della directory */var/log*, sotto il nome di *dracut.log*.

```
$ cat /var/log/dracut.log
```

E' possibile inserire i parametri anche da linea di comando come descritto nella pagina man:

```
$ man dracut.kernel
```

dove vediamo anche opzioni da passare al kernel all'avvio; conoscete ai più attenti perchè utilizzate anche nel grub.cfg di Fedora.

Può essere interessante guardare all'interno del file della initramfs utilizzando il tool *lsinitrd* che ne permette l'ispezione:

```
# lsinitrd /boot/initramfs-$(uname -r).img | more
```

oppure verificare la presenza e contenuto di uno specifico file:

```
# lsinitrd /boot/initramfs-$(uname -r).img run/initramfs/shutdown
```

I file di configurazione di Dracut, in Fedora, devono essere posizionati all'interno della directory */etc/dracut.conf.d/*, mentre il file da poter adattare alle proprie esigenze, commentandone e/o decommentandone le opzioni per meglio raffinare l'avvio, è */etc/dracut.conf*.

```
$ man dracut.conf
```

Quanto inserito in */etc/dracut.conf.d/*, in ordine numerico, va a sovrascrivere le

impostazioni di `/etc/dracut.conf`, che invece è il file di prima lettura di dracut.

La particolarità di Dracut è data dal fatto che, proprio per adattare la singola macchina alle proprie necessità, si possa personalizzare la initramfs.

Utilizzando l'opzione `-include`, infatti, si possono inserire all'interno della initramfs i propri file di utilità particolare, ad esempio con il comando:

```
# dracut -include /bin/cat  
/ run/initramfs/usr/sbin  
initramfs-cat.img
```

è possibile inserire il comando cat all'interno della directory `run/initramfs/usr/sbin` della initial ramdisk.

L'opzione `-install` consente di poter specificare diversi eseguibili che dovessimo ritenere utili all'avvio, rilevando anche le relative dipendenze:

```
# dracut -install 'nmap ssh' initramfs-  
rete.img
```

Vale la pena, anche, di ricordare che Dracut ha accesso ad una CLI, da dove si può cercare di fare un avvio manuale del sistema.

Togliendo `rhgb` e `quiet` ed aggiungendo `rdshell` alle opzioni del kernel, si avrà a

disposizione una shell con la quale cercare di eseguire il boot (per poi, magari, ricostruire la initial ramdisk una volta autenticati).

Tra i parametri utili per una analisi del sistema avremo a disposizione:

- `rdshell`: come detto, inserendo questo comando si avrà accesso ad una shell di Dracut;
- `rdinitdebug`: per un livello di debug più approfondito, con stampa dei comandi;
- `rdbreak=[pre-udev / pre-mount / mount / pre-pivot //]`: aggiungendo questo parametro, la shell si fermerà prima di superare il punto dato come argomento;
- `rdudevinfo`: imposta udev ad un livello informativo massimo;
- `rdudevdebug`: imposta udev ad un livello di debug;
- `rdnetdebug`: verifica le connessioni network, l'output viene inserito in `/tmp`.

Questa breve panoramica non ha certo la pretesa di essere esaustiva ma quanto detto in questo articolo può concedere ampi spazi di approfondimento ai lettori.

Creazione di RPM

Gabriele Trombini(mailga@fedoraonline.it)

(Parte prima) RPM - introduzione

Con l'acronimo *rpm* (RedHat Package Manager) si identifica sia il formato dei pacchetti usati da Fedora che l'utility per la loro gestione.

A dire il vero, seppur nato da Red Hat, questo formato riscosse fin dall'inizio un grande successo (al pari del formato .deb usato da debian e derivate), tant'è che ancora oggi viene utilizzato da altre distribuzioni (come Mandriva e Opensuse, inizialmente derivate da Red Hat) per l'installazione degli applicativi.

Lo stesso Fedora Project ne ha adottato lo standard come riferimento.

Malgrado vi siano diversi tools (come alien o checkinstall, anche se quest'ultimo non ha avuto nuove versione dal 2009) che permettono la creazione di *rpm*, con questo articolo vogliamo seguire le specifiche del Fedora Project riguardanti la creazione di un pacchetto dal punto di vista di un collaboratore packager, così come specificato all'indirizzo <http://fedoraproject.org/wiki/Packaging:Guidelines>, tralasciando quindi le modalità di gestione, manutenzione e ispezione dei pacchetti mediante l'utilizzo del comando rpm e

relative opzioni

Le linee guida sono molto dettagliate ed è consigliabile seguirle analiticamente in modo particolare qualora si volesse contribuire al progetto o mantenere i pacchetti di Fedora.

Preparazione

Come prima cosa è indispensabile che gli strumenti necessari alla corretta esecuzione delle operazioni siano installati sul proprio computer:

```
# yum install @Strumenti\ di\ sviluppo
```

```
# yum install fedora-packager
```

Quest'ultimo comando installa come dipendenza anche l'utility *rpmdevtools* che ci consente di utilizzare il comando adatto alla creazione dell'albero per lo sviluppo:

```
$ rpmdev-setuptree
```

Al termine del comando possiamo controllare che nella home dell'utente in uso sia presente la cartella rpmbuild, che a sua volta contiene altre directory (al momento vuote).

```
[gabri@amministrazione ~]$ tree rpmbuild/
rpmbuild/
├── BUILD
├── RPMS
├── SOURCES
├── SPECS
└── SRPMS

5 directories, 0 files
[gabri@amministrazione ~]$
```

Figura 13: l'albero di sviluppo dei pacchetti

- SOURCES — dove devono essere inseriti i file sorgenti, le patch ed i file di icone;
- SPECS — dove devono essere inseriti i file .spec;
- BUILD — dove i sorgenti saranno scompattati ed il pacchetto costruito;
- RPMS — dove il processo build inserirà il pacchetto binario;
- SRPMS — dove il processo build inserirà il pacchetto sorgente.

A questo punto tutto è pronto per cominciare il nostro lavoro di packaging ma, se vogliamo essere conformi a quanto previsto dal Fedora Project, è bene che si legga in particolare le specifiche riguardanti:

1. nomenclatura (http://fedoraproject.org/wiki/Packaging:NamingGuidelines#Package_Naming_and_Versioning_Guidelines);
2. versione e release (http://fedoraproject.org/wiki/Packaging:NamingGuidelines#Package_Version);
3. aspetti legali (<https://fedoraproject.org>)

[ct.org/wiki/Packaging/Guidelines#Legal\);](http://ct.org/wiki/Packaging/Guidelines#Legal);)

4. licenza (<http://fedoraproject.org/wiki/Packaging:LicensingGuidelines>);

Oltre a quanto sopra, occorre tenere presente anche altri aspetti, di carattere più pratico, legati alla preparazione dei pacchetti per la nostra distribuzione.

- tutto il software presente deve essere libero ed opensource;
- le inclusioni devono essere inserite partendo dai sorgenti;
- non costruire i pacchetti utilizzando l'utente root (è possibile creare un utente per lo sviluppo);
- seguire le direttive di layout del FHS (Filesystem Hierarchy Standard) che definisce come devono essere distribuiti i file nel sistema e, se si dovesse avere necessità di creare nuove directory, è bene chiedere, motivando l'operazione, al Fedora Packaging Committee;
- non inserire file in /bin, /sbin, /lib o /lib64, bensì utilizzare /usr/bin, /usr/sbin, /usr/lib, and /usr/lib64 (è possibile usare dei link simbolici a queste directory);

Come già detto Il Fedora Packaging Committee può rispondere ad eventuali dubbi.

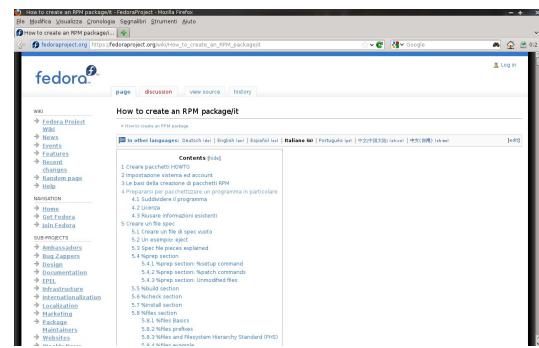


Figura 14: come creare un rpm, la pagina tradotta dal team italiano

Arrivati a questo punto abbiamo preparato il nostro sistema per il packaging e siamo pronti ad eseguire il processo secondo le indicazioni del Progetto Fedora.

Il passo successivo che vedremo nel prossimo numero, sarà quello di creare il file *.spec*, un fondamentale file di testo contenente le informazioni necessarie alla pacchettizzazione dell'applicativo.

(fine prima parte)

Il Kernel Fedora

Giuseppe Delvecchio (virus@fedoraonline.it)

(Parte prima) Kernel - introduzione

Ll giorno 15 agosto 1991 – Linus Torvalds scrive : “*Sto programmando un sistema operativo (gratuito e solo per hobby, non vuole essere grande e professionale come GNU) per cloni di AT 386 (486). È in preparazione da Aprile, e sta iniziando a funzionare...*”

Questa data può essere considerata la data di nascita del kernel linux.



Figura 15: Linus Torvalds

Cosa è un kernel?

Esso è il nucleo software fondante di un sistema e fornisce tutte le funzioni essenziali per la gestione e il funzionamento del sistema.

Gestisce la memoria, la cpu, l'Input/Output, la rete, i filesystem, i dispositivi periferi-

ci, i processi ad essi collegati, le comunicazioni tra processi, le priorità di questi ultimi, e molto altro ancora.

Tutto questo è solitamente trasparente all'utente medio che non si accorge di (quasi) nulla.



Figura 16: Tux

Uno dei primi problemi che Linus ha dovuto affrontare è stata la scelta del tipo di kernel da costruire. Egli costruì un kernel monolitico ovvero un kernel che sia un blocco di codice unico omnicomprensivo; ben presto però questo tipo di approccio rivelò i propri limiti, in quanto le dimensioni del kernel diventavano sempre più grandi (troppo) e il suo sviluppo complicato da gestire. La soluzione fu trovata usando un metodo diverso.

Pur mantenendo la struttura monolitica, furono creati moduli kernel che potevano essere caricati nel sistema in funzione delle esigenze, questi moduli che ge-

stiscono determinati dispositivi, processi, filesystem, possono essere inclusi o meno nel kernel mediante opportuni comandi oppure inclusi in esso all'atto della compilazione.

Questo approccio è chiamato kernel modulare.

Quindi sostanzialmente il kernel attuale ha la possibilità di avere moduli interni che fanno blocco unico e/o vengono caricati al momento del boot anche in due momenti successivi (kernel e initramfs), moduli caricabili successivamente al boot (operazione a cui provvede systemd oppure udev), e moduli che vengono totalmente esclusi, quindi mai caricabili a meno di non ricompilare il kernel.

I kernel rilasciati dalle varie distribuzioni linux differiscono proprio, anche se non esclusivamente, in questo; ognuna di esse ha fatto delle scelte precise di cosa inserire nel blocco unico, cosa rendere modulo caricaabile, cosa escludere.

Spieghiamo brevemente il significato delle sigle presenti nel nome di un pacchetto rpm del kernel installato su fedora, prendendo per esempio il *kernel-PAE-3.1.9-4.fc16.i686*:

- la sigla PAE indica che si tratta di un kernel che supporta l'estensione dell'indirizzo fisico per la memoria,

ovvero è in grado di rilevare ed utilizzare una ram maggiore di 4 Gb, questa versione PAE esiste solo per la 32 bit, ovviamente.

- 3.1.9 – è la nuova numerazione semplificata introdotta da Linus Torvalds il 29 maggio 2011 in occasione del ventesimo anniversario del kernel linux:

- 3 — è la major version che corrisponde alla “vecchia” 2.6;
- 1 — enumera i nuovi rilasci;
- 9 — enumera i rilasci di aggiornamenti di sicurezza e bug fix sul rilascio precedente;
- 4 — dopo il trattino seguono valori numerici tipici dei kernel compilati con patch oppure bug fix specifici introdotti dagli sviluppatori kernel della nostra fedora.
- fc16 — indica per quale versione fedora è stato rilasciato
- i686 — indica l'architettura del kernel, si tratta di una 32 bit sesta generazione delle cpu x86 intel e compatibili.

Una descrizione grafica di un kernel potrebbe essere la seguente: http://www.makelinux.net/kernel_map/.

La grande complessità di questa mappa ci spaventa ma ci fa rendere conto del lavoro che è stato eseguito per costruire un

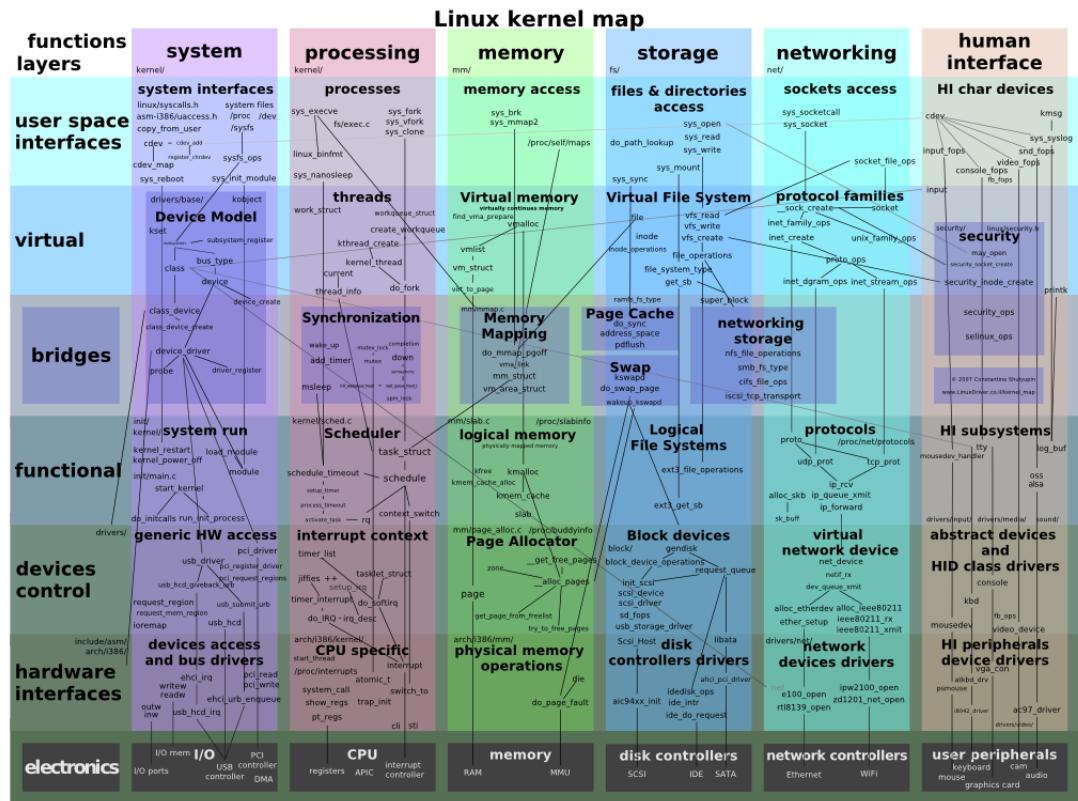


Figura 17: mappa del kernel (fonte: <http://it.wikipedia.org/>)

kernel.

Quando è necessario compilare un kernel?

1. per far funzionare un componente particolare che non è gestito dal kernel che si utilizza, il cui modulo esiste ma non è stato incluso;
2. per utilizzare l'ultima versione kernel se non è disponibile il pacchetto precompilato;
3. per fare test “particolari” al proprio sistema;

4. a scopi “didattici”.

(fine prima parte)

Keyserver e generazione di chiavi GPG

Andrea Veri (averi@fedoraproject.org)

Vediamo come usare il keyserver Fedora per la generazione di chiavi GPG

Da pochi giorni ha debuttato in Fedora un interessante servizio denominato Keys (raggiungibile all'indirizzo <http://keys.fedoraproject.org>), si tratta, come si può ben intendere dal nome usato, di un server adibito allo storage di chiavi OpenPGP.

Questa tipologia di servizio permette l'invio e la ricezione di specifiche chiavi di cifratura PGP, le quali vengono selezionate dall'utente a seconda della sua necessità di identificare o di firmare la chiave di un altro soggetto.

Ma come funziona esattamente GPG?

Il meccanismo risulta essere alquanto semplice ed è, inoltre, similare a quello utilizzato dalla firma digitale, forma di autenticazione dei documenti telematici che oramai sta prendendo piede anche in Italia.

GPG, quindi, cifra un determinato messaggio utilizzando una coppia di chiavi

(pubblica e privata) generate dall'utente. Il soggetto che riceverà tale messaggio cifrato non dovrà fare altro che verificare l'autenticità della firma e verificarne l'attendibilità. Onde evitare la divulgazione di chiavi false o fittizie e favorire l'associazione chiave pubblica - utente, si è giunti a dare particolare rilevanza al cosiddetto Web of Trust (o più comunemente WoT).

Tale concetto è di facile comprensione: si pensi ad un determinato utente che durante un meeting incontra altri individui e dopo aver provveduto ad identificarli tramite un documento d'identità e tramite la verifica della loro impronta digitale (o più comunemente fingerprint), firma la loro chiave PGP con la propria, certificandone l'autenticità della provenienza.

Questa semplice ma efficace operazione aiuta a rafforzare la rete di fiducia e di conseguenza la stessa credibilità dell'utente stesso.

Detto ciò seguirà una brevissima guida che coprirà principalmente tre tematiche:

1. la creazione di una chiave PGP
2. la firma di una chiave PGP altrui
3. la cifratura di un file decifrabile solamente da una determinata chiave

Punto 1. Creazione di una chiave PGP.

- 1a. Installiamo il software di riferimento.

```
$ sudo yum install gnupg
```

- 1b. Apriamo un terminale e creiamo la nostra chiave di cifratura.

```
$ gpg --gen-key
```

Le impostazioni di default, ovvero quelle che otterrete con un semplice 'Invio' ad ogni richiesta di input da parte di GPG, sono solitamente quelle consigliate ad un utente normale.

È però consigliabile impostare una data di scadenza alla chiave nel qual

caso perdeste o cadesse in mani sbagliate la vostra chiave privata.

- 1c. Appena generata la chiave, inviatela al Keyserver di Fedora, rendendola così pubblicamente disponibile.

```
$ gpg --send-keys KEYID --keyserver hkp://keys.fedoraproject.org
```

Per visualizzare il vostro KEYID, vi basterà digitare il seguente comando:

```
$ gpg --list-secret-keys
```

Il KEYID sarà la prima linea numerica che vedrete, ad esempio: sec 4096R/93ECD6F8 2012-01-31

Punto 2. Firma di una chiave PGP altrui.

- 2a. Otteniamo la chiave che vogliamo firmare dal Keyserver di riferimento.

```
$ gpg --recv-key KEYID --keyserver hkp://keys.fedoraproject.org
```

l'importazione è avvenuta correttamente:

- 2b. Firmiamo la chiave.

```
$ gpg --sign-key KEYID
```

- 2c. Inviamola nuovamente al keyserver con appresso la nostra firma.

```
$ gpg --send-key KEYID
```

Punto 3. Cifratura di un file decifrabile solamente da una determinata chiave.

- 3a. Otteniamo la chiave di cifratura dell'utente che poi dovrà effettivamente decifrare il nostro messaggio.

```
$ gpg --search-keys 'nomeutente@example.com' --keyserver hkp://keys.fedoraproject.org
```

- 3b. Dopo aver selezionato la chiave corretta dalla lista, verificate che

```
$ gpg --list-keys
```

- 3c. Infine, cifrate il messaggio:

```
$ gpg --encrypt --recipient 'nomeutente@example.com' nomefile.txt
```

- 3d. Inviate via mail il file appena generato ('nomefile.txt.gpg') all'utente da voi designato, che potrà, quindi, decifrarlo:

```
$ gpg --output nomefile.txt --decrypt nomefile.txt.gpg
```

In questo caso non servirà importare nessuna chiave dal momento in cui il file di cui sopra è stato cifrato utilizzando la vostra chiave pubblica.

Fedora 17

Robert Mayr (robyduck@fedoraonline.it)

Fedora 17 - una panoramica sulle novità

E' notizia di pochi giorni fa l'annuncio di *Dennis Gilmore* riguardante l'uscita della versione *alpha* di Fedora 17 "Beefy Miracle", per la quale sono previste le caratteristiche che possiamo vedere complete a questo indirizzo: http://fedoraproject.org/wiki/Releases/17/FeatureList#Fedora_17_Accepted_Features.

La roadmap per il rilascio della versione *stable* si può trovare alla pagina <http://fedoraproject.org/wiki/Schedule>.

Key Milestones

2011-11-08	Fedora 16 Release
	Planning & Development Begins
2012-01-24	Feature Submission Deadline
2012-02-07	Feature Freeze--Planning & Development Ends
2012-02-07	Branch Fedora 17 from Rawhide-- Branch Freeze
2012-02-14	Software String Freeze
2012-02-14	Alpha Change Deadline
2012-02-28	Alpha Release
2012-03-13	Software Translation Deadline
2012-03-20	Beta Change Deadline
	Features 100% Complete
2012-04-03	Beta Release
2012-04-23	Final Change Deadline
2012-04-24	Compose 'Final' RC
2012-05-08	Fedora 17 Final Release

Figura 18: roadmap per Fedora 17 *Beefy Miracle*

Come al solito la lista delle features accettate è molto differenziata, ma tra queste possiamo rilevarne alcune molto importanti sia per gli utenti finali che per chi è interessato al sistema ed allo sviluppo:

- Gnome 3.4.
- KDE 4.8.
- Gimp 2.8.
- Kernel 3.3.
- GCC 4.7.x.
- PHP 5.4.
- Software rendering for gnome-shell.
- Move all to /usr.
- ABRT Backtrace Deduplication Service.
- Eucalyptus.
- Firewalld.
- KVM Live Block Migration.
- NetworkManager Hotspots.
- Riak.

Vediamone alcune più dettagliatamente:

Gnome 3.4

Tra le novità spicca, anche se ancora in fase di sviluppo e con qualche ostacolo di trademark, *Boxes* che sempificherà l'accesso ad altri sistemi operativi, siano essi fisici, virtuali o raggiungibili via rete. Sarà incluso anche l'accesso remoto, il tutto utilizzando la tecnologia Qemu. Nonostante lo sviluppo di virt-manager *Boxes* si pone l'obiettivo di mettere a disposizione dell'utente un'interfaccia più amichevole.

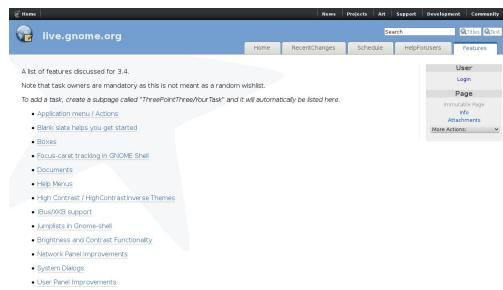


Figura 19: live.gnome.org

KDE 4.8

In questa versione, l'utilità *KDE Power Management System Settings* è stata aggiornata sia per quanto riguarda l'interfaccia utente che per il layout. Sono stati introdotti miglioramenti per *Dolphin*, *Gwenview*, *KMail* (non si dovranno riscontrare i problemi di sicurezza e stabilità delle versioni precedenti), *Kate* e molto altro.



Figura 20: kde.org

Kernel 3.3

Uno dei cambiamenti nella versione 3.3 del kernel è dovuto all'inserimento di un nuovo meccanismo per regolare la dimensione dei filesystems ext4; questo meccanismo dovrebbe ora lavorare più velocemente. Miglioramenti sono stati apportati anche per l'audio HDMI su tipologie di chipset AMD e Nvidia. Sono stati raggruppati, inoltre, alcuni dispositivi ethernet in un unico dispositivo virtuale. E' stato inserito il supporto a Open vSwitch. Il filesystem btrfs è stato ottimizzato per operazioni con volumi RAID ed è inserita in via sperimentale la verifica dell'integrità dei dischi in alcune operazioni.



Figura 21: kernel.org

Software rendering for gnome-shell

Gnome-shell sarà in grado di funzionare anche con gran parte di quelle schede grafiche che non supportano l'accelerazione 3D, un enorme passo avanti per quanto riguarda l'usabilità del DE di default.

Move all to /usr

Come già visto nell'articolo relativo ai pacchetti rpm, è stato stabilito lo spostamento di tutte le /lib e /bin sotto la directory /usr. Di fatto, le directory /lib, /lib64, /bin e /sbin conterranno soltanto dei link simbolici ai file che si troveranno sotto la directory /usr.

ABRT Backtrace Deduplication Service

Durante l'inserimento di un nuovo bug nel RedHat Bugzilla, ABRT controllerà se ne è già stato aperto uno per lo stesso problema, onde evitare duplicati.

Eucalyptus

Eucalyptus è un software cloud computing per l'utilizzo privato, che permette di utilizzare le infrastrutture esistenti per creare clouds per computer, memoria di massa e rete. Nelle novità del cloud computing si inserisce anche il toolkit Open-Nebula.

Firewalld - default firewall solution

Il nuovo firewall di default sostituisce i servizi iptables, iptables-ipv6 e ebtables.

KVM Live Block Migration

KVM avrà la possibilità di migrare immagini del disco, mentre questo è in funzione (live block migration). oVirt è ancora in fase sperimentale, ma dovrebbe essere in grado di virtualizzare server, fare migrazioni live block e molto altro.

NetworkManager Hotspots

Questa aggiunta mette NetworkManager nelle condizioni di funzionare come Access Point.

Riak

Il database in clustering NoSQL viene incluso nei repo di Fedora. Riak avrà anche una consolle di amministrazione dei cluster, che rende più agevole la loro gestione.

Per concludere, un'ultima notizia: il filesystem di default rimarrà ext4 (il quale avrà il supporto per gestire dischi oltre i 16TB) in quanto btrfs è slittato, per alcune incompatibilità e restrizioni di anaconda, almeno alla versione 18.

Il resto lo scopriremo all'uscita di **Fedor**a 17 “Beefy Miracle” l' 8 Maggio 2012, salvo ritardi.

FOLIO - IL WEBMAGAZINE DI FEDORAONLINE (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.