



I programmi e le soluzioni che permettono di offrire direttamente in un'applicazione gli stessi contenuti e la funzionalità di un sito Web.

Un browser Linux monouso, per ogni sito Web. È la strada giusta?

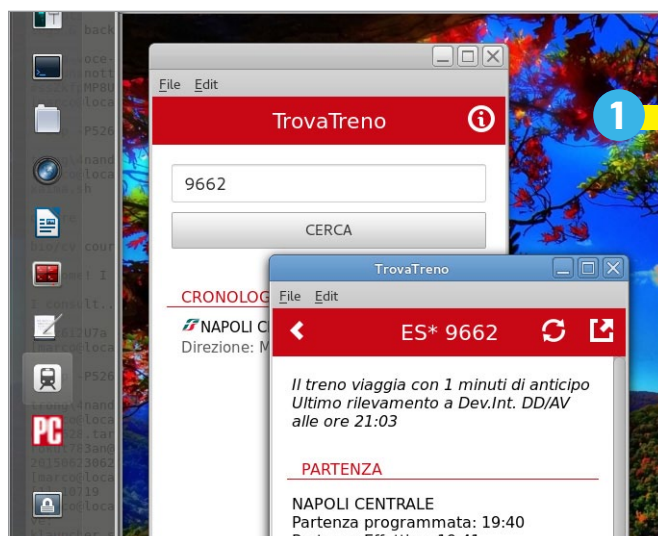
La migliore introduzione possibile all'argomento di questo mese potrebbe venire da Xkcd (<http://xkcd.org>), il fumetto online di Randall Munroe che da anni è un cult fra gli appassionati di informatica e cultura hacker in generale. Nella sua vignetta 1367, intitolata "Installing", qualcuno dice che ormai installare app negli smartphone è talmente facile e rapido che forse potremmo finalmente smettere di preoccuparcene noi. Molto meglio, invece, uno smartphone che ogni volta riscarichi e reinstalli da solo, e solo finché ne abbiamo effettivamente bisogno, quei pochi dati e funzioni che ci servono in ogni momento. A questa osservazione la voce fuori campo dell'autore replica "sembrava un'idea intelligente, finché non mi sono reso conto di aver inventato... le pagine Web".

Distinguere o meno fra "app" e pagine Web, o fra app e programmi software, è un tema cruciale, perché mette in discussione il modo stesso di presentarsi o utilizzare servizi online: cos'è meglio, installare solo un browser tuttofare e creare siti e servizi Web che funzionino abbastanza bene su qualsiasi browser del genere, qualunque sia il dispositivo su cui è installato? Oppure sostituire ogni "sito" con varie app, cioè offrire gli stessi contenuti o servizi con tanti programmini da scaricarsi a parte, ognuno ottimizzato per una sola piattaforma hardware e software?

Fino a poco tempo fa il problema si poneva solo con i terminali mobili, ma la grande popolarità di questi ultimi ha portato a sperimentare lo stesso modello anche sui desktop tradizionali. Al momento questi tentativi non hanno ancora seriamente insidiato né i tradizionali "programmi software nativi", né l'uso da semplici browser di servizi come Gmail o Google Docs, ma non demordono. Questo mese descriveremo perciò alcune di queste soluzioni Open Source, temporanee ma già utilizzabili senza alcuna particolare competenza, per avere "app" su Linux o altri sistemi operativi. Il mese prossimo, invece guarderemo a due proposte più ambiziose, che Linux come lo conosciamo vorrebbero addirittura sostituirlo.

UN PROGRAMMA, UNA APP, UNA PAGINA WEB?

La risposta a queste domande è facile solo se si risponde con l'atteggiamento sbagliato, cioè pensando a *dove*, finora, abbiamo incontrato una cosa o l'altra sui nostri schermi. Se è così, un programma è software permanentemente presente in un computer desktop o portatile, che si "lancia" facendo clic su qualche menu di sistema; una pagina Web è un insieme di contenuti, o un qualche servizio, accessibili dall'interno di quei programmi tuttofare per computer o smartphone, chiamati *Web browser*, che li scaricano temporaneamente, su richiesta; e una app è sempre software, o al limite un *singolo* sito Web reimballato per sembrare software, che

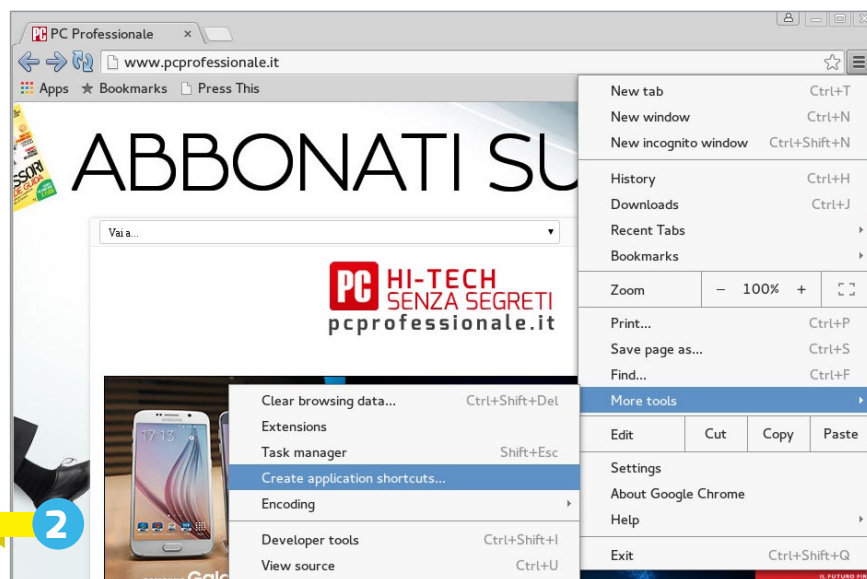


Un browser su misura per per PC Professionale e il TrovaTreno per Firefox (a destra) diventano "app" accessibili dai menu di sistema come Libre Office, digiKam e tanti altri programmi "normali".

si installa permanentemente come un programma, ma solo su smartphone o tablet. Questo, fino a oggi. Ma le tecniche e i prodotti che incontrerete nei prossimi paragrafi cercano proprio di eliminare queste differenze, pensando non a *dove* abbiamo incontrato finora certe cose, ma a *cosa* ci facciamo, e al modo più efficace di farlo.

Molti, infatti, sostengono che *qualsiasi* "sito Web" su cui si *fanno* cose, anziché limitarsi a fruire di testi o contenuti in maniera sostanzialmente passiva, dovrebbe diventare una "app" anche sui normali computer. Qualcosa, cioè, che è immediatamente disponibile e funziona sempre allo stesso modo, anche quando non si è connessi a Internet. Ma, soprattutto, qualcosa che non sembri più un oggetto *esterno* guardato attraverso un vetro (il browser), ma sia completamente integrato col desktop, dall'uso delle periferiche hardware all'accesso a file locali e notifiche di sistema. E che sia accessibile da menu di sistema o icone sullo schermo, non dalla barra degli indirizzi di un browser.

Questo approccio non pretende certo di eliminare completamente il World Wide Web o i browser tradizionali, né potrebbe mai farlo, nel futuro prevedibile. Basta un'occhiata alle centinaia o migliaia di bookmark presenti nei browser di qualsiasi utente medio di Internet per capire che sarebbe impossibile convincere tutti i rispettivi webmaster a convertire i propri siti in "app". Nonostante questo, oggi è già possibile su desktop trasformare praticamente qualsiasi sito Web in qualcosa che non è una vera app, ma all'atto pratico offre un aspetto analogo, almeno per certe applicazioni. Vediamo come farlo.



Trasformare qualsiasi sito in una "app" separata con Chrome richiede solo un paio di clic.

SSB, I BROWSER MONOUSO: PRO...

La soluzione più semplice per trasformare, su desktop Linux e non solo, qualsiasi sito Web in qualcosa che *sembra* una "app" si chiama *Site Specific Browser* (Ssb). Questa semplice procedura è disponibile in diverse forme più o meno dal 2010, senza aver mai goduto di grande popolarità. Nel 2015 è però tornata sotto le luci della ribalta Linux con l'arrivo di Peppermint Six, di cui ripareremo fra poco.

Gli Ssb non sono altro che copie di un normale browser, riconfigurate per partire con meno interfaccia possibile in vista (ovvero senza barra degli indirizzi, pulsanti e menu) e *sempre con lo stesso sito*, magari anche con bookmark e cookie distinti da quelli di default dello stesso browser. Quest'ultimo, ovviamente, rimane sempre a disposizione

sul computer, per navigazione in tutto il resto del Web.

Creare un Ssb è particolarmente utile per siti che si usano molto spesso e/o per operazioni particolarmente importanti. Trattare quelle pagine come tutte le altre potrebbe rallentarne troppo l'uso, o creare problemi, se *altri* siti, in altre schede o finestre della stessa istanza del browser, ne provocassero il crash. Oltre a stabilità e spazio su schermo, gli Ssb possono anche far risparmiare tempo, in vari modi. Il primo è l'integrazione con il desktop, ovvero l'aggiunta dei vari Ssb ai menu di sistema mostrata nella figura 1. Ci sono tanti servizi online, come quello nel Trova Treno di Firefox sempre nella figura 1, che in certi momenti servono molto spesso, ma per pochi secondi alla volta. Quand'è così, aprirli e spegnerli con un clic nel menu di sistema può far perdere meno tempo che tenere sempre aperta una scheda

ECCO DUE SSB DA INTENDITORI

WEB/EPIPHANY E WEBBY

I browser che oggi vanno per la maggiore, su Linux e non solo, non sono certo gli unici con cui creare Ssb. Un po' per completezza e interesse storico, un po' per stimolare la voglia di provare applicazioni diverse, non possiamo non segnalare almeno Web e Webby.

Il primo (<https://wiki.gnome.org/Apps/Web>) non è altro che il browser fatto in casa del progetto Gnome, quello precedentemente chiamato Epiphany e prima ancora Galeon: se si vuole usare Web come Ssb basta fare clic sul menu di destra e scegliere "Salva come Web App". Webby (per Ubuntu, su <https://launchpad.net/webby-browser>) è un browser molto semplice fatto apposta per l'uso Ssb, con un pannello di configurazione simile a quello Ice di Peppermint: nella sua finestra si possono inserire l'indirizzo della pagina Web da trasformare e scegliere se gestire a parte oppure no i relativi cookie e password, e il risultato è un Ssb lanciabile sia dal cruscotto sia dal Menu Applicazioni standard di Unity.

del browser, da ritrovare ogni volta in mezzo a decine di altre. Il risparmio di tempo aumenta parecchio se si associa ogni Ssb a una combinazione di tasti veloci.

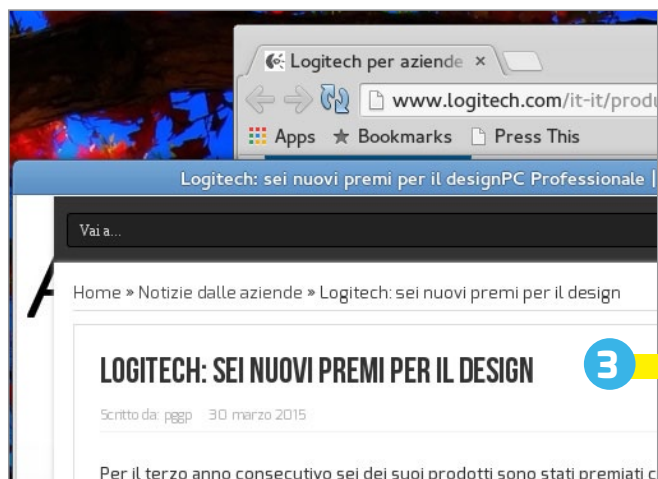
L'altro modo di risparmiare tempo con un Ssb è ridurre le distrazioni ed errori. Inserendo la propria casella webmail in un Ssb, per concentrarsi sulla posta da smaltire basta un clic per spegnere l'"altro browser", senza perdere tutte le sue schede e finestre, fossero pure decine, né rischiare di chiudere *anche* l'email! Ovviamente vale anche l'inverso: se la posta elettronica, o un gioco online, sono in un loro Ssb, se ne potranno chiudere le finestre senza alcun timore di chiudere *altre applicazioni Web*.

L'ultimo, non trascurabile vantaggio degli Ssb è la flessibilità di configurazione. Almeno in certi casi, si possono avere collezioni di bookmark, cookie o addirittura plugin (per aumentare le prestazioni) diverse da Ssb a Ssb.

..E CONTRO

La prima, ovvia conseguenza di usare gli Ssb, quindi più copie completamente indipendenti di un browser, è un maggior consumo di memoria.

Questo è inevitabile, tranne in un caso in cui potrebbe accadere proprio l'opposto. Se un certo sito, che serve spesso, manda alle stelle il consumo di memoria del browser se rimane sempre aperto in una sua scheda, rinchiuderlo in un Ssb lo terrebbe a portata di clic, ma senza fargli disturbare la navigazione. Gli Ssb possono anche interferire con l'uso di Internet se vengono configurati per *non* usare la stessa cronologia e gli stessi cookie del browser principale, perché è con quei piccoli file di testo che i vari siti ci riconoscono. Questo *potrebbe* sembrare un vantaggio,



L'Ssb di Pc Professionale è sempre lo stesso sito visitato con Chrome, ma senza l'interfaccia completa di quel browser (nello sfondo) che in questo caso sarebbe inutile.

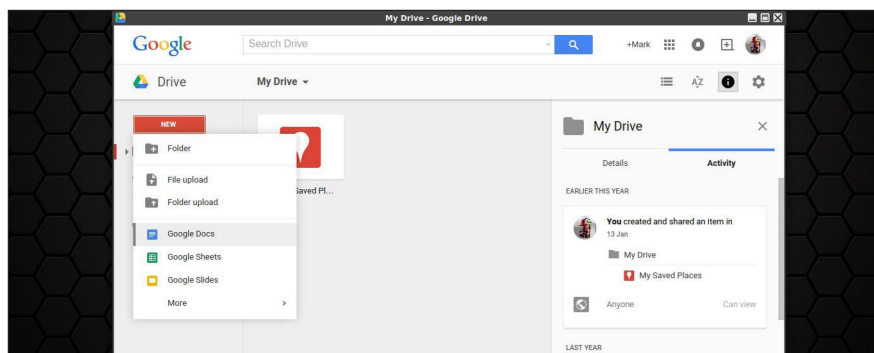
per navigare "anonimamente" su Internet. In realtà la protezione sarebbe minima, ma potrebbe costringere a reinserire user name e password di altri siti più spesso del necessario. Da questo punto di vista, i siti migliori da usare con Ssb sono quelli da cui non si andrebbe comunque da nessun'altra parte (per esempio un portale di giochi online). Sempre in tema di limitazioni, i sistemi Ssb più semplici non consentono installazioni condivise. Se cinque utenti dello stesso computer hanno bisogno dello stesso Ssb, ognuno di loro dovrà installare una sua copia. Non si può nemmeno escludere che, in modalità Ssb, certi plugin o estensioni non funzioneranno correttamente. Infine, c'è un'altra cosa che non è esattamente uno svantaggio degli Ssb ma, almeno sotto Linux, può complicare la vita. Anche quando un sistema per creare Ssb permette di aggiungerlo ai menu del sistema, non è detto che quella impostazione funzionerà automaticamente su *tutti* gli ambienti desktop o window manager che si potrebbero usare.

SSB FAI DA TE CON CHROME

Google Chrome può fornire Ssb, che chiama "scorciatoie per le applicazioni" (in inglese *Application Shortcut*) nella maniera semplicissima mostrata nella figura 2. Basta un clic sulla voce di menu corrispondente, dalla scheda in cui è già aperta la pagina da trasformare in Ssb. Le scorciatoie di Chrome possono avere ognuna delle estensioni diverse, o condividere quelle dell'installazione principale. Nel primo caso occorre installare in ogni Ssb le sole estensioni di cui avrà bisogno, nell'altro se ne può fare a meno collegandosi al proprio account Google ed effettuando la sincronizzazione della configurazione.

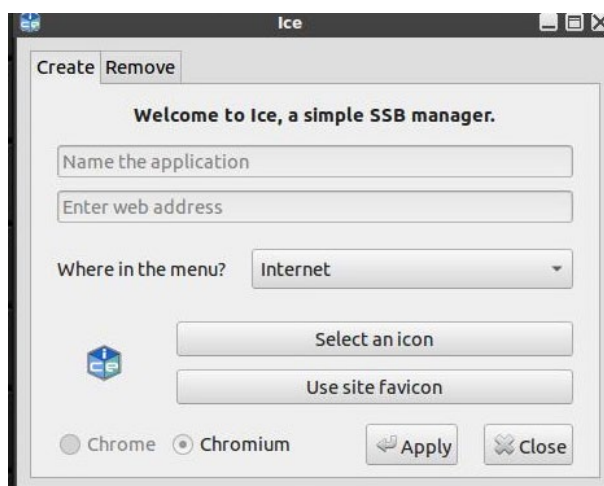
ICE

La caratteristica più originale e importante di Peppermint è la facilità con cui permette di aggiungere o rimuovere Ssb dal desktop. Questo compito è infatti assegnato a un'interfaccia grafica appositamente sviluppata, chiamata Ice. Nel suo pannello bastano pochi clic per impostare un Ssb su qualsiasi sito Web e inserire la voce corrispondente in qualsiasi sezione del menu di sistema. Volendo, si può assegnare come icona al nuovo Ssb l'icona predefinita ufficiale del sito ("favicon"), o una qualsiasi altra immagine presente nel computer. Cancellare un Ssb è ancora più facile, basta un clic sulla scheda "Remove" di Ice per selezionare cosa si vuole eliminare.



Google Drive impacchettato come "app" in Peppermint: quanta differenza, nella sostanza, c'è con la schermata d'avvio di Libre Office?

L'interfaccia Ice di Peppermint è appena più complicata della procedura per Ssb di Chrome, ma il risultato è completamente integrato nei menu di sistema.



I PROFILI E IL MERCATINO DI FIREFOX

Firefox, al momento, offre ai suoi utenti due tipi molto diversi di "app" o Ssb. Il primo è sostanzialmente equivalente, alla fine, alle scorciatoie di Chrome, ma molto più macchinoso: si può creare un profilo separato apposta per il sito da usare in modalità Ssb e poi aggiungerlo a mano ai menu di sistema. Quel profilo, ovviamente, potrà avere tutte le barre strumenti disabilitate, così come i suoi cookie, bookmark e così via. L'altro sistema, profondamente diverso, è molto più sofisticato ma anche, al momento, più limitato, perché dipende dalla buona volontà di sviluppatori esterni. Il Marketplace (mercato, vedi il box Risorse) di Firefox offre "applicazioni", sviluppate con tecnologie su cui torneremo nel prossimo numero, come il TrovaTreno nella figura 1. Queste applicazioni funzionano su qualsiasi piattaforma su cui gira Firefox e basta installarle su uno dei propri dispositivi per averle a disposizione anche sugli altri (se si ha un account Firefox).

PEPPERMINT, O GLI SSB DI SERIE

Peppermint (<http://peppermintos.com>) è, almeno a prima vista, l'ennesima distribuzione Linux clonata da Ubuntu. La versione Six, disponibile dall'estate 2015, è costruita su Ubuntu 14.04 LTS, quindi dovrebbe ricevere aggiornamenti fino al 2017. Le caratteristiche più ordinarie, ma non per questo meno

disprezzabili di Peppermint sono semplicità e leggerezza. Questo è un ambiente che potrebbe funzionare in maniera accettabile anche su computer con 512 Mb di Ram. L'interfaccia grafica è quella spartana ma gradevole di Lxde, personalizzabile con i normali temi grafici Gtk di Gnome. Come in altre distribuzioni, sono presenti workspace multipli e un pannello in cui alloggiare orologi e gadget di vario tipo, dalle previsioni del tempo alla temperatura della Cpu. Le applicazioni di default sono semplici e rapide quanto il desktop. La prima che si

**Web App
siti in scatola**
Una delle soluzioni ideali è considerare in fase di progettazione del sito la sua fruibilità via app.

incontra dopo il login è il file manager Nemo, preconfigurato per gestire con la stessa facilità sia partizioni locali sia remote, da Smb a WebDav. Il browser preinstallato è Chromium, descritto in dettaglio nell'altro articolo del mese.

Configurazione del sistema e installazione di software funzionano come in Ubuntu, Mint e ambienti del genere, tramite un "Control Center" e gli stessi software manager di Linux Mint o Ubuntu. Però installare software su Peppermint, almeno alcune applicazioni, sarebbe un uso in conflitto con la filosofia di questa distribuzione.

La caratteristica principale di Peppermint, infatti, è proprio il fatto che fa a meno per principio, se appena possibile, delle normali applicazioni desktop. Al loro posto vengono invece proposti proprio degli Ssb costruiti con Chromium. Quelli più importanti sono in prevalenza di applicazioni

Google, come Google Drive al posto di Libre Office, o Gmail, ma ci sono anche giochi online. Google Drive è in realtà integrato in maniera più spinta di un normale Ssb grazie al programma GWOOffice, che carica automaticamente lo stesso tema grafico del resto del desktop, e supporta parecchi comandi da tastiera e combinazioni di tasti veloci personalizzate. Pur avendo tanti Ssb precaricati e facilitando molto la creazione di nuovi (vedi Box "Ice"), Peppermint lascia all'utente la massima libertà di scelta. Le Ssb preconfigurate si possono rimuovere in qualsiasi momento, senza alcun problema.

CONCLUSIONI?

Le "Web App" da desktop mostrate in queste pagine possono già semplificare la vita di parecchi utenti. Ma finché si rimarrà a questo livello, ovvero a versioni inscatolate di siti Web, spesso all'insaputa dei relativi webmaster, cambierà ben poco. Un passo avanti potrebbero essere programmi scritti fin dall'inizio per desktop, ma come se fossero siti da caricare nei browser, cioè combinando Html e JavaScript. In pratica questo avverrà su larga scala solo quando per questi programmi saranno disponibili standard aperti e accettati da tutti, che da un lato sfruttino in maniera adeguata, quando disponibili, monitor di grandi dimensioni e tastiere e mouse veri, non touch. E dall'altro, che siano integrati davvero con gli ambienti desktop tradizionali, a livello (almeno) di gestione dei file, controlli hardware e notifiche all'utente. Ne riparliamo nel prossimo numero.



RISORSE

Il mercato di Firefox è all'indirizzo <https://support.mozilla.org/en-US/kb/marketplace-apps-firefox-desktop>. L'elenco ufficiale dei dispositivi hardware su cui Chrome OS è installato dal fabbricante, e su cui quindi si potrebbe tecnicamente installare Chrome, si trova nella pagina www.chromium.org/chromium-os/developer-information-for-chrome-os-devices.



Due browser, un sistema operativo: sono i componenti dell'arsenale di Google per la navigazione Web e per dispositivi sempre connessi.

Chromium o Chrome

Browser o sistema operativo?

Ai lettori più attenti non sarà sfuggito che nell'articolo principale, parlando di Peppermint, abbiamo scritto che il suo browser preinstallato non è quel Chrome che tutti sicuramente conoscono, ma qualcos'altro meno famoso, Chromium. L'importante ruolo di Chrome nella creazione di applicazioni Ssb è un'ottima occasione per ricordare quest'altro browser (che svolge lo stesso ruolo in Peppermint, ma potrebbe farlo su qualsiasi distribuzione), il sistema operativo da esso derivato e la loro relazione con i prodotti di Google a essi collegati.

Chrome è il browser che Google ha deciso di creare e mantenere alcuni anni fa, per avere più controllo all'accesso ai suoi vari servizi online di quanto non gli sarebbe possibile se questo avvenisse solo da browser di terze parti. Oltre ad avere come obiettivo primario il maggior supporto possibile a Gmail, Google Drive e via dicendo, ogni versione di Chrome fornisce sistematicamente a Google varie informazioni su come e dove si naviga, per "migliorare la qualità dei servizi" sia di Chrome sia dello stesso Google.

Google ha rilanciato questa sua offerta nel 2009, annunciando un intero sistema operativo che è arrivato nel 2011, ed essendo costituito quasi esclusivamente da Chrome e sue estensioni è chiamato appunto Chrome OS. Inizialmente questa piattaforma doveva servire solo per connettersi a Internet e lavorare con servizi "cloud" come Gmail o Drive, senza quindi installare o eseguire alcun programma software in

locale. Nel 2014 la strategia si è leggermente modificata, con la richiesta che le applicazioni per Chrome/Chrome OS del Play Store ufficiale possano funzionare anche offline, e soprattutto con il supporto nativo in Chrome OS delle app di Android.

Chrome OS rimane comunque un ambiente pensato non tanto per essere installato da utenti finali su loro hardware, quanto per essere caricato direttamente in fabbrica su pochi modelli di netbook che garantiscano la migliore "esperienza" possibile. A titolo di esempio, un Chromebook ufficiale (www.google.com/chromebook) si accende in pochissimi secondi, si aggiorna automaticamente (senza chiedere permessi) tutte le volte che è necessario e stampa con il servizio Google Cloud Print. Chrome OS contiene anche funzioni firmware per recupero del sistema, e alcune app proprietarie, tra cui quella per Google Talk.

CHROMIUM E CHROMIUM OS

Quando Google decise di sviluppare prima un nuovo browser, poi un intero sistema operativo su cui farlo girare, scelse la via più economica e sensata, ovvero partire da software Open Source. Il nucleo del browser è tutto software libero, quello del sistema operativo è Linux. Entrambi quei prodotti sono funzionanti e disponibili in maniera indipendente da Google, e sono appunto il browser Chromium

(www.chromium.org) e il sistema operativo Chromium OS (www.chromium.org/chromium-os). Le differenze complete fra un browser e l'altro sono descritte in dettaglio nella pagina <https://code.google.com/p/chromium/wiki/Chromium-BrowserVsGoogleChrome>. Quelle più importanti, al di là dell'ovvia differenza fra loghi e mancanza nei due Chromium di qualsiasi menzione "ufficiale" di Google, sono quelle già citate. Nel browser, sia per principio sia per ragioni legali, mancano innanzitutto le varie funzioni di monitoraggio di Google. Per gli stessi motivi, non sono integrate in Chromium parti proprietarie come l'interfaccia Cloud Print, codec AAC e MP3, player Flash e lettore PDF. Diversi di quei componenti sono comunque aggregabili a mano,

o sostituibili con prodotti sostanzialmente equivalenti, anche se con procedure più complicate che su altre distribuzioni Linux.

Quanto a Chromium OS, si aggiorna solo se lo desidera l'utente ed è utilizzabile e supportato, ovviamente non in maniera ufficiale da alcuna azienda, su un maggior numero di piattaforme hardware del suo parente più famoso. Ancora più importante è il fatto che Chromium non include alcune delle componenti proprietarie del suo fratello sponsorizzato, o almeno il relativo codice sorgente.

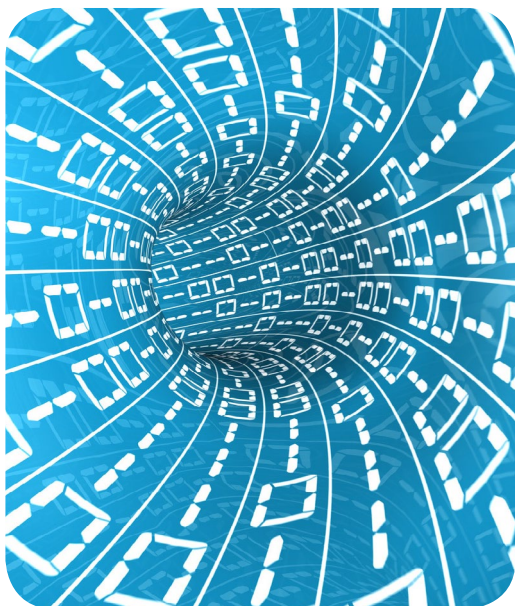
In sostanza, probabilmente Chromium OS non è ancora pronto per la maggioranza degli attuali utenti Linux. Non quanto Chrome OS, almeno. Però, sono entrambi sistemi da tenere d'occhio nel prossimo futuro, come ponte fra il desktop tradizionale e uno un po' più vicino al mondo mobile, o almeno fra Linux e le applicazioni Android. •

Sistema operativo

Chrome OS, un ambiente pronto all'uso per una nuova generazione di device "always on"

NEWS

WEBASSEMBLY, UN NUOVO **LINGUAGGIO** PER IL WEB.



Oggi come oggi, lo standard di fatto per la programmazione di qualsiasi parte dinamica o interattiva di una pagina Web, dai menu a tendina alle animazioni, è JavaScript. Questo linguaggio ha però diverse limitazioni sempre meno trascurabili, in un mondo in cui aumentano continuamente i browser che devono scaricare ed eseguire quel codice su terminali mobili, quindi con connessioni e processori relativamente lenti. I programmi o, come si diceva una volta, gli applet Javascript sono infatti semplici file di testo, codice sorgente così com'è stato scritto dal loro autore. Ogni volta che vengono scaricati, devono essere interpretati, ovvero tradotti al volo in codice eseguibile da un'apposito motore software che provvederà a eseguirli. Questa procedura è semplicissima, ma lenta sia in fase di download sia nell'interpretazione.

Per ovviare a questi colli di bottiglia i maggiori produttori di browser (Mozilla, Google, Microsoft e Webkit, cioè Safari) si sono riuniti nel giugno 2015 per lo sviluppo di un formato binario intermedio, chiamato WebAssembly (<https://github.com/WebAssembly>), che avrà due obiettivi principali. Il primo è aumentare la velocità delle applicazioni eseguite nei browser: il codice di WebAssembly, infatti, sarà sia scaricabile sia eseguibile più velocemente di JavaScript, perché costituito da istruzioni binarie (byte code) precotte, ancora più comprimibili del testo, e già quasi pronte per essere eseguite. L'altro motivo per lanciare WebAssembly è facilitare l'uso dall'interno dei browser, di librerie software o anche intere applicazioni scritte in linguaggi diversi da JavaScript. Le prime versioni di compilatori WebAssembly possono infatti accettare sorgenti in linguaggio C e C++. Altri linguaggi, come Rust e C#, sono già in lista d'attesa, ma potrebbero dover aspettare il primo rilascio di un traduttore da WebAssembly a JavaScript! Questa funzione sarà in pratica indispensabile nei primi tempi, per consentire l'uso di WebAssembly anche a browser non ancora aggiornati col relativo interprete.

SOFTWARE **LIBERO** SEMPRE UGUALE, FINO ALL'ULTIMO BIT

Ivari processi di produzione di software potrebbero sembrare completamente deterministici, ma questo non completamente vero. Non sempre, almeno, non fino a quando non diventerà operativo un nuovo progetto nato in Debian. La traduzione da codice sorgente a linguaggio macchina che chiamiamo (semplificando) compilazione non funziona sempre, esattamente, allo stesso modo. A volte può bastare qualche spazio bianco in più in un file, oppure una data di modifica diversa, perché i compilatori e tutti i loro programmi ausiliari producano file binari diversi ogni volta che girano. Non dal punto di vista funzionale, ovviamente! Il programma ottenuto dalla compilazione di una specifica sequenza di istruzioni in codice sorgente avrà sempre lo stesso comportamento, anche se ogni volta il suo autore dovesse introdurre linee bianche, o nuovi commenti di qualsiasi tipo. Cambi come quelli, o altri dello stesso genere, potrebbero però portare il compilatore a creare file eseguibili che, considerati come sequenze di bit, non sono completamente identiche. Questo è un problema perché, fra le altre cose, rende difficile la tracciatura dei programmi, per capire senza incertezze chi e quando ha compilato una certa versione. Il progetto "Reproducible Builds" (<https://wiki.debian.org/ReproducibleBuilds>) di Debian intende proprio porre fine a queste



incertezze, garantendo che ogni singolo pacchetto Debian, e ogni passo della sua compilazione, siano sempre riproducibili allo stesso modo, fino all'ultimo byte. Dopo i primi risultati in Debian, a inizio estate 2015, gli autori hanno iniziato a fornire versioni dei loro strumenti anche ad altre distribuzioni Linux come Ubuntu e Fedora.