

Firma digitale e accesso alla PA online con Linux



Posso dire, ormai da luglio, che si può firmare digitalmente e accedere alla Pubblica Amministrazione (PA) online con [GNU/Linux](#), nel mio caso [Ubuntu Mate](#).

Anche se vi conviene leggere i vari “[Aggiornamenti](#)” in fondo a questo articolo.

A luglio, infatti, una delle associazioni di cui faccio parte doveva fare i permessi per poter fare una manifestazione. Permessi da fare presso il SUAP (Sportello Unico per le Attività Produttive) locale, e che da un paio di anni vanno fatti per forza via web.

Fino a luglio, quindi, un po' per pigrizia, un po' per poco tempo, le pratiche SUAP le facevamo con i pc Windows dei compagni di associazione. A luglio, però, tempo di ferie e quindi di un po' più di tempo libero, mi sono messo d'impegno, per capire se e come fosse possibile poter fare questa procedura con GNU/Linux, ed è possibile, almeno con la mia configurazione:

- [Ubuntu Mate](#) 17.04 x64;
- Lettore di smart card Bludrive II (<http://www.blutronics.com/Default.aspx>);
- Dike6 (https://rinnovofirma.infocert.it/download/x86_64/latest);
- Firefox, l'ultima versione ufficiale disponibile nei repository di Ubuntu (chi usa un altro browser dovrà cercare di capire dove e come configurare quanto spiegherò sotto).

Data questa configurazione il processo è, tutto sommato, abbastanza semplice.

- Collegare il dispositivo al computer:

Nel mio caso, collegato il dispositivo al computer questo lo riconosce immediatamente, come si può evincere dai log di sistema:

```
Sep 28 14:57:51 Astio kernel: [2298796.040746] usb 2-1.2: new full-  
speed USB device number 16 using ehci-pci  
Sep 28 14:57:51 Astio kernel: [2298796.153906] usb 2-1.2: New USB  
device found, idVendor=1b0e, idProduct=1078  
Sep 28 14:57:51 Astio kernel: [2298796.153909] usb 2-1.2: New USB  
device strings: Mfr=1, Product=2, SerialNumber=0  
Sep 28 14:57:51 Astio kernel: [2298796.153911] usb 2-1.2: Product:  
BLUDRIVE II CCID  
Sep 28 14:57:51 Astio kernel: [2298796.153913] usb 2-1.2:  
Manufacturer: BLUTRONICS  
Sep 28 14:57:51 Astio mtp-probe: checking bus 2, device 16:  
"/sys/devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.2"  
Sep 28 14:57:51 Astio mtp-probe: bus: 2, device: 16 was not an MTP  
device  
Sep 28 14:57:56 Astio colord-sane: io/hpmud/pp.c 627: unable to  
read device-id ret=-1
```

Se tutto fila liscio, il led sul lettore diventa verde. Fine.

Se si vogliono maggiori info, c'è anche uno strumento di verifica fatta dalla stessa Blutronics, con tanto di video che spiega come installarlo:

- Installare [Dike6](#):

Fatto questo si va sul sito di [Infocert](#) e si scarica la versione gratuita di Dike6, più che sufficiente per quel che dobbiamo fare noi. Per chi usa una distro *Debian like* la cosa è ancora più semplice, perché si può scaricare un comodo .deb e installarlo come sempre.

A questo punto siamo già nella possibilità di accedere ai dati della nostra smart card, sia essa la semplice tessera sanitaria / codice fiscale, oppure – come è indispensabile nel nostro caso – la [Carta Nazionale dei Servizi](#), cosa che è fattibile solo dopo aver preso ed attivata la carta (operazione che non c'entra nulla con l'informatica, ma con la burocrazia italiota).

- Configurare Firefox per accedere al sito del proprio SUAP locale (o altro sito della PA):

A questo punto ci siamo quasi: per poter fare una pratica online su un portale SUAP, infatti, bisogna aver configurato il proprio browser perché legga la carta inserita nel lettore. Quindi bisogna aprire firefox ed andare in

Preferenze -> Avanzate -> Certificati -> Dispositivi di sicurezza

cliccare su Carica, inserire una descrizione (es.: TesseraSanitaria o SUAP o quel che si vuole) ed il percorso, che nel mio caso è:

/opt/dike6/libbit4xpki.so

A questo punto, sul sito SUAP si clicca su “Sportello online”, si sceglie il proprio Comune e si avvia la pratica. Il sistema vedrà – grazie alla configurazione di cui sopra – la carta inserita nel lettore, chiederà il PIN (rilasciato dall’autorità competente), ed a quel punto si potrà fare tutta la procedura del caso.

Usando la Carta dei servizi si può accedere alle varie piattaforme esattamente come con la Tessera Sanitaria, ma si può anche firmare digitalmente i vari documenti che bisogna presentare, e la cosa è assai comodo.

Fine ☐

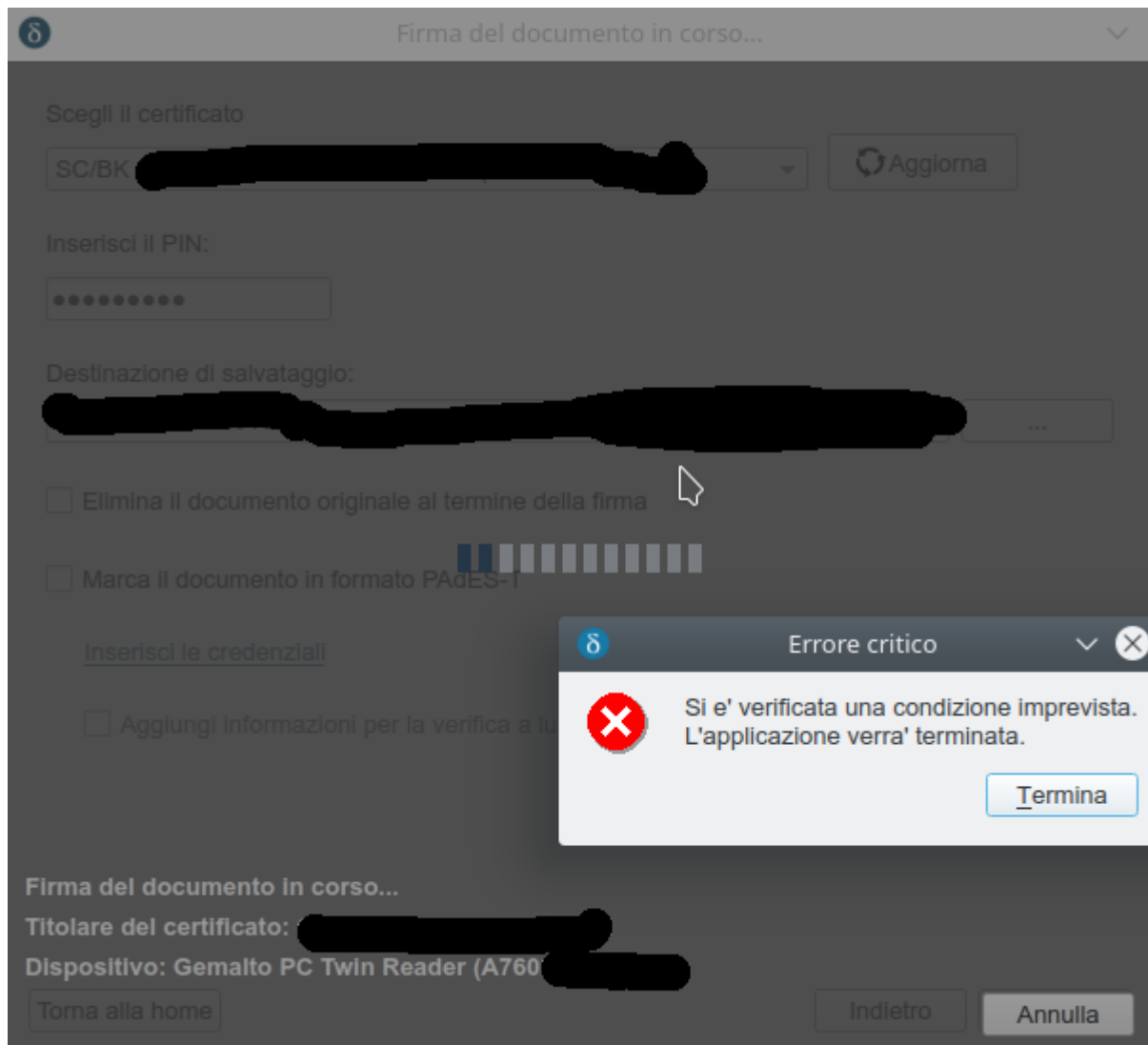
Aggiornamento Ubuntu-(qualsiasi flavors) 18.04 e derivate (anche Mint 19)

Arriva aprile 2018, faccio come sempre passare qualche settimana, spulcio la rete e vedo che non c’è nessun problema a passare alla nuova LTS della famiglia Ubuntu.

[Aggiorno!](#)

Tutto fila liscio (+ o -) e mi ritrovo con la nuova et fiammante versione che va come una freccia. Evviva.

Poi si avvicina l’estate, ed iniziano a fioccare le iniziative paesane e la necessità, [sempre maggiore e sempre più incomprensibile](#), di fare miliardi di permessi online alla PA, e quindi – forte di questo mio articolo – attacco il lettore di smartcard al computer, infilo la tessera per la firma digitale, apro Dike6, che chiede subito di essere aggiornato e...



E niente, non c'è versi di farlo funzionare.

[Contro la mia religione provo](#) con [Aruba Sign](#), che fino alla 17.10 funzionava senza problemi, ma questo non trova il dispositivo.

Come ho risolto?

Recuperando, fortunatamente, un pc con Ubuntu 16.04 installato, e tutto ha funzionato senza problemi, le procedure sono state portate a termine, i permessi rilasciati, le feste svolte senza altri problemi.

Mi permetto, a questo punto, una piccola riflessione:

se non potete permettervi di stare dietro ai rilasci **STABILI** delle varie distribuzioni GNU/Linux, ed obbligate i cittadini – ed i professionisti – ad usare versioni del sistema operativo vecchie di anni, allora – a mio modesto avviso – avete dei problemi e non siete aziende di software serie.

Aggiornamento settembre 2018

Ora (21 settembre) l'ultima versione di [Dike6](#) funziona con ubuntu 18.04, si per quel che riguarda la firma sia per quel che riguarda l'accesso al SUAP.

Bravi gli sviluppatori!

Cambio di dominio e redirect 301



I più attenti di voi 7 si saranno sicuramente accorti che da ieri questo sito ha subito un cambio di dominio. Ciò è accaduto perché mi stava “stretto” il precedente .com (va a sapere perché lo registrarai: devo parlarne col mio psycho...) e quindi, con una botta d’istinto ho registrato un più equilibrato .info.

Un altro motivo era tecnico (che non sto qui a spiegare), che però mi ha portato a non poter fare un clone del vecchio sito sul nuovo, ma a spostarlo quasi manualmente (quasi), andando a perdere tutta una serie di informazioni, di configurazioni e – soprattutto – la quantità di lettori e di letture che avevo sul vecchio dominio.

Perciò mi sono trovato nella condizione di voler e dover far puntare il vecchio dominio a questo nuovo, così quando quella persona all’anno viene a cercarmi sui motori di ricerca e trova (ancora) il .com e clicca su un link, viene automagicamente reindirizzata su questo sito, alla pagina o al post giusti.

Ma come si fa questa magia? Gugol mi ha aiutato a scoprirlo, e dopo qualche lettura sono arrivato alla conclusione più consona per il mio caso.

Il mio caso – torno a ricordarlo – è quello di chi vuole che il vecchio dominio (.com) vengare **completamente** reindirizzato sul nuovo: clicco su un link che punta ad una pagina del .com, finisco sulla stessa del .info; idem con gli articoli e tutte le altre risorse.

Ciò che ho usato è il famoso **redirect 301 “Moved Permanently”**, cioè il meccanismo che permette il risultato di cui sopra. Il 301, tra l’altro, dice

sia parecchio apprezzato da gugol e dai motori di ricerca in genere, quindi viva.

Nel mio caso, poi, ho potuto farlo perché sul server che ospita il mio sito c'è il noto server web [Apache](#), su cui è configurato, tra le tante cose, anche il [Mod_rewrite](#); e come se non bastasse ho accesso al mio file [.htaccess](#).

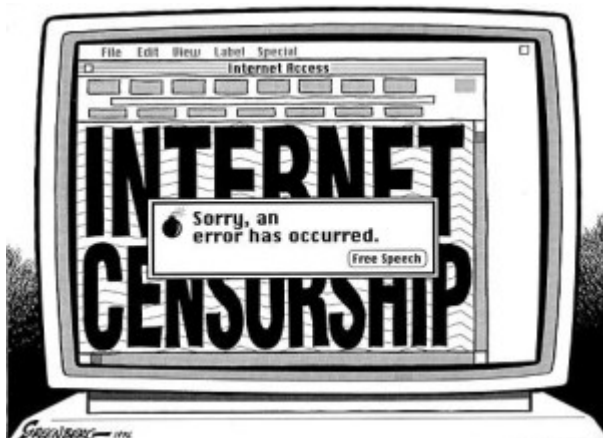
Dati questi strumenti cosa ho fatto?

Mi sono loggato sul mio spazio di hosting e sono andato a modificare il mio file .htaccess, aggiungendo in cima a tutto le seguenti righe di codice:

```
Options +FollowSymLinks
RewriteEngine on
RewriteRule (.*) http://francovite.info/$1 [R=301,L]
# BEGIN iThemes Security
```

Fatto: a questo punto, tutte le volte che quel povero sfortunato che decide, per chissà quale folle motivo, di cliccare su un link che porta al mio vecchio .com ... taaac, viene rediretto direttamente qui al .info

[Controllo di Stato. Tutela la tua privacy con Frepto](#)



Sabato scorso si è tenuto, presso la sede della [Libreria AmiataAutogestita](#), ad Arcidosso, un incontro dal titolo "Meeting sulla **CONSAPEVOLEZZA INFORMATICA**", il cui scopo era quello di iniziare un percorso di consapevolezza nell'uso degli strumenti informatici.

Ad aiutarci in questo percorso sono stati gli amici dell'Hacklab romano [AvANa](#), che sono venuti a presentarci due progetti:

1. uno è un "opuscolo" dal titolo significativo: [Crypt'я'Die](#);
2. l'altro è il progetto [Frepto](#).

Nel primo caso si tratta di un opuscolo in cui si fa il punto della

situazione rispetto al controllo che si può subire quando si usa un computer o uno smartphone/tablet. Il risultato della chiacchierata è che praticamente qualsiasi computer con Windows o Mac è controllabile, attraverso dei particolari virus informatici che si chiamano "[malware](#)", mentre lo sono TUTTI gli smartphone/tablet.

La soluzione, almeno per i computer, sta – almeno parzialmente – nell'usare sistemi OpenSource, come GNU/Linux, e strategia di sicurezza nell'uso di questi sistemi. Una delle possibilità sta proprio in Freept, che è

un sistema GNU/Linux completo su penna USB. Questo significa che puoi portare la pennetta sempre con te ed utilizzare qualsiasi computer proprio come se fosse il tuo portatile. Inoltre i dati che salverai all'interno di questa pennetta saranno automaticamente cifrati (ovvero non potranno essere letti da nessun altro).

L'idea che sta alla base dello sviluppo di Freept è quella di offrire un sistema semplice per la gestione sicura degli strumenti utilizzati più frequentemente dagli attivisti, senza però rinunciare alla comodità di un sistema operativo tradizionale

Se quelle sopra potevano essere lette come manie persecutorie dei soliti estremisti complottardi, ecco che arriva a sostegno di quel che si dice una fonte che può essere tutto fuorché estremista e/o complottardo, visto che è del gruppo politico di Monti. Leggo, infatti, proprio stamani che, sul blog di [Stefano Quintarelli](#) che:

[Una svista rilevante nel provvedimento antiterrorismo](#)

E si legge che

Il provvedimento antiterrorismo modifica il codice di procedura penale così:

All'articolo 266-bis, comma 1, del codice di procedura penale, dopo le parole: «è consentita l'intercettazione del flusso di comunicazioni relativo a sistemi informatici o telematici ovvero intercorrente tra più sistemi», sono aggiunte le seguenti: anche attraverso l'impiego di strumenti o di programmi informatici per l'acquisizione da remoto delle comunicazioni e dei dati presenti in un sistema informatico.

con questo emendamento l'Italia diventa, per quanto a me noto, il primo paese europeo che rende esplicitamente ed in via generalizzata legale e autorizzato la "remote computer searches" e l'utilizzo di captatori occulti da parte dello Stato!

il fatto grave è che questo non lo fa in relazione a specifici reati di matrice terroristica (come fa pensare il provvedimento), ma per tutti i reati "commessi mediante l'impiego di tecnologie

informatiche o telematiche" (art.266 bis).

Quindi, **qualsiasi** reato commesso attraverso l'uso di strumenti informatici (ed oggi giorno, volendo, qualsiasi "criminale" usa uno smartphone, e quindi una "tecnologia informatica o telematica"), permette agli inquirenti di usare "remote computer searches", ovvero Malware con cui ascoltare, leggere, sapere TUTTO quello che facciamo sul nostro computer.

Perché questa cosa è così grave? Perché, ci dice Quintarelli

l'uso di captatori informatici (Trojan, Keylogger, sniffer ecc.ecc.) quale mezzo di ricerca delle prove da parte delle Autorità Statali (giudiziarie o di sicurezza) è controverso in tutti i paesi democratici per una ragione tecnica: con quei sistemi compio una delle operazioni più invasive che lo Stato possa fare nei confronti dei cittadini, poiché quella metodologia è contestualmente

una ispezione (art. 244 c.p.p.)

una perquisizione (art. 247 c.p.p.)

una intercettazione di comunicazioni (266 c.p.p.)

una acquisizione occulta di documenti e dati anche personali (253 c.p.p.).

tutte attività compiute in un luogo, i sistemi informatici privati, che equivalgono al domicilio. E tutte quelle attività vengono fatte al di fuori delle regole e dei limiti dettate per ognuna di esse dal Codice di Procedura Penale.

[...]

[Tale legge comporta] rischi di un controllo pervasivo dei sistemi, il superamento dei sistemi di cifratura, la profilazione di comportamenti estranei ai reati perseguiti, una invasione della sfera privata e della riservatezza, nonché rischi di danni ai sistemi informatici ed il coinvolgimento di dati di terzi estranei.

Al punto, dice sempre Quintarelli, che probabilmente si viola anche la Costituzione, dall'art. 13 al 15.

Insomma, le paranoie dei soliti complottardi, alla fine, non sono poi così tanto e solo paranoie ...

Ubuntu + Mate Desktop: ci siamo quasi!



Il mio desktop con Mate

Ricapitoliamo la vicenda:

qualche anno fa – ormai un paio, se non erro – [Canonical](#), l'azienda madre che ha creato e distribuisce [Ubuntu](#) – la più usata distribuzione [GNU/Linux](#) al mondo – decide di abbandonare [Gnome](#) per creare un proprio desktop manager: [Unity](#).

A sua volta Gnome decide di cambiare pagina, di abbandonare il desktop come si è sempre conosciuto fino a quel momento (cosa non necessariamente negativa), per passare ad un nuovo modello di sviluppo e di utilizzo del desktop: [Gnome Shell](#).

Urge scegliere: con l'arrivo di Ubuntu 12.04 LTS (la versione di punta nel 2012), bisognava “per forza” adeguarsi alle nuove scelte o trovare un'alternativa (come racconto in [un mio post](#) dell'anno scorso):

provo prima con Gnome Shell, ma è una tragedia. Di base fa il giusto, ma senza [estensioni esterne](#) è veramente molto povero (parlo del 2012, ovviamente);

provo poi Unity, ma col mio Pentium Dual Core e 4 gb di Ram torno ad avere un pc così lento che rischio di rimpiangere XP (che, tra l'altro, non ho mai usato :).

Quali alternative?

Allora erano presenti – e lo sono ancora oggi – varie alternative:

- [Gnome Classic](#)
- [Kde](#)
- [Xfce](#)
- [Lxde](#)

Solo per rimanere nel fantastico mondo dei Desktop Manager. Ma nessuno di questi, per un motivo o per l'altro, mi soddisfa.

Il più papabile sarebbe Kde, che ho usato per parecchi anni, ma pure loro, col passaggio dalla versione 3 alla 4, hanno deciso di cambiare completamente filosofia, raggiungendo sicuramente risultati superiori a Gnome Shell e Unity (per quel che mi riguarda, almeno), ma anche ottenendo una pesantezza del sistema agghiacciante.

Alla fine leggo una notizia interessante: hanno iniziato lo sviluppo di un nuovo/vecchio desktop manager, [Mate Desktop](#), che altro non è che un [fork](#) del caro vecchio Gnome 2, quello che ho usato felicemente per anni fino al 2012.

Mi fiondo a provarlo!

Inizio a provarlo che ancora non è arrivato alla 1.0 e ne resto soddisfatto. E da allora non lo mollo più. Col passare del tempo installarlo nella mia Ubuntu è sempre più semplice, i [PPA](#) arrivano direttamente dagli sviluppatori di Mate, il sistema è sempre più stabile e sofisticato. Evviva!

Non mancano i problemi, ovviamente. Un po' per la giovane età del progetto, un po' per la difficile integrazione di un desktop manager esterno all'interno di un altro sistema che non lo prevede. Ogni tanto sparisce qualcosa dalla [system tray](#), ogni tanto [crasha](#) qualcosa, ma alla fine sono tutti problemi relativamente semplici da risolvere, e tutto funziona a meraviglia.



Il logo ufficiale di Ubuntu-MATE
Remix

Fino a quando LA notizia:

nientepopodimenoche [Marks Suttleworth annuncia su Google Plus](#) che [Ubuntu Mate Remix](#) sarà una nuova derivata ufficiale di Ubuntu, e che verrà rilasciata ad ottobre, con la prossima Ubuntu 14.10!!

Gioia et gaudio, che bellezza, non c'è che da aspettare l'autunno e poi buttarsi sulla nuova distro.

È di questi giorni, però, che le splendide novità non finiscono (doppia gioia et gaudio):

gli sviluppatori di Ubuntu-MATE hanno rilasciato dei repository ufficiali per Ubuntu 14.04 LTS, l'attuale versione stabile di Ubuntu, con cui si può già installare sul proprio pc Ubuntu-MATE. Ecco come:

```
sudo apt-add-repository ppa:ubuntu-mate-dev/ppa
sudo apt-add-repository ppa:ubuntu-mate-dev/trusty-mate
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install --no-install-recommends ubuntu-mate-core
ubuntu-mate-desktop
```

Fatto. Se proprio vogliamo fare i raffinati e vogliamo che il sistema, all'accensione e allo spegnimento del pc, ci dica "Oh, sono Ubuntu-MATE", allora basta dare i seguenti comandi:

```
sudo update-alternatives --config default.plymouth
```

e scegliere il tema mate per plymouth; e poi

```
sudo update-initramfs -u
```

per far si che venga inserito all'avvio del sistema.

Tutto ciò ci permette non tanto e non solo un sistema più carino graficamente (che alla fine ok, ma chi se ne frega!); quanto un sistema più integrato e funzionante: tutti i problemini di cui parlavo sopra (e qualcuno me lo sono pure dimenticato), con questo aggiornamento sono spariti. Tutto funziona perfettamente, tutto è perfettamente integrato e, finalmente, si inizia a vedere la luce!!

Buon smanettamento! ☐

[Ubuntu, come perdere utenti. Ovvero avventure con Unity, gnome3 e Mate Desktop](#)



Mate Desktop

Ok, premettiamo che pure io sono un po' pirla, questo si sa:

nonostante la mia [Ubuntu 12.04 LTS](#) funzionasse a dovere era un po' che ero incuriosito da cose che su quest'ultima non giravano, tipo gnome 3.8, per fare un esempio. Ho resistito, sono stato bravo, poi però ho ceduto.

Alla fine l'uscita di [GNU/Linux Mint 15](#) – mercoledì scorso – ha dato la botta finale: ho scaricato la iso della versione con [Mate Desktop](#), masterizzata, avviata per vedere com'era e se era performante, e poi, in un impulso di follia, l'ho installata. Tutto liscio, tutto veloce (ha già i codec video/audio installati, quindi non ho dovuto aspettare la solita mezzora che mi tocca qui alla periferia del mondo, dove la nostra adsl va ancora a 70 ks ...).

E già la prima bestemmia: Mate Desktop su Mint 15 / [Ubuntu 13.04](#) non ha più l'indicatore con l'audio, i messenger e tutte quelle cosine, forse non essenziali, ma tremendamente comode. Un po' di ricerca, un po' di smanettamenti, e alla fine almeno il sound indicator rispunta fuori e ci contenta.

Poi, però, uno fa tutta sta fatica e non lo prova gnome 3.8? Eh, taac, [OMG Ubuntu](#) mi dice come e io eseguo:

```
$ sudo add-apt-repository ppa:gnome3-team/gnome3
```

```
$ sudo apt-get update && sudo apt-get install gnome-shell ubuntu-gnome-desktop
```

Errore! Moccolo!!

Insomma, ve la faccio breve, gira e ti rigira, alla fine incasino un po' di roba che uso comunemente – tipo il mio amato [rhythmbox](#) – mi incazzo, tiro fuori il cd di ubuntu 13.04 e reinstallo tutto da capo. Di domenica.

15 minuti di installazione, la solita mezzora per aggiornamenti e codec vari, riavvio e taac, [unity](#) 7.0 fiammante in tutta la sua bellezza. E pesantezza.

Per dire:

io non è che abbia un computer da mille neuri, ma è pure sempre un Intel Core i3 con 8 (dico otto) giba di ram. E nonostante tutto il maledetto unity fa una fatica boia a girare. Io boh. Ma tant'è, stamani mi ci metto d'impegno (relativo: mentre lavoro) e provo a rifarmi una mattinata con unity e vediamo com'è.

Alla fine non è male, se non fosse, appunto, mostruosamente pesante. Ma lo è. E poi...

La goccia che ha fatto traboccare il vaso è stata l'assenza totale della [system tray](#), che per gli italiani sarebbe l'area di notifica, il “vassoio di sistema”, come tradurrebbe gugol. Cioè quella parte del pannello in cui si vanno felicemente a posizionare le icone dei programmi attivi ma minimizzati, tipo – chissò – [Thunderbird](#), con la sua iconuccia che ci dice se ci sono mail nuove e quante, [Liferea](#) che fa lo stesso coi feed rss, skype e così via. Ecco, il Signor [Shuttleworth](#), il Padrone di Ubuntu ha deciso che no, ora la system tray è una cosa vecchia e va cavata che tanto c'è la [dash](#) e il [launcher](#) e non fatemi dire parolacce.

Ok, [parolaccia]:

```
$ sudo add-apt-repository “deb http://repo.mate-desktop.org/ubuntu $(lsb_release -sc) main”
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install mate-archive-keyring
```

```
$ sudo apt-get install mate-core mate-desktop-environment
```

Piccolo smanettamento per due sciocchezze, e taac, rieccomi col mio **vecchio**, caro desktop col pannello, il vassoio di sistema, le applet e tutte le cosine **antiche** che mi fanno lavorare tanto bene e tanto velocemente.

PS:

E gnome 3.8?

Installato e avviato:

non sono riuscito manco a configurare una [parolaccia] di estensione e manco un tema. Dopo mezz'ora che giravo per internet a cercare di capire come fare mi sono fermato e mi sono detto:

ma per lavorare e far funzionare un desktop devo girare mezz'ora per internet? Nel 2013? Ma siamo scemi??

E' morto un genio dell'informatica, Dennis MacAlistair Ritchie

Tra l'8 e il 9 di ottobre è morto un **genio** dell'informatica. Un uomo grazie a al cui lavoro – suo e di tanti e tante altr@ hacker degli anni '60 e '70 – oggi possiamo usare un computer, usare la rete, e tante altre cose che crediamo banali. Grazie a **lui** oggi uomini di marketing, che si e no sapevano cosa fosse il codice, sono passati da forma umana a forma semidivina.

E visto che [c'è già chi ha scritto](#) meglio di quanto io potrei quel che c'era da dire, non faccio altro che riportarlo tale e quale.

Grazie dmr, e massimo rispetto.

Per Dennis Ritchie



Un piccolo e sincero tributo a Dennis MacAlistair Ritchie lo voglio scrivere pure io, che non riesco a mettere giù 50 righe in C senza fare qualche macello e combino casini fenomenali ogni volta che tocco i config su di uno *nix. Perché Ritchie, tra la fine degli anni Sessanta e i Settanta, ha creato il linguaggio C e ha dato un contributo fondamentale allo sviluppo di Unix. E

se non sai valutare in pieno queste affermazioni posso forse aggiungere che ha vinto, insieme al compagno di sempre Ken Thompson, i premi più prestigiosi a cui una persona impegnata nel campo del *computing* possa ambire, Turing Award (il "Nobel dell'informatica") e National Medal of Technology degli Stati Uniti inclusi, e ha scritto il libro più (odios)amato della storia della programmazione, *The C Programming Language*, insieme all'altro compagno di Bell Labs, Brian Kernighan. Il testo nella tradizione geek è religiosamente indicato con gli acronimi degli autori, *K&R*, mentre Ritchie è noto semplicemente come *dmr*, dal suo username. Creato in tempi in cui gli 8 caratteri oggi spesso richiesti sarebbero sembrati inammissibile spreco ingegneristico, e per lo stesso motivo i comandi di Unix sono così corti, *ls* va benissimo, *list* è inutile sperpero di bit.

A questo punto dovrei forse scendere negli aneddoti, spiegare ai profani che oggi stampare "Hello, World!" è irrinunciabile primo esempio di un nuovo linguaggio di programmazione proprio in omaggio a K&R, o citare i suoi famosissimi giudizi su C "quirky, flawed, and an enormous success" e su Unix " basically a simple operating system, but you have to be a genius to understand the simplicity", o ricordare l'ambiente leggendario dei Bell Labs (e proprio un compagno dei Bell Labs, [Rob Pike](#) ha dato la notizia della scomparsa su G+, alle 3 ora italiana, mentre da Wikipedia apprendo che Ritchie è mancato tra l'8 e il 9 Ottobre).

Se non hai una preparazione informatica forse queste cose ti sembreranno notevoli ma non così decisive, insomma *questo "dmr" era un capoccione tanto ma chiusa lì*. In fondo la sua [voce Wikipedia](#) è bella breve, in fondo non è stato un famoso imprenditore, in fondo cosa c'entrano queste cose dall'apparenza molto "teorica" con il mondo di oggi, con gli smartphone e Facebook e le case intelligenti? C'entrano tutto, poiché in una prospettiva moderatamente riduzionista è lecito dire che ogni pezzo contemporaneo di tecnologia avanzata contiene (almeno potenzialmente) del C (scritto in C, compilato in C, fornito di interprete C ecc.) e spesso uno Unix. Questi sono i mattoni e le fondamenta su cui è stata costruita tutta la realtà digitale, da Internet al computer nell'automobile. Non dico che sia solo un pregio, ma è così, e se non vedi che è così, ciò è dovuto solo al fatto che, per comodità e progresso nella tecnica, questo ultimo livello di C (e Unix) viene spesso nascosto attraverso strumenti più comodi (per chi è troppo *lazy*, direbbero ovviamente gli hacker C). E anche il tuo Mac quando ha voluto un sistema operativo serio si è convertito allo Unix-like, ovviamente restando sempre, al contrario di Linux e compagnia, nel closed source. E se non ti fidi delle mie parole senti anche il notissimo blogger di tecnologia [Stefano Quintarelli](#): "di lui sì che si può dire che sul suo lavoro si è costruita Internet e che il frutto del suo lavoro è ormai ovunque, dai telefonini agli ascensori, in tutto il mondo. Decisamente un Grande, non un genio del marketing."

E' oggi troppo facile contrapporre Dennis Ritchie al co-fondatore di Apple scomparso qualche giorno fa. Per lui non verranno tappezzate le città con manifesti politici di compianto né diffusi speciali tv (qui da noi, ovviamente, la BBC forse farà qualcosa) e il suo nome risuonerà oggi solo tra i computer scientists, compresi tutti quei CEO come Bezos di Amazon e Zuckerberg di Facebook che sono in corsa per prendere il posto del co-fondatore di Apple nell'immagine popolare di "genio informatico". Il suo nome

risuonerà oggi solo tra i computer scientists, ovvero in centinaia e migliaia di post anche su siti molto noti (ad es. [Boing Boing](#)), ma come una notizia settoriale, un culto infine “privato”, nonostante ogni trending mondiale sul nerdistico Twitter, perché davvero di spendibile mediaticamente per il pubblico più largo c'è poco, *solo il C e Unix...* E la contrapposizione, tutta occasionale per le date di scomparsa e le manifestazioni di cordoglio, con il CEO di Apple. Lo ricorderanno in questi giorni i grandi compagni dei Bell Labs e forse gli stessi imprenditori citati sopra e i sommi che con i suoi strumenti hanno direttamente contribuito a plasmare il mondo tecnologico di oggi, da Guido Van Rossum di Python a Linus Torvalds di Linux e oltre. E ho voluto farlo anch'io, prima di mettermi a scrivere sopra il mio Linux Ubuntu un 40 righe di Python per un piccolo servizio che si appoggia ai Google Docs. Perché, come chiunque stia programmando in questo momento, io sto costruendo sopra quello che ha costruito Richie o il costruttore di livello n che ha infine costruito sulle sue spalle. Perché siamo tutti nani sulle spalle di quel gigante.

[E' morto un uomo di marketing, Steve Jobs](#)

Debbo dire che sono abbastanza disgustato per il diluvio di lacrime che ha scatenato la morte del miliardario Steve Jobs. Mi ha disgustato di più del dovuto forse in seguito all'assordante silenzio [sulla morte delle 4 operaie in nero pugliesi](#), schiacciate dalle macerie dell'azienda per cui lavoravano a 4€ l'ora...

Un anno fa usciva un libro sulla Apple ([Mela marcia](#), alla cui stesura ho partecipato pure io) e su cosa era diventata dal ritorno di Steve Jobs alla sua guida: una delle maggiori multinazionali dell'informatica attuale, per i soci; una delle peggiori per i lavoratori e gli utenti (se attenti alle proprie libertà e alla propria privacy). Visto che fu pubblicato sotto licenza [Creative Commons](#), ripubblico qui il mio capitolo, quello sulla storia della Apple. Una storia un po' diversa da quella canonica.

E 'fanculo.

1. L' Homebrew Computer Club

Il 5 marzo 1975 si riunisce per la prima volta, nel garage di uno degli organizzatori, Gordon French, quello che diventerà l'Homebrew Computer Club, un luogo in cui si potessero riunire le persone che volessero costruirsi il loro computer, e vi partecipano alcuni dei migliori e più innovativi appassionati del settore informatico dell'epoca. Che fossero ingegneri o quattordicenni brufolosi, la cosa non interessava a nessuno: bastava dimostrare di saperci mettere le mani. Tra i tanti – e non stiamo parlando di “comuni mortali”, ma di gente del calibro di Lee Felsenstein, uno dei più

importanti personaggi della storia dell'hacking mondiale; Bob Albrecht, fondatore della People's Computer Company; o personaggi come John "Captain Crunch" Draper, il primo Phreaker (1) del mondo, divenuto celebre come delle Blue Box, piccoli apparecchi con cui scroccare telefonate a gratis per tutto il mondo. Tra i tanti, si diceva, c'era anche un ingegnere della HP, Alan Baum, che si era portato dietro un amico e collega, tal Stephen Wozniak. Insomma, anche se nessuno ancora se lo immaginava, quello era "il più bel raduno di ingegneri e tecnici che fosse stato possibile mettere sotto lo stesso tetto" (2). Dopo quel primo meeting ne seguirono tanti altri, a cui parteciparono sempre più persone, tutte con la stessa fissa: fare hacking sull'hardware, costruirsi (home brew, farsi in casa) il proprio computer. Ma facciamo un passo indietro: cosa significa fare hacking?

2. Hacking ed etica hacker

I termini hack, hacking ed hacker diventano famosi con l'informatica, anche se la loro origine è esterna: con l'espressione "good hack" veniva generalmente indicato un "bel lavoro", un lavoro "ben fatto", nel mondo dei lavoratori manuali. Quando quando si sviluppa l'informatica moderna, negli Stati Uniti del secondo dopoguerra, un "good hack" era riuscire, per esempio, a fare una procedura come meno istruzioni possibili o riuscire a far fare ad una macchina cose che non erano state minimamente previste. Un atteggiamento, insomma, che obbligava ad un certo tipo d'approccio: intanto che quello che si faceva doveva essere di pubblico dominio, che chiunque avrebbe dovuto poterci mettere le mani per capire cosa faceva e come; e che le macchine – che a quei tempi erano enormi blocchi di acciaio e cavi elettrici e vavolone grossi come tante stanze messe insieme per un costo di svariati milioni di dollari – dovevano essere a disposizione di chi voleva metterci le mani per programmare, sperimentare, modificare e quant'altro un good hack prevedesse. Non sempre le autorità preposte erano dello stesso avviso; raramente la cosa ha provocato grossi problemi etici, e tanto meno pratici, a questi primi hacker. L'esplorazione dei sistemi, la possibilità di migliorarli, di modificarli, di trovare nuove vie per raggiungere un obiettivo, ma anche di trovare nuovi scopi a strumenti nati per tutt'altro, ecco, questa è sempre stata la tendenza di base di ogni buon hacker – fosse esso un informatico, un boscaiolo, un contadino. Fino alla fine degli anni '60 la scena hacker era "rinchiusa" nel recinto, comodo ma angusto, dei campus universitari americani – soprattutto il MIT di Boston (il mitico Massachusetts Institute of Technology, in cui si sviluppò gran parte della prima scena hacker). E' qui che si sviluppa quella che verrà comunemente conosciuta come etica hacker, canonizzata dal giornalista Steven Levy nel suo classico libro Hackers. Gli eroi della rivoluzione informatica del 1984, e che consiste in alcuni punti fondamentali: 1. L'accesso ai computer – e a tutto ciò che potrebbe insegnare qualcosa su come funziona il mondo – dev'essere assolutamente illimitato e completo. Dare sempre precedenza all'imperativo di metterci su le mani! Secondo Levy, occorre permettere agli hacker di esaminare sistemi già esistenti perché possano capirli e conoscerli; in questo modo sarà possibile sia migliorare i detti sistemi, che permetterne la creazione di nuovi. L'accesso permette lo sviluppo di nuova tecnologia. 2. Tutta l'informazione dev'essere libera. Le idee e informazioni sono necessarie per migliorare, aggiustare e creare nuovi sistemi; per questo devono essere libere. Secondo

il punto di vista hacker, ogni sistema può beneficiare di un libero scambio di informazioni. 3. Dubitare dell'autorità. Promuovere il decentramento Il miglior metodo per promuovere un libero scambio di informazioni è quello di avere un sistema aperto, privo di ogni ostacolo tra un hacker e quello che sta cercando di imparare. Per questo motivo gli hacker sono contrari a ogni forma di burocrazia, che ritengono non abbia nessun'altra funzione se non quello di rallentare la conoscenza. 4. Gli hacker dovranno essere giudicati per il loro operato, e non sulla base di falsi criteri quali ceto, età, razza o posizione sociale In una comunità hacker, l'abilità conta più di ogni altra sua caratteristica. Levy descrive il caso di L. Peter Deutsch che, sebbene appena dodicenne, venne accettato dagli altri hacker del TX-0 nonostante non fosse neppure uno studente del MIT. 5. Con un computer puoi creare arte Una degli aspetti maggiormente apprezzati dagli hacker è quello di realizzare software anche molto complesso funzionante con il minor numero di istruzioni possibile. Il codice sorgente di un programma, se particolarmente curato e ben scritto, è considerato un'opera d'arte. 6. I computer possono cambiare la vita in meglio Gli hacker considerano i computer come parte integrante delle loro vite, avendogli donato obiettivi e avventure: sono considerati come lampade di Aladino che essi possono controllare. Per questo pensano che chiunque possa beneficiare di essi, e che, se venissero utilizzati nel modo giusto, l'etica hacker si espanderebbe nel mondo, migliorandolo (3). E' evidente come questa etica sia figlia del contesto in cui nasce – quello delle università e della cultura scientifica, dove una teoria per essere accettata deve essere trasmessa a tutti i ricercatori del settore che devono poterla studiare liberamente, smontare, analizzare e, se necessario, modificare.

3. La contestazione

Con la fine degli anni '60 la contestazione studentesca e giovanile attraversa tutti gli Stati Uniti, soprattutto le università ma non solo. Ma all'inizio il mondo dell'informatica è visto con sospetto se non con aperta ostilità dal movimento. L'informatica, infatti, nasce direttamente all'interno degli apparati militari e di spionaggio durante la seconda guerra mondiale, apparati che continueranno a sostenerla e finanziarla fino all'avvento di internet (compresa: il progetto da cui nasce internet – Arpanet – venne studiata e realizzata dal DARPA, l'agenzia del Dipartimento della Difesa degli Stati Uniti) (4). Il '68 americano, quindi, nasce e si sviluppa anche e soprattutto come rifiuto e lotta contro la guerra nel Vietnam e le università scientifiche statunitensi, ed in particolare quelle dove si stava sviluppando il mondo dell'informatica e dei computer, erano in prima fila nello sforzo bellico del governo e dei militari, che finanziavano queste ricerche. Fu solo nei primi anni '70 che parte del movimento americano iniziò ad interessarsi in maniera positiva di informatica, e ciò non poteva che avvenire a Berkeley, in California. Il primo progetto fu di un gruppo che si chiamava Community memory, che piazzò un terminale collegato ad un mainframe (un grande computer aziendale) in uno dei punti di maggior accesso del movimento californiano con lo scopo di permettere "un sistema di comunicazione che educava la gente a prendere contatto con gli altri sulla base di interessi condivisi, senza dover sottostare al giudizio di una terza parte" (5). Un progetto di comunicazione decentrata, quindi, dove il computer, per la prima volta,

andava "al popolo", per aiutarlo a comunicare, vivere e lottare liberamente. Uno dei fondatori del Community memory era quel Lee Felsenstein che abbiamo trovato alla prima riunione del Homebrew Computer Club...Un'altra organizzazione che fece proselitismo per i computer nel movimento californiano fu la People's Computer Company, il cui fomentatore era quel Bob Albrecht che abbiamo trovato alla prima riunione del Homebrew Computer Club... e che aveva come scopo di rivelare alla gente, e specialmente ai bambini, il "verbo" del computer. Fu in questo brodo, nel mondo della Community memory e del PCC, che nacque l'idea dell'Homebrew Computer Club, grazie all'idea di due frequentatori frustrati di questi luoghi, Fred More e il già citato Gordon French, ospite del primo meeting, che decisero di proporre una riunione a persone che volessero costruirsi il loro computer.

4. I due Steve

Steve Wozniak è il classico esempio di geek, l'appassionato di tecnologia che fin da ragazzo passa ore e ore a disegnare strani circuiti, a progettare improbabili computer e cose così. Fin dall'adolescenza riesce a passare molto tempo sui computer – vivere nella Silicon Valley ha sicuramente aiutato lo sviluppo di questa passione. Una volta diplomato Woz – così lo chiamavano gli amici – il nostro proverà a fare l'università, a Berkeley, e fu qui che sentì parlare per la prima volta delle blue box, quei simpatici aggeggi inventati da John "Captain Crunch" Draper (6) di cui s'è già parlato. Per Woz le blue box erano simpatici oggetti con cui fare del buon hacking ma anche con cui divertirsi in scherzi esilaranti (come spacciarsi per Henry Kissinger per arrivare a parlare col Papa...). Per Jobs, invece, questi apparecchi erano la possibilità di far soldi, e così fu: "fu però Steve Jobs a rendere questo passatempo una fonte di guadagno: nel giro di poco tempo, fecero affari d'oro vendendo agli studenti le blue box" (7). Questo episodio ci dà il primo scorcio di quello che sarà una costante nel rapporto tra i due Steve: Woz, il genio dell'informatica, l'hacker ironico e burlone; Jobs il mediocre ingegnere, ma abile business man per il quale ogni gioco dell'amico può diventare fonte di soldi e fama. Alla fine Woz non finirà l'università e andrà a lavorare per l'Hewlett-Packard, una delle più importanti aziende high-tech degli Stati Uniti. Qui ritroverà il suo amico Baum e una sua dimensione di felicità: di giorno lo stimolante lavoro in una delle più importanti aziende informatiche del paese, nel tempo libero l'hacking per se stesso e per gli amici, e soprattutto i giochi elettronici che tanto gli piacevano. Aiutò spesso anche l'amico Jobs, che era andato a lavorare per l'Atari, dove questi progettava, appunto, giochi. E' a questo punto che Woz – con l'amico Baum – iniziò a frequentare l'Homebrew Computer Club, un ambiente dove stava coi "suoi simili", dove si parlava e ci si confrontava sui suoi argomenti, dove c'era gente come lui che voleva costruirsi il proprio computer!

5. L'Apple I

"Wozniak non stava pensando a costruire un computer da vendere. Voleva costruire un computer con cui divertirsi e da mostrare ai suoi amici. Raccontò quel che stava facendo al suo amico Steve Jobs all'Atari, il quale si interessava di terminali e pensava di mettere su una compagnia che li producesse. Ogni quindici giorni Woz andava all'Homebrew Computer Club a vedere o sentire cosa c'era di nuovo e non incontrava problema a ottenere

dettagli tecnici perché le informazioni venivano scambiate liberamente". Fu così che Woz si mise a costruire il suo computer, che doveva avere lo schermo, la grafica a colori e doveva contenere il BASIC, il linguaggio di programmazione sviluppato a partire dal 1963 nell'Università di Dartmouth. Dopo un lungo lavoro riuscì nel suo intento, anche se quello che aveva prodotto non era un vero e proprio computer ma una scheda piena di chip e circuiti. Attaccandogli monitor, tastiera e registratore a cassetta, si avrebbe avuto un vero e proprio computer. E fu proprio questo "aggeggio" che Steven Wozniak presentò una sera al Homebrew: in breve tempo tutta la sala era intorno a lui e al suo computer, a far domande e ad offrire proposte e soluzioni. «Quando Wozniak portò il suo risultato all'Homebrew e iniziò a distribuire fotocopie del suo progetto (in modo che gli altri potessero duplicarlo), tutti gli altri membri del club rimasero stupiti. Wozniak chiamò questa sua creatura "Apple"» (8). Tutto il lavoro di Woz era a disposizione di tutti, tanto che «chiunque all'Homebrew poteva dare un'occhiata ai diagrammi del progetto, il BASIC di Woz (perché se ne era scritto uno da solo ...) era stato distribuito gratuitamente insieme all'acquisto di un pezzo dell'attrezzatura che connetteva il computer al registratore a cassetta e Woz pubblicò, su riviste come "Dr. Dobbs" [...]. Gli annunci della Apple dicevano persino "la nostra filosofia è fornire software per le nostre macchine gratuitamente o a un costo minimo"» (9). Fu però l'altro Steve, Jobs, a fiutare l'occasione fornita da quella macchina per hobbysti: ed iniziò ad organizzarsi, con Woz, per far sì che l'Apple I diventasse un computer pronto all'uso, e non uno dei tanti giocattoli per smanettoni, in cui la maggior parte del divertimento sta nel tentare di montare l'attrezzo piuttosto che di usarlo. Dal successo dell'Apple I «Steve Jobs, però, vide in quella macchina un'ottima occasione per fondare una società. Wozniak era sfavorevole: l'idea di trasformare il suo passatempo in un business gli dava un po' fastidio» (10).

6. L'Apple II

«Era la fertile atmosfera dell'Homebrew che fece da guida a Steve Wozniak mentre preparava l'Apple II. Lo scambio di informazioni, l'accesso a esoterici espedienti tecnici, la turbinante energia creatrice e la possibilità di far andare fuori di testa chiunque con un progetto o un programma ben hackerato ... questi incentivi potevano soltanto aumentare il desiderio già intenso che Steve Wozniak aveva: costruire il genere di computer con cui avrebbe voluto giocare» (11). L'Apple II fu un vero e proprio successo, tanto che Levy nel suo fondamentale Hackers lo definisce "il più importante computer della storia" (12). Costruito grazie all'Homebrew, dove Woz poteva trovare soluzioni ed idee tutte le volte che finiva in un vicolo cieco nella costruzione del suo gioiello. Fino a quando, nel dicembre del 1976, fu pronto a presentare la sua macchina ad una riunione del club. Fu sul successo di queste due prime macchine che Jobs costruì la Apple come azienda, avendo la grandissima intelligenza di circondarsi di manager di grande esperienza che credevano nel progetto e trovando finanziatori adeguati. E fu per questo che chiese a Woz di lasciare l'HP per lavorare a tempo pieno alla Apple. Accettò a fatica, e per lui fu una specie di sconfitta: «anche se diede il suo consenso a lasciare l'HP e lavorare con Jobs a tempo pieno, confidò a se stesso che quel che stava facendo non era più hackeraggio puro. La

verità era che fondare un'azienda non aveva nulla a che vedere con l'hackeraggio o con la progettazione creativa. Era qualcosa per fare soldi. Era "passare il confine", come diceva Wozniak; [...] "non c'era modo che io associassi nella mia testa la Apple con il fare un buon progetto di computer. Non era questa la motivazione per creare la Apple. La ragione, dopo aver progettato il computer, era un'altra: far soldi!"» (13).Ma nonostante tutto Woz avrebbe continuato a lottare per far sì che i suoi progetti, le sue macchine, fossero compatibili con la sua etica hacker.«L'architettura "aperta" della macchina doveva essere un aiuto; conformemente all'etica hacker, Woz si assicurò che l'Apple non avesse segreti che impedissero alle persone di esprimerne con esso tutta la propria creatività. Ogni particolare del suo progetto, ogni trucco nel codice nel suo interprete BASIC sarebbe stato documentato e distribuito a chiunque lo volesse vedere» (14).Insomma, per Woz, figlio dell'etica hacker, la cosa più importante era poterci mettere le mani dentro. E non è un caso che, nonostante il successo, quando Apple si trasferì nella grande sede in Bandley Drive a Cupertino, all'ingresso del nuovo edificio sventolava la Jolly Roger, la bandiera dei pirati... (15)

7. Il Software Libero

Il 1985 è un anno particolare per il nostro racconto: è infatti l'anno in cui Jobs lascia (momentaneamente, vedi il paragrafo successivo) la Apple per uno scontro interno all'azienda; ma è anche l'anno in cui Richard M. Stallman (16), uno dei maggiori hacker del MIT, decide di fondare la Free Software Foundation, una fondazione ormai diffusa a livello mondiale, il cui scopo è la produzione e la diffusione di software la cui licenza permetta di utilizzarlo in maniera completamente libera. I quattro principi del software libero sono:1. Libertà 0 Libertà di eseguire il programma per qualsiasi scopo. 2. Libertà 1 Libertà di studiare come funziona il programma e adattarlo alle proprie necessità. L'accesso al codice sorgente ne è un prerequisito.3. Libertà 2 Libertà di ridistribuire copie del programma in modo da aiutare il prossimo. 4. Libertà 3 Libertà di migliorare il programma e di distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio. L'accesso al codice sorgente ne è un prerequisito (17).Come si evince chiaramente dalla lettura delle "4 libertà del software libero", esse sono strettamente legate a quella che fin'ora abbiamo chiamato "etica hacker". Cioè quell'etica che è stata, fin dall'inizio, il brodo di cultura di Wozniak e della nascente Apple.

8. Oggi

A che punto siamo, 30 anni dopo? Apparentemente non solo tutto è continuato sulla stessa lunghezza d'onda, quella dell'etica hacker, ma le cose sembrerebbero migliorate.Dal 2000, infatti, la Apple – dopo il ritorno in auge di Jobs precedentemente uscito dall'azienda nel 1985 – ha virato completamente verso il mondo Unix (18) con l'uscita del sistema operativo Mac OS X (19) il cui "cuore" (20) è un derivato diretto della famiglia BSD (21). Una scelta, quella del mondo Unix, fatta all'insegna del software libero.Molte delle componenti principali del sistema operativo Mac OS X, ancora oggi, sono frutto diretto della comunità del free software (oltre al "cuore", come si è detto sopra, che è un prodotto della Apple): dal sistema di stampa a quello per condividere i file in ambito Windows; dal server Web

alla shell; dal motore di rendering di Safari alle librerie OpenGL. Questo ha permesso alla Apple di poter usufruire del lavoro della comunità e dei suoi prodotti di successo. Apparentemente Apple ha dato in cambio molto, "rendendo" alla comunità molte migliorie dei software che ha deciso di usare sui suoi sistemi operativi. Non fosse che oggi il mercato principale di Apple non è più quello dei computer, ma quello dei cosiddetti device, cioè quei dispositivi nati per fare alcune cose – il lettore di file audio, il lettore di file video, il telefono cellulare – ma che alla fine ne fanno tante altre, proprio come un computer, ma che si possono portare in tasca. Questi device si chiamano iPod, iPhone e il recentissimo iPad, ed hanno una caratteristica, una pessima caratteristica in comune: sono assolutamente chiusi. Tutti e tre questi device, infatti, non solo prodotti esclusivamente con software proprietario (22), ma non permettono tutta una serie di operazioni ai loro possessori: essi bloccano completamente il software libero. Gli sviluppatori devono pagare una tassa ad Apple, che diventa l'unica autorità che decide ciò che può e non può essere installato sul device di ogni persona; essi approvano e supporta il Sistema di Gestione delle Restrizioni Digitali (DRM) (23); essi tramite il GPS scoprono dove ti trovi e dà ad altre persone la possibilità di rintracciare la tua posizione senza che tu te ne accorga (24); non riproducono formati liberi da brevetto e DRM come Ogg Vorbis e Theora; essi contengono il cosiddetto kill switch, che altro non è che una connessione effettuata a insaputa del proprietario, verso un sito Apple che decide se far avviare o no le Applicazioni (di terze parti e non) sul device (25); non sono l'unica possibilità. Ci sono migliori alternative all'orizzonte che rispettano la tua libertà, non ti spiano, riproducono formati multimediali liberi, e ti permettono di usare software libero (26). La cosa assolutamente sconcertante dei nuovi device della Apple è che sono tutti il contrario di quello che è stata la sua storia fin dalle origini: dare alla comunità degli strumenti aperti, che l'Apple non avesse segreti che impedissero alle persone di esprimerne con esso tutta la propria creatività (27). Oggi, invece, non solo l'utente non può esprimere la propria creatività, coi dispositivi della Apple, ma non può neanche installare i software che preferisce – perché deve prenderli solo ed esclusivamente dall'Apple Store – non può sentire e condividere i propri file multimediali liberamente, non può essere sicuro di avere sotto controllo la propria privacy. E non è tutto: anche gli autori di contenuti si trovano sotto il la spada della censura di casa Apple. Censura, sì, non stiamo esagerando: è recente il caso di una rivista di moda che si è trovata dover cancellare i capezzoli di Madonna – la celebre pop star – da una foto della sua rivista per iPad, altrimenti l'applicazione non sarebbe potuta entrare nell'Apple Store (28). Come dice il redattore di Gizmondo – la rivista online da cui prendiamo la notizia – «davvero una brutta cosa dettare politiche che portano alla censura arbitraria delle applicazioni. La censura è una bestia insidiosa che dilaga molto facilmente. Invece di decidere voi quali nudità, quale linguaggio o quali sentimenti politici sono accettabili, perché non lasciare scegliere ai vostri utenti che non sono un manipolo di idioti, né di incapaci e non hanno bisogno di una madre o di un padre a Cupertino. Apple dovrebbe concentrarsi sul modo migliore di far comunicare le persone, non su quello che queste persone possono o non possono comunicare (29)». Sicuramente Apple non è l'unica azienda che si comporta in questa maniera (Google, da molti punti di vista non è da meno, e in altri è sicuramente peggio) (30); ma un'azienda nata come pilastro commerciale

dell'etica hacker, che di questa etica, del suo mondo, dei suoi frutti, ha tratto gran parte della propria fama e prosperità, oggi si trova sul fronte opposto: quello del controllo, della repressione (la vicenda di cui si tratta in questo libro è esemplare, ma non è l'unico caso. E' recentissima la vicenda di un ingegnere della Apple che viene licenziato perché ha mostrato l'anteprima dell'iPad G3 a... Wozniak! (31), della censura.

NOTE

1) Da Phreaking, cioè quell'attività svolta da persone che vogliono studiare ed esplorare i sistemi telefonici. Il termine è l'unione delle parole phone (telefono) e freak (persona bizzarra). Questa attività è parente strettissima dell'hacking...

2) Gran parte di quanto scritto qua deve molto, se non tutto, al fondamentale testo di [Steven Levy, Hackers. Gli eroi della rivoluzione informatica, Milano, Shake Edizioni, 1994](#). In questo caso pag. 159.

3) http://it.wikipedia.org/wiki/Etica_hacker

4) <http://it.wikipedia.org/wiki/Arpanet>

5) Steven Levy, Hackers, cit., pag. 123

6) [http://www.wikiartpedia.org/index.php?title=Draper_John_T._"Captain_Crunch"](http://www.wikiartpedia.org/index.php?title=Draper_John_T._)

7) <http://www.macitynet.it/immagini/speciali/applestory2.shtml>

8) <http://www.macitynet.it/immagini/speciali/applestory3.shtml>

9) S. Levy, Hackers, cit. pag. 199

10) <http://www.macitynet.it/immagini/speciali/applestory3.shtml>

11) Levy, Hackers, cit. pag. 200

12) Ivi

13) Ivi, pag. 204

14) Ivi, pag. 205

15) Cfr. Owen W. Linzmayer, Apple Confidential 2.0 – The Definitive History of the World's Most Colorful Company. No Starch Press, 2004. «Symbolizing Jobs' defiant attitude and Apple's internecine rivalry was the Jolly Roger that flew over the Mac team's newest building Bandle III (10460 Bandle Drive)»

16) <http://it.wikipedia.org/wiki/Stallman>. Vedi anche <http://www.stallman.org/>

17) <http://www.gnu.org/philosophy/free-sw.it.html>

18) Unix è un sistema operativo sviluppato a partire dal 1969. Di questa famiglia fanno parte, tra gli altri, i sistemi operativi della famiglia *BSD e quelli della famiglia GNU/Linux. Cfr. <http://it.wikipedia.org/wiki/Unix>

19) http://it.wikipedia.org/wiki/Mac_OS_X

20) In informatica, il kernel costituisce il nucleo di un sistema operativo. Si tratta di un software avente il compito di fornire ai processi in esecuzione sull'elaboratore un accesso sicuro e controllato all'hardware. Cfr <http://it.wikipedia.org/wiki/Kernel>

21) L'acronimo BSD (Berkeley Software Distribution) indica la variante originaria di Unix sviluppata presso l'Università di Berkeley in California e alla base di una delle due famiglie principali di sistemi operativi liberi attualmente più diffusi Cfr. <http://it.wikipedia.org/wiki/BSD>

22) Il software proprietario è quello che non è libero o semilibero. Il suo utilizzo, la redistribuzione o modifica sono proibiti o richiedono un permesso o sono sottoposti a tali vincoli che in pratica non si possono fare liberamente. Cfr <http://www.gnu.org/philosophy/categories.it.html#ProprietarySoftware>

23) I DRM attaccano la nostra libertà su due livelli. Il loro scopo è quello di attaccare la nostra libertà limitando l'utilizzo di copie di opere pubblicate. Ciò significa forzarci a usare software proprietario e questo comporta che noi non abbiamo il controllo di ciò che questo fa. Quando le aziende si organizzano per progettare prodotti che ci limitano, allora noi dobbiamo organizzarci per sconfiggerli, come ha dichiarato recentemente Richard M. Stallman. cfr. <http://www.fsf.org/it/may-4-day-against-drm-ita>

24) <http://www.webnews.it/news/leggi/13183/apple-vede-registra-e-condivide-la-tua-posizione/>

25) Cfr. c'è una quantità di informazioni in rete su questa "caratteristica" dei device della Apple, così come c'è una quantità di informazioni per disabilitarla. E' interessante notare che nel momento in cui la si disabilita, decade automaticamente la garanzia del produttore...

26) Cfr. <http://www.fsf.org/it/le-cinque-ragioni-contro-l-iphone>

27) Cfr. nota 14

28) <http://www.gizmodo.it/2010/05/12/la-censura-di-apple-sulle-applicazioni-non-ha-piu-freni.html>

29) Ivi. Il grassetto è nell'originale.

30) Il controllo esercitato da Google su internet attraverso il suo motore di ricerca e tutti i vari servizi web che offre è sicuramente un argomento interessante e che potrebbe essere messo in relazione con le vicende sopra

narrate, anche se non è questo l'argomento di questo testo. Non è superfluo sottolineare, però, che con l'entrata di Google nel mondo dei device, alcune pratiche di Apple che stiamo denunciando vengono riproposte pari pari dal gigante di Mountain View, come quella del kill switch di cui s'è già parlato nel

testo.

Cfr.

<http://www.itespresso.it/google-rimuove-due-applicazioni-dagli-smartphone-and-roid-46658.html>

31)

<http://www.macitynet.it/macity/articolo/Ingegnere-Apple-mostra-iPad-in-anteprima-a-Wozniak-e-viene-licenziato>

Ebooks e Software Libero (e Open Source) – Seconda parte: quali strumenti?



In questa seconda parte di questo ragionamento sul mondo degli Ebook andremo a vedere quali sono i principali [formati](#) usati per produrre libri digitali e con quali strumenti, a loro volta liberi, possiamo crearli.

La maggior parte degli [Ebook Reader](#) odierni permette di visualizzare i formati più comuni per personal computer:

- [ASCII](#);
- [Microsoft Compressed HTML Help](#) (CHM);
- [HTML](#);
- Ms Word (.doc);
- [PostScript](#);
- [RTF](#);
- [TeX](#);
- [PDF](#).

Alcuni di questi formati sono molto famosi, altri meno; alcuni sono liberi, altri no. La caratteristica comune a tutti questi formati è che **non** sono nati per essere visualizzati su un ebook reader. molti di questi formati possono essere convertiti in uno o più formati specifici per ebook reader, per cui anche per l'utente meno smaliziato è possibile passare da un formato

all'altro o con degli [script](#) trovati in rete o usando programmi grafici per la conversione.

I formati specifici per essere letti sugli ebook reader sono:

- [ePub](#), formato aperto, evoluzione di [OeB](#), che si sta affermando come standard più diffuso
- [Mobipocket](#) (mobi), formato chiuso, utilizzato esclusivamente da [Amazon Kindle](#);
- [FictionBook](#) (fb2);
- [LIT](#), formato nativo Microsoft, ormai obsoleto;
- [PDB](#), formato per [Palm OS](#).

In questo articolo ci occuperemo solo del formato aperto [ePub](#), che come dicevamo sopra è ormai lo *standard de facto* e di cui ci si aspetta entro l'anno il rilascio della [versione 3.0](#). Ad ora siamo alla [versione 2.0.1](#), rilasciata nel 2007 dall'[International Digital Publishing Forum](#).

Gli strumenti descritti in questo articolo, così come in quasi tutti quelli raccolti in questo blog, hanno la caratteristica di essere rilasciati sotto una [licenza libera](#), e quindi di essere usati nella massima libertà, fino a poterli modificare e rivendere. Per fare questo useremo alcuni testi, liberamente scaricabili dalla rete e rilasciati sotto licenza [Creative Commons](#), una licenza libera per i testi, ormai molto diffusa e non solo per argomenti tecnici, ma anche per romanzi (i più famosi, in questo caso, sono il collettivo letterario [Wu Ming](#), che dal 1999 pubblica tutti i suoi romanzi e racconti sotto licenza libera Creative Commons, pur pubblicando con un grossissimo editore come [Einaudi](#); ma non sono gli unici):

- [Editoria digitale, di Letizia Sechi, Apogeo Editore](#);
- [L'Umanista Informatico, di Fabio Brivio, Apogeo Editore](#).

E' importante premettere che in questi libri non si parla specificatamente di *software libero* o *open source*; anzi, nella maggior parte dei casi gli esempi riportati lo sono attraverso l'uso di software proprietario. **Ma** i linguaggi di cui si parla, quelli che a noi interessano e che sono il cuore di un file ePub – l'xml, l'xhtml, i css – sono linguaggi liberi, nel senso del software libero, liberamente utilizzabili e, ormai, standard tanto nella comunicazione web quanto nella formattazione e produzione di contenuti, anche multimediali. Quindi, ovviamente, realizzabili anche e soprattutto attraverso l'uso di strumenti completamente liberi.

Libri elettronici: cosa sono e come sono fatti

Come dicevamo sopra i libri elettronici non sono altro che file, di tanti tipi, e qui noi ci occuperemo solo dello standard, l'**ePub**.

Un file con estensione .epub altro non è che un file zip rinominato, al cui interno troviamo due cartelle (directory) ed un file:

META-INF/

OEBPS/

mimetype

Quest'ultimo file deve esserci **sempre**, e il suo contenuto è una stringa, questa sotto:

```
application/epub+zip
```

La cartella **META-INF** è fondamentale: in sua assenza una pubblicazione ePub non potrebbe funzionare. Deve quindi essere sempre presente insieme e sullo stesso livello della cartella OEBPS e del file mimetype, e non può avere altro nome.

Questa cartella deve contenere almeno un file, `container.xml`. Inoltre qui possono essere presenti altri file opzionali, i cui nomi sono comunque riservati e non modificabili.

La cartella META-INF può quindi avere il seguente contenuto, sebbene nella pratica al suo interno venga posizionato di norma il solo file `container.xml`:

```
META-INF/
```

```
container.xml
```

```
manifest.xml [opzionale]
```

```
metadata.xml [opzionale]
```

```
signatures.xml [opzionale]
```

```
encryption.xml [opzionale]
```

```
rights.xml [opzionale]
```

Infine la cartella **OEBPS**, che eredita il suo nome dalla specifica Open eBook Publication Structure che, rilasciata nel 1999 dall'Open eBook Forum, ha costituito la base delle specifiche ePub.

Il nome "OEBPS" non è riservato, cioè questa cartella potrebbe avere qualsiasi altro nome, ma ormai è diventato una convenzione diffusa. Ogni creatore di ePub è comunque libero di rinominarla come ritiene più opportuno o comodo, nel rispetto però delle regole sintattiche che valgono per tutti i nomi di file e cartelle di una pubblicazione ePub.

Questa cartella contiene tutti i file che compongono e organizzano la pubblicazione:

i file di testo;

i file delle immagini;

il file del foglio di stile CSS;

i file dei font eventualmente inclusi;

il file dell'indice NCX;

il file OPF.

Questa lunga citazione dal libro di [Fabio Brivio, Giovanni Trezzi, ePub per autori, redattori, grafici, Apogeo, 2011](#), ma qualcosa di molto simile lo trovate anche nel già citato libro di Letizia Sechi, per capire con cosa ci si trova a lavorare quando si maneggia un ePub: roba standard e libera, file [.zip](#), [.xhtml](#) e [css](#). Tutta roba che si può maneggiare tranquillamente con qualsiasi sistema operativo, che nasce libera, aperta, standard. Non fatevi fregare!

Desktop Publishing: Scribus. E il LaTeX?

Diciamolo subito, così ci togliamo il pensiero: la stragrande maggioranza di coloroso che si sono buttati o si butteranno nel *business* dell'editoria digitale lo faranno usando gli strumenti tipici dei grafici o, ad essere ottimisti, dei tipografi. Quindi strumenti [WYSIWYG](#), cioè strumenti grafici di sviluppo che permettano – come dice l'acronimo – *ottenere quello che vedi*. Il più usato ad ora è Indesign dell'Adobe; nel mondo del software libero l'equivalente è il noto [Scribus](#). Purtroppo, per ora, questo potente strumento di publishing non esporta i suoi lavori in ePub, ma "solo" in Pdf... Vi accontenterete... ☐

Un'altra soluzione che ho cercato è stata quella del classico e potentissimo [LaTeX](#), straordinario linguaggio di [markup](#), molto utilizzato in ambito di pubblicazioni scientifiche (l'ho usato per fare la mia tesi di laurea... in storia! :). Leggendo un po' in rete, però, ho scoperto che la cosa, anche se fattibile, non è consigliabile. Infatti:

tutti i file generati dai compilatori TeX (dvi, pdf) sono a dimensione di pagina fissa, e il compilatore usa questo fatto pesantemente. Il formato epub, invece, adatta il testo alle dimensioni della pagina, come l'html. Non solo, ma il font con cui visualizzare il testo (tipo, dimensione, ecc.) lo sceglie l'utente e non l'autore.

Chiaramente alcune delle cose che oggi si fanno in LaTeX sono impossibili da produrre in formato epub: pensa soltanto ai riferimenti di pagina.

Riporto questa [citazione](#) per sgomberare il campo da speranze che, purtroppo, andrebbero deluse. Ma tant'è, non rimane che buttarsi su altre soluzioni

Non rimangono tante soluzioni. La più scomoda sarebbe quella più amata dagli spippoloni, dagli *hackers*, e cioè quella di farsi tutto a mano. Ma visto che noi siamo pigri, ed abbiamo di meglio da fare per passare il tempo (ogni riferimento è puramente casuale), vorremmo poterci dedicare al

perfezionamento del risultato finale, ma poterci avvantaggiare con alcune comodità di inizio lavoro. Ecco perciò che andiamo a cercare strumenti che ci permettano di produrre il più semplicemente possibile un buon ePub, usando strumenti liberi, ma di buona qualità. Il sistema migliore che ho trovato fino a questo punto sono la messa insieme di questi strumenti:

LibreOffice + l'estensione Writer2ePub + Sigil

Dove il primo altri non è che la nuova versione dell'ormai pacchetto office libero (il caro vecchio OpenOffice, [cambiato di nome](#) da che è stato acquistato dai cattivoni di Oracle); il secondo è un'estensione di Libre(Open)Office, che permette di convertire un documento di testo in ePub, mentre il terzo è un piccolo gioiellino del software libero, un editor per ePub **non grafico**, che ci permette di mettere a posto cioè che viene prodotto con i due strumenti di cui sopra.

Tratteremo, infine, di altri accessori simpatici, per far sì che il nostro ePub sia il più personalizzato possibile. Accessori, non fondamentali, ma simpatici (come il mitico [ePub Zen Garden](#)).

Ebooks e Software Libero (e Open Source) – Prima parte



l'Ebook Reader Leggo IBS

Alla fine – meglio tardi che mai – il fenomeno ebook è esploso anche nel nostro paese. Ormai tutti i principali negozi online offrono anche ebook, e sono nati piccoli ed agguerriti store specializzati in editoria elettronica, che stanno facendo veramente un buon lavoro, per offrire tutti i servizi necessari a che l'utenza possa destreggiarsi in un mondo ancora nuovo.

Il fenomeno ebook sta cambiando radicalmente le carte nella tavola dell'editoria, nel mondo e quindi anche da noi, ma la reazione della maggior parte degli editori tradizionali è stata quanto meno conservatrice: tra chi si è buttato sono ben pochi quelli che l'hanno fatto con un reale slancio innovativo. Tutti gli altri l'hanno fatto perché, in qualche modo, trascinati dal boom (quindi per i capelli), e la loro principale preoccupazione è stata (ed è) quella di fare il minimo sforzo per ottenere il maggior risultato. Quindi prezzi assurdamente alti (ci sono parecchi casi di libri cartacei che costano meno dell'equivalente elettronico; un bel testo su questa scabrosa faccenda è quello di [Antonio Tombolini](#) nel numero 8 del suo [Ebook Trend Weekly](#)), nessuna pressione sulle istituzioni perché venga parificata l'aliquota iva tra cartaceo (4%) e l'elettronico (20%), uso sconsigliato dei

DRM proprietari dell'[Adobe](#), dimostrando una folle paura del p2p, cioè del fatto che i libri elettronici – che alla fine non sono altro che banalissimi file – siano scambiati sulle reti “pirata” del [Peer to \(2\) Peer](#). Come se fosse un problema togliere ad un file il DRM dell'Adobe: [la rete è piena di guide](#), anche nella nostra lingua, per portare a termine l'operazione in maniera tutto sommato semplice.

Questo fenomeno dimostra, a mio modesto avviso, che le grandi case editrici, al pari delle etichette discografiche all'apparire degli mp3, non hanno capito nulla di quel che sta succedendo, non hanno saputo cogliere l'occasione e si troveranno ben presto tagliate fuori da un nuovo e diverso modo di fare e di fruire dei “libri”.

Quel che c'è da capire di fondamentale, a mio avviso, del fenomeno ebook, è che allo stato attuale chiunque può diventare facilmente editore di se stesso. Sempre che abbia qualcosa di valido (o anche solo di sensato) di proporre agli altri. Essendo un file, oltretutto mediamente leggero (anche meno di un singolo pezzo in mp3, spesso), la sua diffusione in rete è la cosa più semplice che si possa fare. Già oggi i principali negozi online specializzati in editoria elettronica offrono spazio, anche gratuito, all'*autopubblicazione* (una volta, neanche tanto tempo fa, si sarebbe chiamata *autoproduzione*...). Dopo di che c'è da promuoverlo, da farlo conoscere, da riuscire a rompere il muro dei media mainstream, che alla fine è diventato il lavoro principale, se non esclusivo, degli editori tradizionali al giorno d'oggi.

Quindi, tolte le capacità artistiche – nel caso della prosa o della poesia – o quelle intellettuali e di scrittura – nel caso della saggistica, quello che l'autore e l'autrice oggi deve imparare a fare è ne più ne meno che un buon prodotto digitale, cioè un ebook. Un file.

Per far ciò già ora sono sul mercato diversi prodotti [proprietary](#) e a pagamento (che non è necessariamente la stessa cosa), alcuni molto accessibili e anche ben fatti, altri cari e molto difficili da usare. Noi non ci occuperemo di questi, ma della considerevole quantità di prodotti [liberi](#) e pure gratuiti (che non è necessariamente la stessa cosa), coi quali è possibile produrre oggi un buon ebook.

[Habemus Android: una recensione dell'HTC Tattoo](#)



HTC Tattoo

Ebbene sì, dopo alcune disavventure con il mio primo smartphone – un Nokia 5230 – grazie ad un mio caro amico sono entrato in possesso di un [Android](#). Nello specifico l'oggetto è un [HTC Tattoo](#), quindi un Android 1.6, versione parecchio vecchiotta, ma tant'è ...

Al di là della questione basilare, cioè il fatto che uno smartphone **deve** essere un cellulare, e il Tattoo fa degnamente il suo lavoro, il motivo per cui una persona spende dei soldi – mica pochi – per un oggetto del genere è quello di essere il più possibile connessi alla rete e di potervi usufruire il più possibile agevolmente i servizi.

Nel mio caso i servizi che uso con maggior frequenza sono:

- **e-mail;**
- **i feed rss;**
- **la navigazione del web;**
- **la scrittura di note e documenti in ambito condiviso, quindi accessibile con qualsiasi device da qualsiasi posizione;**
- **i feed rss;**
- **twitter e i principali social network in generale;**
- **ma chiacchiera con GNU/Linux?**
- **varie ed eventuali, che la fame vien mangiando...**

Un'altra condizione fondamentale, nel mio caso, sarebbe quella di usare il più possibile [software libero](#), e in questo caso, nonostante Android sia, alla fine, una [distribuzione GNU/Linux](#), attorno a questo sistema operativo non c'è ancora un grosso movimento di rilascio di software, appunto, libero (sono pronto a correggere questo punto con vivo entusiasmo, ovviamente!). Quello che ho trovato fin'ora è il bel repository [Cute Android](#), che comunque contiene già un monte di roba. Purtroppo non sono ancora riuscito ad avere tutto quello che voglio come software libero; ma penso sia un traguardo raggiungibile...

Comunque andiamo a vedere cosa ho trovato, partendo dalle situazioni più *semplici*.

E-mail

Il client e-mail del tattoo è un ottimo prodotto: supporta tutti i principali protocolli – pop, imap ed exchange – e, soprattutto, supporta la crittazione della comunicazione attraverso il protocollo di rete [SSL/TLS](#). Volendo con questo strumento si può accedere a tutti i propri indirizzi, anche gmail, attivando i protocolli pop o imap nella webmail di Google; **ma**, come tutti sanno, Android è un prodotto di Google, quindi contiene al suo interno (quasi) tutti i client per accedere a (quasi) tutti i servizi del gigante di Mountain View, ad iniziare dal client mail.

La navigazione del web

Anche in questo caso, appena si accende il dispositivo, ci si trova a poter immediatamente navigare il web con un buono strumento. Anche in questo caso c'è il pieno supporto al protocollo SSL/TLS, quindi la possibilità di

navigare in maniera sicura. Ovvio che stiamo parlando di un oggetto che ha un monitor piccolissimo, appena 2.8 pollici, quindi navigare siti che non gestiscono una doppia grafica, normale e *mobile*, significa soffrire.

La scrittura di note e documenti in ambito condiviso, quindi accessibile con qualsiasi device da qualsiasi posizione

In questo caso si entra in un primo momento di sofferenza, se si considera la cosa con l'obiettivo di usare *esclusivamente* software libero. Per quel che riguarda la creazione di note, c'è solo l'imbarazzo della scelta, i prodotti sono tanti e tanti sono *liberi*. Già più complessa la situazione per quel che riguarda lo *sharing*, cioè la possibilità di condividere le proprie note con altri, o comunque di metterle online da qualche parte, così da potervi accedere anche da altri *dispositivi*, come pc o portatili o altri smartphone.

Sono assolutamente presenti client di notissimi servizi di *storage online*, come Evernote e Dropbox, che però, purtroppo, non sono *liberi* (cioè, per evernote un client libero c'è, ma per GNU/Linux e non per Android; anche se la speranza è sempre l'ultima a morire...). Alla fine ho accettato di usare quest'ultima soluzione, in attesa di sviluppi positivi.

I feed rss

Nella mia esperienza di *navigazione* del web – e parliamo di quasi tre lustri, mica paglia – il vero punto di svolta fu la scoperta dei [feed rss](#): la possibilità di poter seguire decine, centinaia di siti con pochi clic. Dici poco! Se poi i siti sono fatti bene, in poco tempo puoi leggere gli articoli per intero o quasi, vedere foto e pure video.

Oggi un buon client di feed rss deve anche permettere, in maniera semplice, di condividere il link di un feed, quindi di una notizia, sui principali social network, oltre che via mail o di salvare il riferimento all'articolo in una nota.

Infine, ma non meno importante, chi usa tanto un client desktop di feed rss, dopo poco si ritrova con un ricco corredo di file [opml](#), cioè un file xml con l'elenco di tutti i feed.

Ecco, un buon client di feed per il mio androido doveva fare tutto quanto sopra, ed in particolare l'importazione del file opml, che aspettava solo di essere trionfalmente accolto. Più facile a dirsi che a farsi, purtroppo.

Di *libero* ho trovato un programma, nel repository di cui sopra, [Sparserss](#), che ufficialmente faceva tutto quello che volevo. Peccato che non sia mai partito...

Per il resto ho trovato una quantità di programmi che mi facevano accedere, in maniera più o meno originale, all'account di [Google Reader](#), il servizio online di Google per leggere i feed; interessante, ma non esattamente quello che cercavo. Fino a che non sono incappato al programma *gratuito* (ma proprietario) [RssDemon](#), che fa egregiamente quello che cercavo.

Twitter e i principali social network in generale

Anche in questo caso la scelta non manca, se uno non si fa il problema del *software libero*. Purtroppo anche in questo caso ci sono svariate soluzioni – io alla fine, per comodità e facilità d'uso e ampiezza di soluzioni – ho scelto [Tweetdeck](#), che è tutt'altro che libero, ma almeno è gratuito e fa bene il suo porco lavoro: unire nello stesso software più Social Network. Che è quello che cercavo: poter avere nella stessa interfaccia più così sociali, così da poter mandare con un... colpo di dito lo stesso messaggio a tutt@ ☐

Dopo di ché, il buon androido offre mille soluzioni super articolate: dalla singola applicazione, magari quella ufficiale, a soluzioni miste a chissà che altro. Io, pigramente, non avendo trovato nulla di libero, dopo breve ricerca mi sono fermato a quel che meglio mi pareva e non sono andato oltre. Non ne avevo voglia. Se qualcuno ha da suggerirmi qualcosa, magari di libero, sono pronto a sperimentare! ☐

Ma chiacchiera con GNU/Linux?

Ovviamene sì, visto che stiamo parlando di GNU/Linux da entrambe le parti. Quello che ho trovato io è un semplice ed efficacissimo software, *Remote Notifier*, software libero ospitato da [Google Code](#), col quale è possibile ricevere le notifiche dello smartphone direttamente nella system tray del pc, sia esso GNU/Linux, Mac o Windows.

Varie ed eventuali, che la fame vien mangiando

Poi ci tornerò meglio, ma uno dei difetti del coso androido – ma mi hanno detto anche della maggior parte degli altri così smartphone – è la batteria: che appena ci fai qualcosa si prosciuga in un lampo. E quindi:

- disabilita l'aggiornamento automatico dei programmi;
- disabilita l'esecuzione in background dell'aggiornamento dei dati;
- disabilita (ed usa il meno possibile) GPS e wi-fi;
- quando non ti serve disabilita la connessione ad internet;
- installa un file manager e controlla ciclicamente che non sia partito qualche maledetto programmino a tua insaputa (cosa che accade, oh se accade!);

E già! Quando hai a disposizione migliaia di applicazioni a gratis e centinaia libere, che fai, non le provi?! E che l'hai preso a fare sto coso se non ci giochi un po'!?

E così mi sono installato l'orario dei treni, il file manager di cui sopra, il lettore di ebook (libero!) quello della Amazon per i mobi (il formato proprietario degli ebook di Amazon), il visualizzatore di pdf (libero!) e quello dei file di Libre/Open Office (libero!) e altre cianfrusaglie inessenziali ma divertenti, che ora non ricordo (il coso androido in questo momento è, ovviamente, in carica ...).

Di tutti quelli sopra, i due software che voglio qui segnalare, da bravo *malato* di libri quale mi diletto di definirmi, sono due software che permettono di accedere ai due principali social network dedicati alla

lettura: [aNobii](#) e [Goodreads](#).

Il primo è [Anobii Android App](#), che purtroppo non ho mai potuto provare, visto che gira su versioni androide dalla 2.1 in poi, mortacci loro; il secondo è [MyBookDroid](#), che invece gira dalla 1.6 in poi, e che oltre a permettere la catalogazione della proprio libreria, permette anche di sincronizzarsi con il proprio account su Goodread.com, il social network alternativo ad aNobii. Una bella cosa ☐

Di tutto un po', perché alla fine, come dicevo sopra, quando decidi di prenderti un oggetto del genere – cosa assolutamente inessenziale – allora lo usi. Con intelligenza, senza sventrarti il portafogli (almeno io, che tengo famiglia e il portafogli è pieno di foglio, ma poco di soldi), senza finire nel tunnel della dipendenza, ma lo usi.

Difetti

A voglia se ci sono!

- Beh, sicuramente quel che dicevo prima, e che ora ripeto: 'sti cosi hanno la batteria fatta di vetro. Usare con delicatezza!
- Controllare bene in rete la funzione base, quella *telefonica*, che spesso capita che 'sti cosi siano eccezionali mini-computer, ma telefoni ridicoli. E non va bene...
- L'ho già detto della batteria...?
- La **privacy**. Facciamo attenzione, con questi oggetti, qui il discorso si fa serio. Quando l'ho acceso la prima volta, il simpatico sistema operativo ha iniziato a chiedermi se volevo aggiungere i miei account dei social network, se volevo configurare la mia posta elettronica, etc etc. Tutto bello, tutto figo. **MA**, attenzione: il sistema operativo è *libero*, ma l'interfaccia che c'ha messo sopra HTC? E quante di quelle funzioni di cui parlo sopra sono libere o proprietarie? Chi lo sa dove finiscono i dati che noi tanto tranquillamente infiliamo qui dentro, e che il simpatico sistema mischia e ci offre in una sola, comodissima, interfaccia? Quando si usano questi piccoli computer – sono piccoli, appunto, e sembrano tanto dei banali cellulare, poco più di giocattoli, ma di fatto sono niente di più e niente di meno che **computer** – loro si connettono alla rete e mandano in giro pacchetti di dati con dentro... cosa?
- I soldi. E' facile non fare attenzione, quando si compra uno di questi simpatici cosini, tanto maneggevoli e *cool*. Se non si ha il profilo tariffario giusto, ci vuole un secondo a prosciugarsi la ricarica. Occhio!

Insomma: questi piccoli, simpatici e costosissimi oggetti, che ormai rappresentano [una fetta in continua crescita](#) della navigazione del web, che sempre più facilmente entrano nelle nostre tasche (per farvi uscire il danaro...), sono oggetti da usare con attivando due caratteristiche fondamentali, e tragicamente fuori moda oggi, di cui dovrebbe essere dotata qualsiasi persona: **buon senso** e **consapevolezza**.