

MAGAZINE **BEYOND**

beos - amiga - osx - linux - freebsd

FREE MAGAZINE - N.1 AGOSTO 2003



ANTEPRIMA
Zeta di YellowTab

Tutorial REBOL

DOSSIER

BeOS e l'ADSL



Giochi d'estate



EmuZone

AMIGA

Alpe Adria 2003

NEWS - SOFTWARE - BECOME - ANTEPRIME - GIOCHI - EMUZONE

Benvenuti a bordo!

Ce l'abbiamo fatta. Quello che avete tra le mani è il primo numero di un magazine interamente dedicato a BeOS e ad altri sistemi "alternativi". Cosa intendiamo, nel nostro gergo, per alternativi? Non ci nasconderemo certo dietro ad un dito, negando l'evidenza dei fatti. Microsoft Windows non è di certo il migliore sistema operativo presente sul mercato, almeno in senso assoluto, ma è il più diffuso e supportato. In una scena dominata dai prodotti della casa di Redmond, pochi altri si ritagliano piccole nicchie di mercato. Apple, per esempio, nonostante il suo ottimo Mac OSX sia per certi versi migliore di Windows, soffre. Ma è costretta a farlo per una precisa limitazione: gira solo sui PowerMac, obiettivamente meno diffusi di un classico PC x86. E gli altri? Linux non ha un mercato commerciale, ma si sta difendendo bene e sta avendo una fortuna forse insperata oggi, insieme al suo cugino FreeBSD. Nonostante la disfatta di Be, BeOS sembra risorgere dalle proprie ceneri come un'araba Fenice, grazie a OBOS, Cosmoe e tutti gli altri progetti Open Source che gli orbitano intorno. Ma se è vero che da un lato un sistema operativo open source può garantire buoni successi, un'accorta politica commerciale può garantire successi ancor più grandi. Ecco allora entrare in scena la tedesca YellowTab, che raccoglie l'eredità di Be e si appresta a commercializzare Zeta.

E neanche AmigaOS resta a guardare, giunto alla svolta con la versione 4.0 si appresta ad aggredire il mercato mondiale con un prodotto all'altezza delle aspettative e che supporta finalmente hardware all'avanguardia.

Ecco quali sono, secondo noi, i sistemi alternativi.

Ci piacerebbe darvi una buona quantità di informazioni, rendervi un servizio completo e preciso e ci auguriamo di cuore che il nostro progetto vi piaccia, cari lettori.

In questo primo numero vi presentiamo in anteprima Zeta, troverete un interessantissimo tutorial sul linguaggio Rebol (il cui padre è Carl Sassenrath, niente meno che lo storico autore di Exec, il kernel di AmigaOS!) e una sezione dedicata ai giochi con i quali trascorrere questa lunga e torrida estate. Nella speciale sezione dedicata all'emulazione, EmuZone, vi presentiamo Prince of Persia e la piattaforma ad 8bit MSX. Chiudiamo lasciandovi al reportage sull'incontro Amiga Alpe Adria 2003 e alle news dell'estate.

Buona lettura!

Gian Davide Alfano

BEYOND

free magazine per utenti di sistemi BeOS - Amiga - OSX - Linux - FreeBSD

Distribuzione gratuita

beyond.itbug.org - beyond@itbug.org

CAPOREDATTORE

Gian Davide Alfano

REDAZIONE

Francesco Baldi - Marco Begliardo
Andrea Bernardi - Giuseppe Gargaro
Davide Gessi - daf
Massimiliano Origgi
Andrea Scatena - Mattia Tristo

PROGETTO GRAFICO

Gian Davide Alfano

Becasso



NEWS

NEWS

Segnali, novità, curiosità dalle comunità di tutto il mondo **4**

RUBRICHE

BECOME

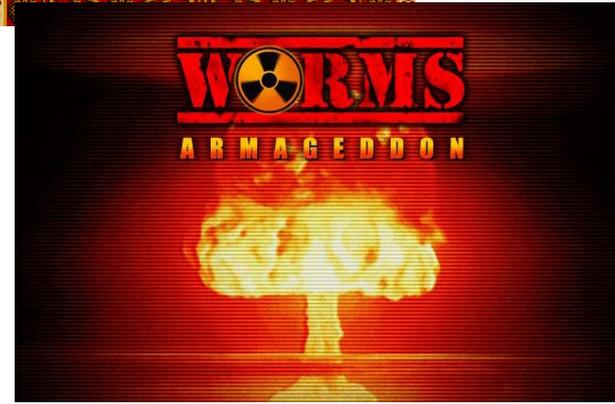
In questo numero ci occupiamo del **FileSystem** di **BeOS** e **OSX** **6**

DOSSIER

Configurare **BeOS** per navigare in Internet con l'**ADSL** **9**

ANTEPRIMA

Alcune indiscrezioni su Zeta, il nuovo sistema operativo di YellowTab **11**

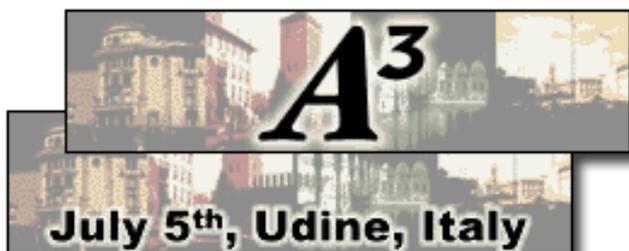


REBOL

SPECIALE

AMIGA ALPE ADRIA 2003

Un reportage sull'incontro tutto italiano per utenti **Amiga...** e non solo **22**



EMUZONE

Scopriamo l'**MSX** e giochiamo a **Prince of Persia**, un mito **13**

PROGRAMMAZIONE

Prima parte del tutorial sul linguaggio di programmazione **REBOL** **16**

OZONE

Il client **BeShare** ora anche per **Windows** **19**

GIOCHI

I giochi più divertenti con cui trascorrere questa torrida estate **20**

BEYOND

BeGeistert



Il sito BeGeistert conferma le date di svolgimento del più importante raduno europeo degli utenti BeOS.

Dunque dal 18 al 19 ottobre 2003 si svolgerà l'undicesima edizione di BeGeistert, come sempre a Düsseldorf, ma nella nuova sede, l'ostello della gioventù, che dovrebbe riuscire ad ospitare tutti gli aspiranti partecipanti.

Il condizionale è d'obbligo, perché i partecipanti alla manifestazione aumentano di anno in anno, e arrivano numerosi da ogni parte d'Europa, Italia compresa!

<http://www.begeistert.org/>

La telenovela continua

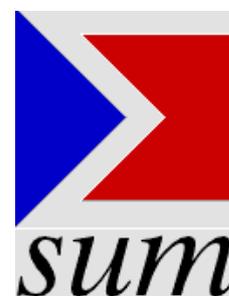
SCO da qualche tempo è al centro di una profonda contestazione, infatti sostiene di aver subito gravi danni a causa della violazione di alcuni copyright.

Secondo SCO parte del codice del kernel linux sarebbe stato copiato da Unix, i cui copyright sono in suo possesso.

Ovviamente SuSE e Red Hat respingono al mittente ogni accusa ed affermano che si tratta di minacce prive di ogni fondamento. Secondo i due Brand, la "richiesta riparatrice" di SCO (che chiede agli utenti linux di acquistare una licenza di UnixWare per regolarizzare la propria posizione) sembra altrettanto assurda.



Sum Software rilascia Becasso 2.3



Era dal lontano 1999 che la Sum Software non rilasciava aggiornamenti per questo programma di grafica. Becasso è stato il primo programma per la grafica commerciale uscito per BeOS. In questa nuova versione risulta di gran lunga migliorato il supporto per gli script attraverso cui è possibile controllare i filtri e i loro parametri.

<http://www.sumware.demon.nl/>

Palm Tungsten T2



Il T2 migliora diversi aspetti del T: in primis il design è decisamente più ricercato, con una scocca completamente argentata, nonostante mantenga l'apertura del flip a scorrimento. Il display è la prima vera novità. Questo, infatti, è il medesimo già visto sul Palm Zire 71, ovvero a 65 mila colori TFT transflective touch screen con una risoluzione da 320 x 320 pixels. Il processore è il Texas Instruments OMAP 1510 ARM a 144 Mhz, corredato da 32 MB di RAM e 8 MB di ROM. Il sistema operativo è Palm OS 5.2.1.

Tra le novità da citare la possibilità di riprodurre file MP3 e filmati di vario genere anche in streaming video grazie al lettore RealOne Player integrato. Il Tungsten T2 avrà anche il software Documents To Go che gli permetterà di leggere ed editare file Microsoft Office.

<http://www.palm.com/>

EarthSoft rilascia scheda di acquisizione compatibile con BeOS



Entro la fine del mese di luglio, EarthSoft renderà disponibile in Giappone una scheda di acquisizione video compatibile con BeOS.

Il prodotto in oggetto, noto con il nome di PV1, verrà fornito con il SDK per BeOS in modo che gli sviluppatori possano creare le proprie applicazioni.

La scheda si preannuncia interessante anche perché sarà possibile acquisire in formato DV da sorgenti analogiche, e la codifica, come pure la decodifica, saranno supportate dall'hardware, non gravando dunque sulla CPU. Purtroppo sembra che tale prodotto sarà disponibile solo per il mercato giapponese.

<http://earthsoft.jp/>

Novell

Imminente il rilascio di Novell NetWare 6.5

Novell ha ufficialmente annunciato il rilascio di NetWare per la metà di Agosto. La novità fondamentale della prossima release di Netware, sarà in un ulteriore "avvicinamento" alle piattaforme Linux.

Va infatti ricordato che nella roadmap di Novell vi è un importante obiettivo chiamato "migrazione Linux", infatti le nuove versioni dei sw Novell gireranno anche su piattaforma Linux.

Netware 6.5 sarà un prodotto integrato, capace di sopperire alle funzionalità di application server, di servizi web ed ovviamente i classici servizi Netware.

<http://www.novell.com>

Apple conferma: Panther non è un OS a 64 bit

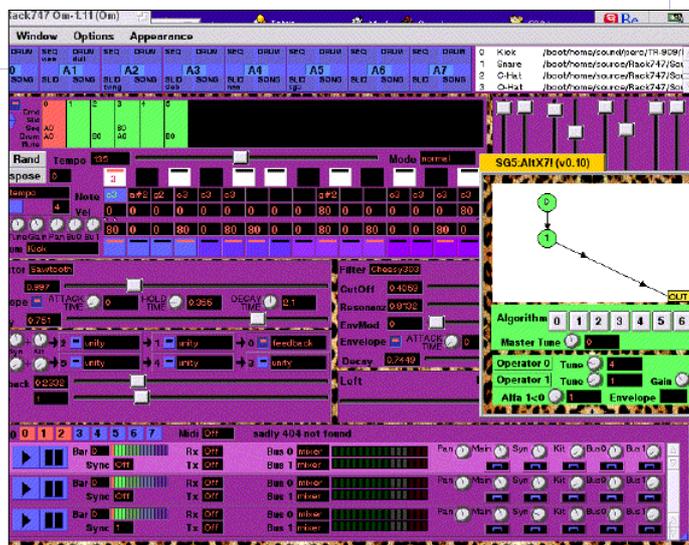


Panther non sarà un sistema operativo a 64-bit. La notizia arriva da Apple stessa che, per bocca di Greg Joswiak, responsabile dei prodotti hardware di Apple, ne ha dato conferma ad alcuni siti Internet. Ma questo

non significa che l'OS non sarà in grado di trarre beneficio dall'utilizzo dei processori PPC 970.

La scelta di Apple si traduce in un significativo vantaggio su tutti i fronti. Gli utenti non saranno obbligati ad aggiornare tutte le loro applicazioni alla versione a 64 bit per vederle funzionare sulle nuove macchine e gli sviluppatori, a loro volta, non saranno costretti ad aggiornare le applicazioni per sfruttare la velocità in MHz del PPC 970 e la nuova architettura del G5.

<http://www.apple.com/it/>



The Rack is Back

Rack747 il migliore virtual groovebox per BeOS è tornato più in forma che mai! La nuova versione inanzitutto non è più demo ma viene distribuita in versione Full, la gestione del Midi è stata sistemata ed arricchita di molti filtri ed effetti.

<http://www.labyrinth.net.au/~dak/rack>

Il filesystem di BeOS

di **Andrea Scatena** [andyscat@libero.it]

Inauguriamo la sezione per i principianti introducendo il File System di BeOS

Una delle caratteristiche fondamentali di un sistema operativo è data dalla struttura del suo filesystem.

In BeOS la struttura ricalca fundamentalmente quella dei sistemi POSIX (ricordiamo che BeOS non è una variante di Unix, ma un sistema operativo scritto completamente da zero): abbiamo quindi una directory principale da cui si dipanano le sottodirectories in cui vengono organizzate per categoria tutte le componenti del sistema e i file e le directories dell'utente.

La directory d'origine è /boot e rappresenta la "root" o "radice" del sistema, la partizione o volume da cui il sistema effettua il boot.

Da /boot si dipanano le seguenti directories:

- ▶ /boot/apps
- ▶ /boot/beos
- ▶ /boot/demos
- ▶ /boot/develop
- ▶ /boot/optional
- ▶ /boot/home
- ▶ /boot/preferences
- ▶ /boot/tmp

Ciascuno è poi libero di aggiungere qualsiasi altra directory sotto /boot, ma è comunque preferibile che le directories create dall'utente vengano inserite sotto /boot/home.

Questo perchè in BeOS esistono due classi di directories: le directories di sistema e le directories a livello dell'utente.

Tutto ciò che si trova in /boot/beos è riservato al sistema, il resto è liberamente accessibile per l'utente.

Attenzione però, perchè nulla vi impedisce di mettere le mani in /boot/beos, ma il risultato potrebbe essere un sistema inutilizzabile.

Il motivo risiede nel fatto che pur essendo un sistema compatibile POSIX non ne eredita tutte le caratteristiche, una quella della multi-utenza: BeOS non è un sistema multi-utente, anche se gran parte del codice necessario è già stato implementato.

Qualora la multi-utenza fosse implementata ne risulterebbe l'incapacità da parte del singolo utente di intervenire sulle directories di sistema.

Il discorso della multi-utenza è molto complesso e comporta la divisione degli utenti in gruppi dotati di poteri particolari, chiamati privilegi, relativi alla possibilità di lettura, scrittura, ed eseguibilità di files e directories (ma questo è un discorso che affronteremo eventualmente in altra sede).

Ecco allora che la suddivisione tra /boot/beos e /boot/home assume un significato ben preciso:

per fare un esempio, in /boot/beos/apps troveremo tutte le applicazioni di sistema, che sarebbero accessibili a tutti gli utenti; in /boot/home/apps troveremo tutte le applicazioni disponibili per il singolo utente proprietario di quella directory home e non eseguibili dagli altri utenti.

L'implementazione della multi-utenza porterebbe, inoltre, ad avere sotto /boot/home tante directories quanti sono gli utenti della macchina su cui è installato BeOS, ognuna contraddistinta da un proprio nome.

Un'altra caratteristica del sistema è data dalla esistenza di directories invisibili nel Tracker, ed accessibili soltanto da riga di comando con l'applicazione Terminal.

L'invisibilità deriva dal fatto che si tratta di directories riservate al funzionamento del sistema ma che non rivestono di alcuna utilità per l'utente finale (la maggior parte), ovvero di essere pericolose se manipolate dall'utente.

Queste directories sono accessibili ed eventualmente manipolabili soltanto tramite il Terminale e utilizzando gli opportuni comandi di shell.

Il filesystem di BeOS ha poi una particolarità unica: è costruito in modo da funzionare come un database.

Ogni elemento che costituisce BeOS, sia esso un file o una directory, viene immagazzinato come un elemento di un database, con la possibilità di effettuare ricerche anche estremamente complesse sull'intero sistema.

Questo tipo di ricerca, chiamata "query", può essere salvata e visualizzata anche succesivamente in una finestra del Tracker.

Le queries sono "live", vive, ossia variano in tempo reale qualora si modificano gli elementi che la costituiscono.

Ad esempio supponiamo di avere effettuato una query per avere l'elenco di tutte le persone appartenenti all'ItBUG (Italian BeOS User Group) che si siano iscritte da dicembre 2002 ad aprile 2003 e di visualizzarne i risultati in una finestra del Tracker: supponiamo ora di avere tralasciato di inserire un nominativo e di farlo ora, mentre la finestra rimane aperta: nel momento in cui completeremo l'inserimento, automaticamente il nuovo nominativo comparirà insieme agli altri della query, senza dovere rilanciare la query stessa.

Le possibili applicazioni di questa caratteristica sono veramente notevoli ed assolutamente esclusive di questo meraviglioso sistema operativo.

Ovviamente il discorso è molto più profondo e complesso, qui ci siamo limitati ad una panoramica a volo d'uccello, riservandoci di approfondire i singoli argomenti in futuro. ■

Il filesystem di OSX

di Andrea Scatena [andyscat@libero.it]

Proseguiamo l'analisi dei filesystem con quello di OSX

O rmai da più di due anni l'Apple ha di fatto sempre più messo da parte l'Os 9 per la nuova generazione del suo sistema operativo: Os X.

La X sta per 10 (Ten, come dice il buon Steve Jobs), ma indica anche la propria matrice di sistema Unix. Il salto da Os 9 a X non poteva essere più grande: qui non si tratta di un'aggiornamento del sistema operativo, bensì di una sua totale riscrittura, fondata su di un micro-kernel, il Mach, di derivazione BSD.

Questa decisione, nasce nel momento in cui Jobs torna alla azienda che aveva egli stesso fondato insieme all'altro Steve, Wozniak, portando con sé la esperienza di NeXTStep, un sistema operativo che è stato un flop dal punto di vista commerciale, ma che era assolutamente all'avanguardia per i tempi. Os X ha avuto finora tre incarnazioni principali: la 10, 10.1 e 10.2.

Se la primissima versione era fondamentalmente non utilizzabile dal punto di vista professionale, le successive sono andate progressivamente raffinandosi, tanto da portare oggi, con la versione 10.2.6, ad avere un sistema operativo assolutamente robusto, stabile e produttivo.

Ed alle porte c'è già la versione 10.3, nome in codice Panther (si nota una certa predilezione, da parte del buon Steve, per i felini, con Cougar, Puma, Jaguar, ed ora appunto Panther), che sposta ancora più in avanti il sistema operativo di casa Apple.

Ma prescindiamo per il momento dagli aspetti innovativi nell'uso delle tecnologie a cui siamo stati abituati, e che scandiscono, in misura ora maggiore, ora minore, ogni nuova release ed aggiornamento dei systems di Apple.

Come abbiamo fatto nel precedente articolo sul filesystem di BeOS, cerchiamo di analizzare la struttura del filesystem di X, per capire meglio questo sistema operativo, che a mio parere, affianca BeOS/Zeta come sistema operativo all'avanguardia per tecnologie e usabilità.

A differenza di BeOs, scritto interamente da zero dalle fondamenta in su, X è a tutti gli effetti un sistema Unix (tant'è che Apple è oggi riconosciuta come il più grande produttore di sistemi Unix in tutto il mondo, con una base di installato che ha ormai superato i due milioni).

Per analizzare la struttura del filesystem possiamo utilizzare un piccolo trucchetto: aprendo una finestra di terminale battiamo al prompt della shell `man hier`.

L'output del comando (TABELLA 1 della pagina

seguinte) ci fornisce una descrizione sintetica della struttura del sistema, con una succinta spiegazione delle varie directory (la directory in unix corrisponde alla cartella in mac).

La directory `/home` viene rimpiazzata dalla directory `/Users` ed in essa avremo le varie home (o inizio, come recita la traduzione italiana) degli utenti eventualmente creati sulla macchina.

Anche in questo caso la struttura è assolutamente logica e coerente, come per il BeOS e per tutti i sistemi *nix.

Os X è un sistema multi-utente e questo si riflette sulla implementazione dei permessi per gli utenti e sulla assoluta separazione delle sfere di azione di questi: ogni utente ha la propria home e nessuno (tranne il super-utente Root) può intervenire sui suoi files e directories (se si accede ad una altra home le directories che si trovano al suo interno presentano un simbolo di divieto d'accesso ad un angolo della icona della directory stessa).

Come si può vedere analizzando l'output di `man hier` abbiamo alcune directories che sembrano ripetersi lungo la gerarchia del system: le directories `/bin`, `/sbin` e `/tmp`.

E' interessante notare come sia per `/bin` che per `/sbin` la directory posta direttamente sotto `/` comprende quei programmi e quelle utilities che "appartengono" al sistema, mentre quella posta sotto `/usr` contenga le stesse tipologie di programmi, ma stavolta di pertinenza del singolo utente.

A differenza di altri sistemi *nix, Os X sotto root (`/`) colloca anche altre cartelle: `Applications`, `Desktop`, `Library`, `System`, `Users` (di cui abbiamo già parlato), `Volumes` e `Developer` (qualora si siano installati i developer tools di Apple).

Potremmo andare avanti ancora per molte pagine, ma ci fermiamo qui, sperando di essere stati sufficientemente chiari e di avervi stuzzicato l'appetito (intellettuale).

Con il prossimo articolo approfondiremo il contenuto di queste cartelle, per cui alla prossima. ■

```
[host1:~] user1% man hier
NAME
    hier - layout of filesystems
DESCRIPTION
    A sketch of the filesystem hierarchy.
    /          root directory of the filesystem
    /bin/      user utilities fundamental to both single-user and multi-user environments
    /dev/      block and character device files
               fd/ file descriptor files; see fd(4)
    /etc/      system configuration files and scripts
    /mach_kernel kernel executable (the operating system loaded into memory at boot time).
    /sbin/     system programs and administration utilities fundamental to both single-user and
               multi-user environments
    /tmp/      temporary files
    /usr/      contains the majority of user utilities and applications
               bin/      common utilities, programming tools, and applications
               include/  standard C include files
                   arpa/      C include files for Internet service protocols
                   hfs/       C include files for HFS
                   machine/   machine specific C include files
                   net/       misc network C include files
                   netinet/   C include files for Internet standard protocols; see inet(4)
                   nfs/       C include files for NFS (Network File System)
                   objc/      C include files for Objective-C
                   protocols/ C include files for Berkeley service protocols
                   sys/       system C include files (kernel data structures)
                   ufs/       C include files for UFS
               lib/        archive libraries
               libexec/    system daemons & system utilities (executed by other programs)
               local/     executables, libraries, etc. not included by the basic operating system
               sbin/      system daemons & system utilities (executed by users)
               share/     architecture-independent data files
               calendar/  a variety of pre-fab calendar files; see calendar(1)
               dict/      word lists; see look(1)
                   web2      words from Webster's 2nd International
                   words     common words
               man/       manual pages
               misc/      misc system-wide ascii text files
               mk/        templates for make; see make(1)
               skel/      example . (dot) files for new accounts
               tabset/    tab description files for a variety of terminals; used in the
                           termcap file; see termcap(5)
               zoneinfo/  timezone configuration information; see tzfile(5)
    /var/      multi-purpose log, temporary, transient, and spool files
               at/        timed command scheduling files; see at(1)
               backups/   misc. backup files
               db/        misc. automatically generated system-specific database files
               log/       misc. system log files
                   wtmp     login/logout log; see wtmp(5)
               mail/     user mailbox files
               run/       system information files describing various info about system since it
                           was booted
                   utmp     database of current users; see utmp(5)
               rwho/     rwho data files; see rwhod(8), rwho(1), and ruptime(1)
               spool/    misc. printer and mail system spooling directories
                   mqueue/  undelivered mail queue; see sendmail(8)
               tmp/      temporary files that are kept between system reboots
```

TABELLA 1

Navigare nella rete

con BeOS e l'ADSL

di Giuseppe Gargaro [giuseppe.gargaro@tiscali.it]

Il Dossier di questo mese ci guida passo passo alla configurazione e all'installazione di un collegamento ADSL per BeOS.

ADSL (**A**symmetric **D**igital **S**ubscriber **L**ine) è una tecnologia che trasforma la linea telefonica tradizionale in una linea di collegamento digitale per il trasferimento dati. L'ADSL usa il doppino telefonico ordinario e permette di trasportare un segnale a frequenza elevata per la trasmissione di dati, contemporaneamente al segnale destinato al trasporto della voce. L'ADSL è una connessione ad alta velocità in modalità asimmetrica, ciò significa che le velocità di invio e di ricezione dei dati non sono uguali; il sistema asseconda le esigenze di un tipico utente internet consentendo di sfruttare la massima banda disponibile in ricezione, a discapito di quella in trasmissione. La velocità che l'ADSL può raggiungere è variabile a seconda del contratto stipulato con il provider, variando da 256 Kbit per secondo a 1.2 Mbit/s in ricezione, e 128 o 256 Kbit/s in trasmissione.

Vediamo di cosa abbiamo bisogno per utilizzare l'ADSL in BeOS:

► driver PPPoE per BeOS, potete scaricarlo da BeBits all'indirizzo:

<http://www.bebits.com/app/1707>;

- un modem ADSL di rete e una scheda di rete Ethernet compatibile con BeOS (es. una basata sul chip RealTek);
- il contratto di fornitura ADSL con un provider.

Configurazione NetServer in BeOS 5

Siamo pronti possiamo iniziare, la procedura che segue vale per BeOS 5 con NetServer in seguito vedremo come configurare BeOS con Boneyard (Zeta, Dano, BeOS 5 + Boneyard).

Una volta scaricato il driver PPPoE aprire il file zip che contiene tre cartelle: la cartella `bin` contiene il driver mentre le cartelle `src` e `documentations` contengono rispettivamente il sorgente e la documentazione del driver disponibile sia per PPC che per x86.

1) Installiamo il driver

Aprire la cartella `bin` e spostate il file `pppoe` nella cartella `/boot/beos/system/add-ons/net-server/` poi cliccate due volte sul file `pppoe` e spuntate entrambe le caselle disponibili (PPPoE Active e Force Manual DNS).

2) Configurazione delle risorse

La vostra scheda Ethernet deve essere riconosciuta dal sistema, nel caso non sia presente tra i *Devices* cercate un driver per la vostra scheda su **www.bebits.com** e installatelo. Se la scheda di rete è correttamente installata aprendo *Network* dalle preferenze, la troverete indicata tra le *Network Interfaces*, inserite l'*Hostname* salvate e fate il restart del network. Cliccate su *Settings* selezionate *Specify settings* e inserite un indirizzo IP es. 192.168.0.1 come *Subnet mask* 255.255.255.0 e come *Gateway* 0.0.0.0 poi premete sul tasto *Done* e riavviate il network.

Aprire la cartella `/boot/home/config/settings` e cliccate due volte sul file `network` se avete seguito bene il procedimento dovrebbe apparire come in **FIGURA 1**, editate (modificate suona meglio) questo file (alla riga `PROTOCOLS="pppoe"`) e salvatelo, poi riavviate di nuovo il network a cui potete accedere sempre dal sottomenù preferenze della Deskbar.

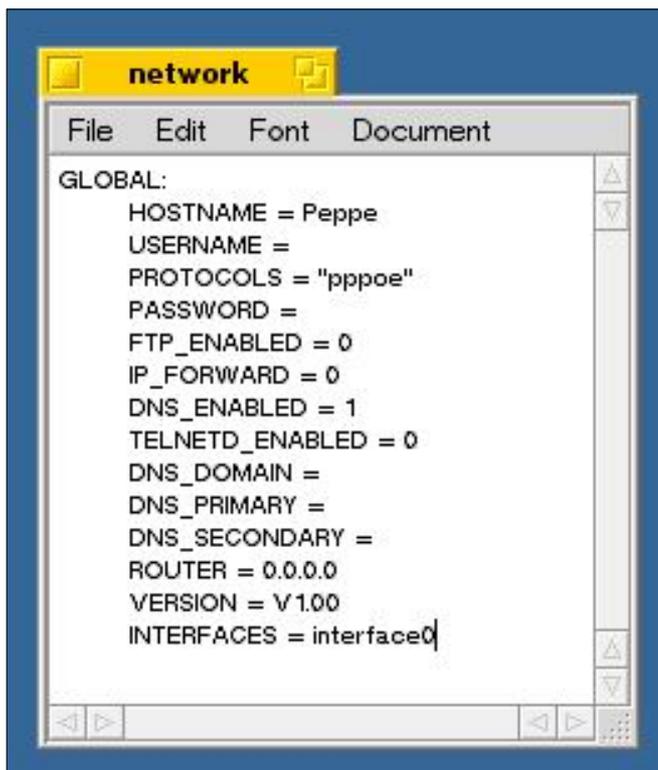


Figura 1

3) Configurazione di Dial-Up Networking

Sempre da preferenze apriamo Dial-Up Networking, clicchiamo su modem selezioniamo "Connect via" pppoe e poi clicchiamo su custom; ora aggiungiamo un nuovo modem e poi clicchiamo su "done" confermate premendo "yes" e cliccate di nuovo su "done".

Tornati alla prima schermata del Dial-Up Networking aggiungiamo una nuova connessione cliccando su "click to add", ora inseriamo l'username e la password del nostro abbonamento ADSL la password può essere salvata selezionando "save password". Clicchiamo su "Settings" e inseriamo il PrimaryDNS e il SecondaryDNS, poi riavviamo il sistema. Al riavvio per connettersi è sufficiente premere "Connect" dal Dial-Up Networking, se tutto è ok in qualche secondo saremo online.

Alcuni casi particolari:

Se utilizzate Fastweb andate in Network, impostate l'IP via DHCP, salvate le modifiche e riavviate Netserver. Dopo una breve attesa di circa un minuto, potrete navigare con Fastweb a patto di non aver già configurato un accesso Dial-up.

Il PPPoE non è supportato da Infostrada, quindi l'unico modo per connettersi a Internet è avere il modem/router in modalità "Always Connected".

Configurazione Boneyard in Zeta, Dano e BeOS 5 + BONE

Dal sottomenù preferenze della Deskbar aprite Boneyard, cliccate sulla linguetta *Interface* dovreste vedere tra i device la vostra scheda Ethernet che nel mio caso ha un chip realtek per cui ho /dev/net/rt18139/0 premete su *Settings* e dopo aver selezionato *Addressed Manually* inserite *IP Address* 192.168.0.1 , *Net Mask* 212.216.112.112 e *Gateway* 212.216.172.62, poi salvate premendo *Done*.

Cliccate sulla linguetta *Dial-UP* ed in *Connect to* selezionate *new* e dopo aver inserito un nome cliccate su *Add Service*, ora selezionate in *PPP Type* pppoe in *Ether Device* il device della scheda di rete... /dev/net/rt18139/0 (nel mio caso), inserite *User Name* e *Password* della vostra connessione e cliccate su *Apply Changes*; cliccate sul pulsante modem come mostrato in **FIGURA 2** e configurate le varie voci così: *Modem Port* /dev/ports/tri_modem , *Port Speed* 115200 , *Init String* ATZ e salvate.

Per connetervi basta che premere sul pulsante *Connect* sempre da Dial-Up.

E' tutto buon divertimento!

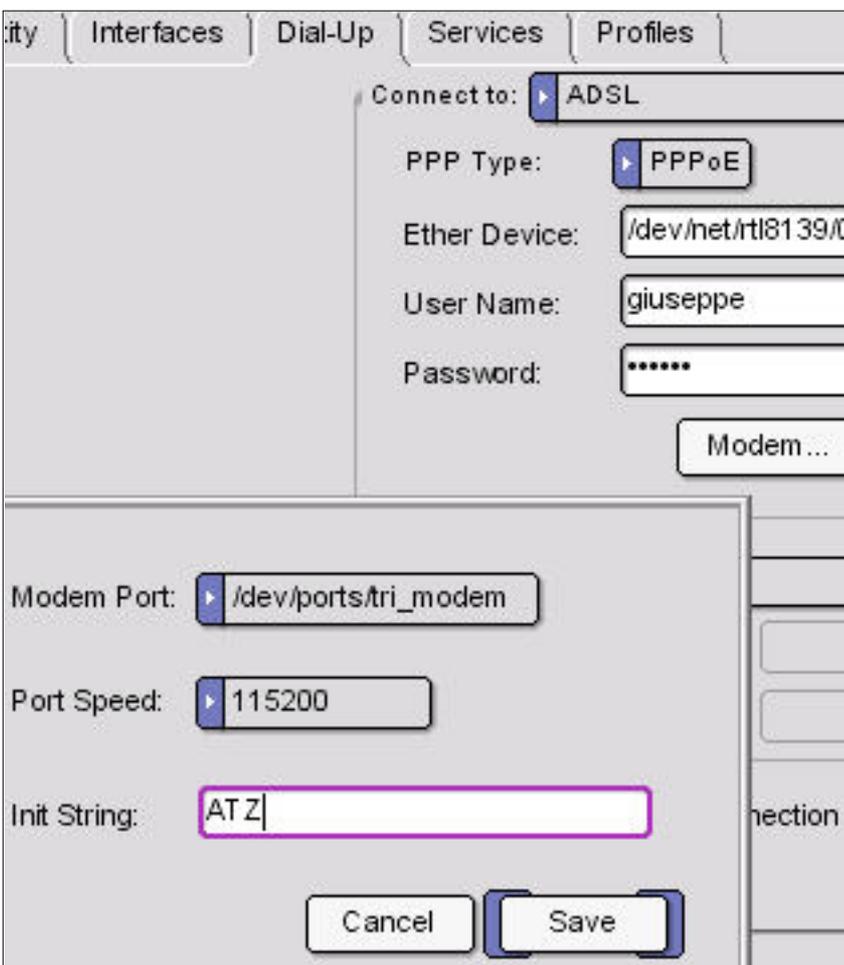


Figura 2

Boneyard ovvero il pannello di configurazione unificato di BONE (BeOS Networking Environment).

Dalla Be alla Zeta

di daf [daf73@interfree.it]

Zeta è il successore di BeOS e YellowTab raccoglie l'eredità di Be

Quando ho conosciuto BeOS, leggendo un articolo di Paolo Pisati su un Linux & C. di seconda mano, il "fattaccio" era già avvenuto. BeOS era stato abbandonato dai propri creatori, venduto a chi non voleva farne niente... GAME OVER.

Però... ormai la scatola l'avevo aperta, non potevo fare a meno di guardarci dentro. Così, senza rendermene conto, sempre più spesso la scelta del sistema operativo all'avvio del computer mi portava ad osservare sette icone viola accendersi in sequenza. In poco tempo avevo già conosciuto tutti i siti che parlavano di BeOS, conoscevo a memoria tutte le applicazioni di bebits, non iniziavo a lavorare se prima non avevo dato un'occhiata alle ultime notizie su *The Beos Journal*.

Ma il futuro non sembrava roseo. La mancanza di sviluppo del sistema operativo cominciava a farsi sentire, il senso di abbandono della comunità beossiana era palpabile nei vari forum che incontravo. C'erano alcuni progetti di creazione di un nuovo sistema operativo più o meno simile a BeOS, c'era soprattutto openbeos, in cui sono confluite le migliori menti informatiche del mondo beossiano al fine di "ricreare la R5, per poi estenderla". Il progetto sembra forte, bene impostato e molto probabilmente destinato al successo, ma

la realizzazione di qualcosa di utilizzabile non pareva essere certo prossima.

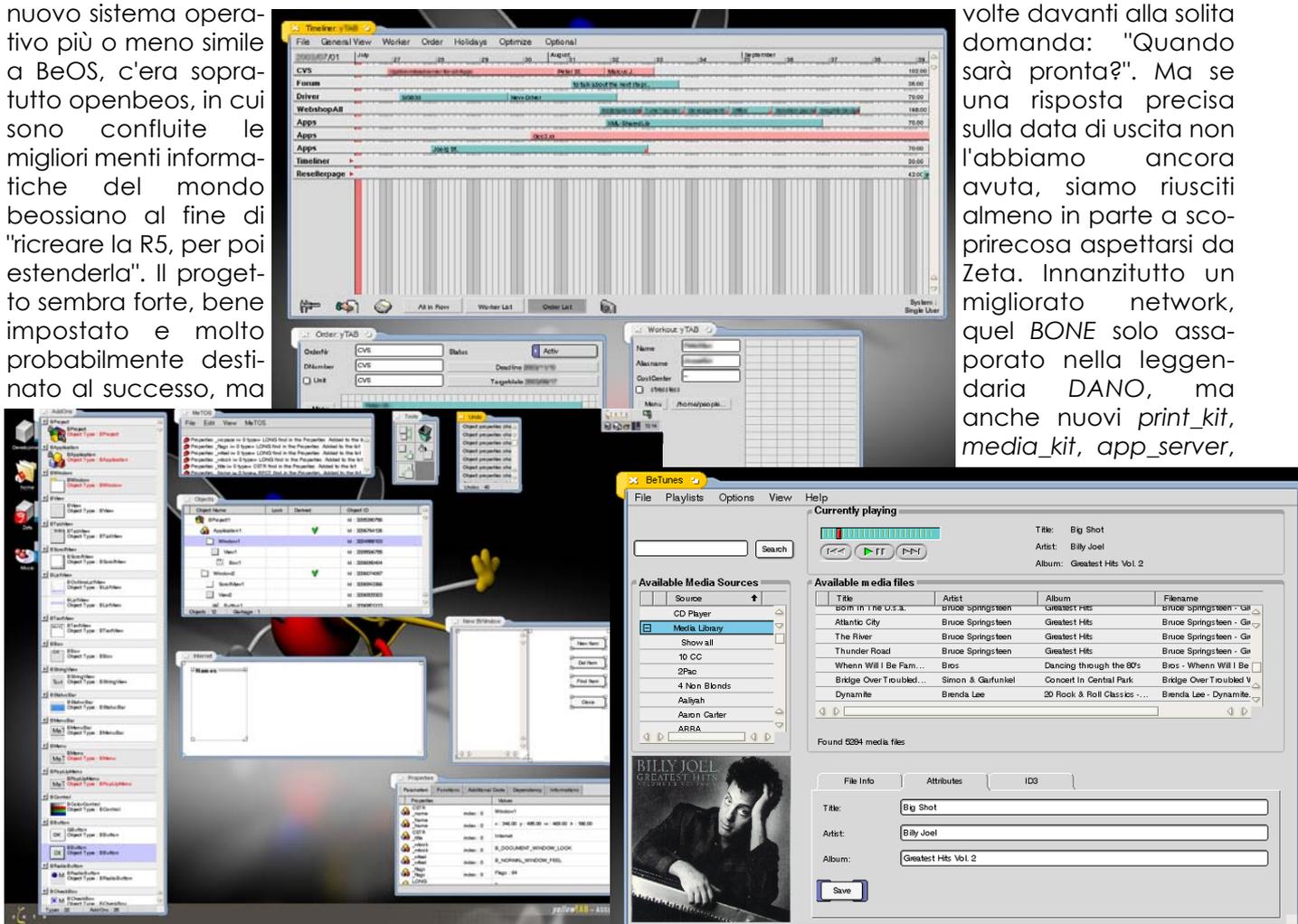
In un piccolo sito tedesco, **yellowtab.de**, un'azienda dichiarava di voler creare una nuova versione commerciale di BeOS. Io ero nuovo di quel mondo, non conoscevo persone, fatti, retroscena. Quel sito finiva praticamente con la Home Page e per mesi non l'ho più cercato.

Verso la fine del 2002 **yellowtab.de** era diventato **yellowtab.com** ed annunciava al mondo di essere prossima al rilascio sul mercato di una nuova versione di BeOS. Non qualcosa che si ispirasse all'originale, ma la vera R6, acquistata ancora grezza da Bernd Kontz direttamente dalla Be, Inc. prima che tutto il resto fosse venduto alla Palm.

Era nata Zeta.

Improvvisamente il mondo di BeOS ha ripreso vita, il forum della YT è diventato presto un punto di riferimento per tutti e Bernd si è trovato migliaia di

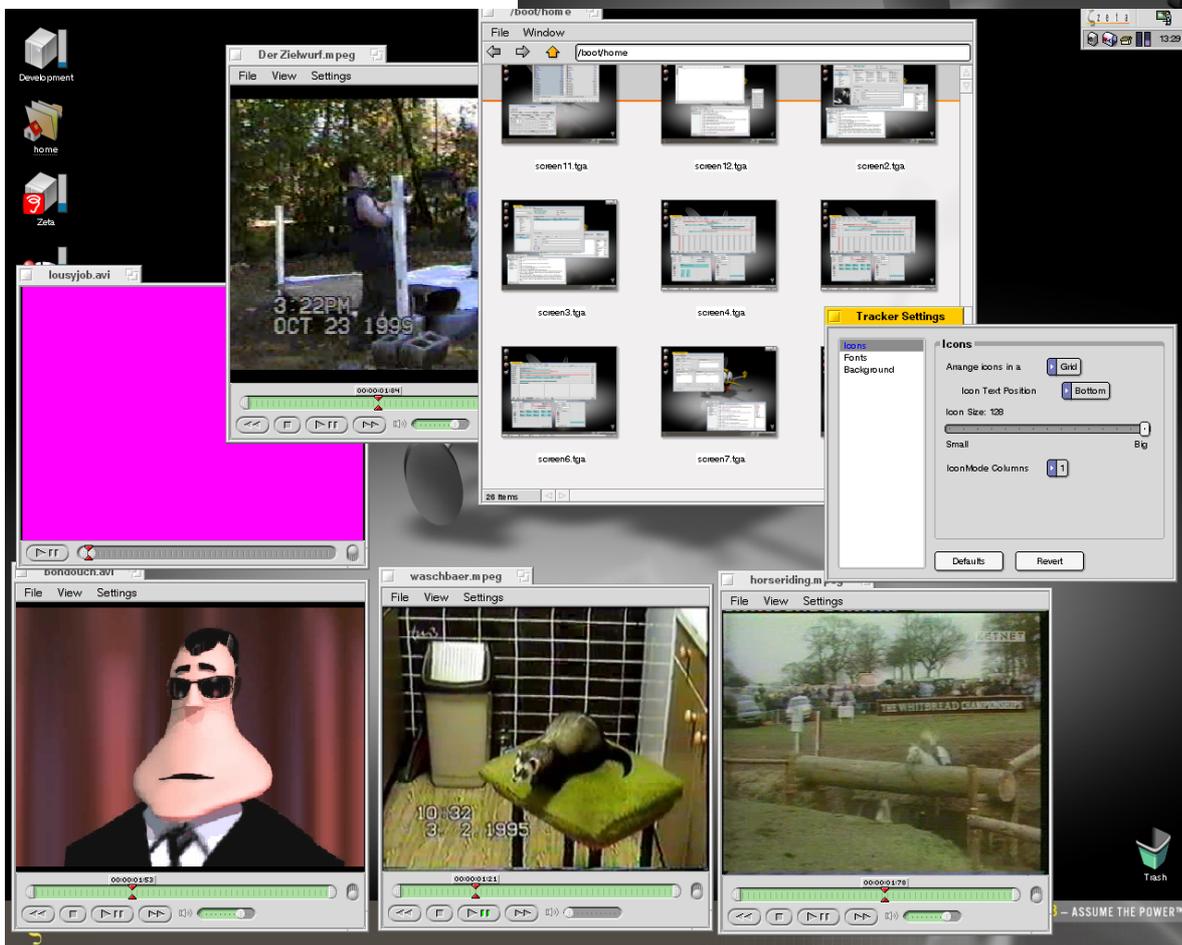
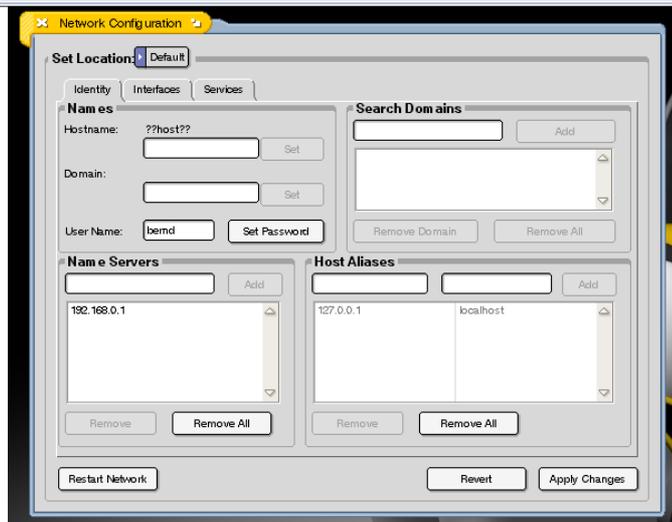
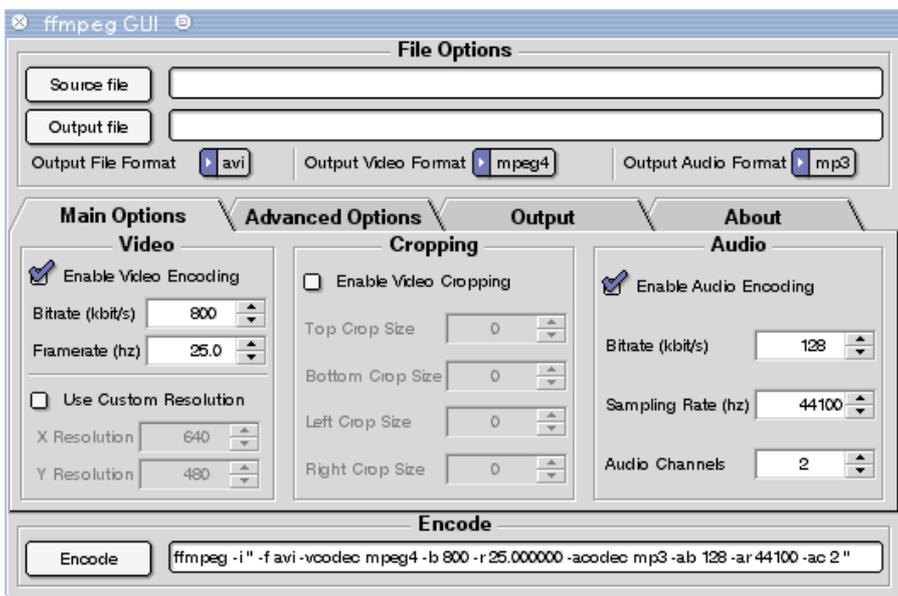
volte davanti alla solita domanda: "Quando sarà pronta?". Ma se una risposta precisa sulla data di uscita non l'abbiamo ancora avuta, siamo riusciti almeno in parte a scoprire cosa aspettarsi da Zeta. Innanzitutto un migliorato network, quel BONE solo assaporato nella leggendaria DANO, ma anche nuovi *print_kit*, *media_kit*, *app_server*,



nuovi drivers per allargare l'ormai ristretto parco hardware, migliorato supporto per le sempre più diffuse periferiche USB e una piena localizzazione del sistema operativo (fondamentale per noi che cerchiamo di diffonderne l'uso in paesi non anglofoni), oltre a nuove applicazioni sia nell'ambito dell'uso quotidiano (multimedia, ufficio, client p2p...) che in quello dello sviluppo, con moderni IDE e tool di sviluppo visuali.

Adesso, a qualche mese di distanza da quell'annuncio, l'attesa sembra volgere alla fine e chiari segnali arrivano dalla Germania ad indicare il prossimo autunno come la fine del tunnel per tutti gli amanti del multimediaOS. Nel frattempo abbiamo ingannato l'attesa guardando i video delle varie presentazioni che Bernd ha fatto a giro per il mondo e nessuno dei presenti potrà dimenticare la magica serata di Padova quando, durante il webb.it 2003, il simpatico teutonico ha dato vita ad un vero e proprio show, sottolineato dagli applausi del folto pubblico, composto in maggior parte da esperti di Linux e BSD.

Ma lo spirito di BeOS e della sua comunità è sempre stato rivolto ad innovare, cercando di costruire il futuro piuttosto che adattarsi al presente. Così non è ancora uscita Zeta che è già il momento di interrogarsi sul dopo. Il team di openbeos potrebbe non essere così lontano dal realizzare qualcosa di grande e la speranza di tutti è che le due anime dello sviluppo, open source e commerciale, sappiano al momento giusto prendersi a braccetto e camminare insieme, per garantire la continuità di quello che, ad un certo punto, temevamo di aver perso per sempre: il miglior sistema operativo possibile, qualunque sarà il suo nome.



Prince of Persia

di Giuseppe Gargaro [giuseppe.gargaro@tiscali.it]

Scopriamo insieme un videogioco che è diventato un classico per tutte le piattaforme

I giochi realizzati a cavallo tra gli anni ottanta e novanta hanno lasciato un segno in tutti noi per l'elevato livello delle idee, oggi invece predomina un'esasperata ricerca estetica e le buone idee sono decisamente pochine; tra i giochi sicuramente indimenticabili c'è "Prince of Persia" realizzato da Jordan Mechner nel 1989, all'epoca era abbastanza comune per un programmatore fare tutto da solo, così Jordan programmò il gioco e realizzò la grafica mentre il resto fu completato grazie ai suoi familiari, infatti il padre compose la musica e il fratello fece da attore per lo studio dei movimenti del principe. **Jordan Mechner** (nella foto al centro) è nato a New York, la sua passione per i computer nasce alle scuole superiori grazie ad un Apple II; ha prodotto diversi titoli per la Broderbund Software tra cui spiccano Karateka (un gioco azione/strategico che con le sue 500,000 copie vendute fu uno straordinario successo) e Prince of Persia (circa 2,000,000 di copie vendute) che vinse numerosi premi e incontrò il favore della critica che lo apprezzò soprattutto per l'animazione; il gioco uscì prima per Apple II e poi per quasi tutte le piattaforme: DOS, Macintosh, Amiga, NES, SNES, Sega Genesis, Commodore 64, ecc. Prince of Persia fu un gioco molto innovativo introducendo l'uso della "barra della vita" (health bar) in un periodo in cui la gran parte dei giochi erano basati sull'obiettivo di raggiungere il punteggio più alto, inoltre fu il primo gioco che utilizzò la "rotoscoped animation" riuscendo a riprodurre movimenti molto simili a quelli umani; per realizzare lo studio sui movimenti Jordan per prima cosa filmò il fratello con la sua videocamera mentre effettuava i



salti e ogni movimento necessario al principe poi grazie ad una nuova macchina, creata da una compagnia Inglese, riuscì ad ottenere il disegno di ogni frame e a registrarlo sul suo Apple II, una volta sul computer i frame furono modificati per assomigliare al principe e così Jordan riuscì ad ottenere una fluidità nei movimenti che era inimmaginabile per l'epoca.

Il gioco ha una storia portante, cosa non comune per i giochi dell'epoca. Il Sultano è lontano in guerra e in sua assenza il Gran Visir Jaffar tiene le redini del potere schiacciando con la tirannia il popolo, voi (il principe) siete l'unico ostacolo tra Jaffar e il trono e poiché siete un aitante giovane avventuriero, la figlia del sultano si è innamorata di voi... praticamente siete nei guai! Jaffar vi fa sbattere nel sotterraneo e, rinchiusa la principessa nella torre del palazzo, le concede un'ora di tempo per decidere se sposarlo o morire (scelta mica da poco :); ma abili come siete dopo aver superato mille tranelli e aver vinto al duello le guardie dei sotterranei, vi troverete su di un ponte sopra un abisso e di fronte a quel cattivone di Jaffar... manco a dirlo dopo un duello sanguinario lo farete a fettine... purtroppo il sultano non ha nessuna intenzione di darvi in moglie la bella figlia poiché il vostro sangue non è abbastanza blu e, se non fosse per la figlia che lo convince, vi farebbe tagliare il collo ... e vissero felici e contenti? Macché! una mattina entrando nella sala del trono vedete qualcuno al vostro posto a fianco della principessa che non vi riconosce più (gran donna, vero?) e mentre il padre ordina alle guardie di prendervi, non vi resta che lanciarsi dalla



Nella foto al centro Jordan Mechner, nella foto a sinistra lo schermo iniziale del gioco originale

finestra per scappare, così inizia "Prince of Persia 2".

Prince of Persia presenta una grafica che, nonostante la semplicità, trasmette appieno l'atmosfera avventurosa e cattura il giocatore calandolo nel ruolo di un eroe solitario alle prese con trabocchetti e meccanismi nascosti, pronto a difendersi da scimitarre e scheletri. Il gioco alterna sapientemente azione e riflessione, il risultato è un arcade adventure godibilissimo sia nelle fasi di combattimento all'arma bianca che nell'esplorazione nel dedalo di tranelli e sorprese, riuscendo anche a trasmettere il pathos che precede un salto e lo scorrere del tempo che ci è concesso per salvare la nostra amata. Nonostante la grande giocabilità Prince of Persia è abbastanza difficile sia per il poco tempo disponibile che per la difficoltà dei salti. Il gioco ha avuto tre sequel, il primo "Prince of Persia 2: The Shadow and the Flame" uscì nel 1993, il secondo "Prince of Persia 3D" uscito nel 1999 ed infine il terzo "Prince of Persia: The Sands of Time" approdato su PS2. Se non avete mai giocato a "Prince of Persia" vi siete persi molto anche se potete sempre recuperare il tempo perso :) ■



Alcuni screenshots presi da le versioni per Genesis e altri sistemi



EMULIAMOLO!

Vediamo come rivivere l'avventura grazie agli emulatori; senza dubbio una delle versioni più belle graficamente è quella per la console Sega Megadrive (Genesis), seguita dalla versione per super Nintendo, Macintosh e per PC. Di seguito alcuni emulatori che possono essere utilizzati per giocare a Prince of Persia:

Gens (emulatore di Sega Megadrive per BeOS)
www.dawnofthegames.com/downloads/gens2.12a_bin.zip

DGen/SDL (emulatore di Sega Megadrive per Linux, BeOS)
www.pknet.com/~joe/dgen-sdl.html
 richiede le librerie SDL (Simple DirectMedia Layer)

Generator (emulatore di Sega Megadrive per MacOS)
www.bannister.org/software/generator.htm

AmiGenerator (emulatore di Sega Megadrive per AmigaPPC)
www.amidog.com/emu/amigenerator/

DOSBox (emulatore di "PC x86" per MorphOS, Amiga68K, Linux e BeOS)
dosbox.sourceforge.net/download.php?main=1
 richiede le librerie SDL (Simple DirectMedia Layer)

EMUNEWS

Genesis/Megadrive

Stéphane Akhoun e Caz hanno portato Gens, un ottimo emulatore di Genesis, su Linux!
<http://gens.consolemul.com/>
 nuova versione DGen/SDL 1.23 per BeOS
http://www.dawnofthegames.com/downloads/dgen-1.23_x86.pkg.zip

NeoGeo

edge ha realizzato il porting di GnGeo un emulatore di NeoGeo molto avanzato
http://www.dawnofthegames.com/downloads/gngeo-0.5.9a_x86.pkg.zip

Gameboy

E' stato aggiornato KiGB un ottimo emulatore compatibile con OS 9 e OS X.
<http://www.bannister.org/software/kigb.htm>

Apple[]

Nuovo emulatore (Virtual []) di Apple II per Mac.
<http://www.xs4all.nl/~gp/Virtuall/>

MSX. Genio ad 8-bit

di Giuseppe Gargaro [giuseppe.gargaro@tiscali.it]

L'MSX è stato uno dei computer ad 8-bit più potenti mai progettati. Peccato che non abbia avuto fortuna. Scopriamo insieme come farlo rivivere sui nostri sistemi

La piattaforma MSX (**Microsoft Extended Basic**) è nata nel 1983 da un consorzio di aziende giapponesi specializzate in microelettronica che intendevano creare un nuovo standard; seguendo le specifiche di questo standard furono realizzati una decina di computer (prodotti inizialmente da **Sharp, Nec, Sony, Hitachi, Toshiba, Mitsubishi, Canon, JVC**, ecc.) con le stesse risorse hardware e soprattutto compatibili tra di loro. Il punto debole di questa idea che avrebbe potuto rivoluzionare il mondo dell'informatica fu il software, infatti le aziende giapponesi distribuirono ovunque l'hardware sperando nello sviluppo locale di valide applicazioni, purtroppo le cose andarono male sia sul mercato americano che su quello Europeo ove molte applicazioni erano spesso ottenute da cattive conversioni di software per altre macchine ad 8bit come lo Spectrum. Gli MSX presentavano un processore Z80A, da 8 a 64 Kb di RAM ed erano tutti compatibili con l'MSX Basic. Questo computer a 8 bit ha avuto diffusione e successo soprattutto in Giappone grazie anche agli ottimi giochi creati dalla mitica **Konami**; in Europa la macchina ebbe una discreta diffusione solo in Germania, Olanda e Spagna. Uno degli msx che ha avuto una certa diffusione in Italia e in Europa è il **VG8020** della **Philips** (Z80 a 3,6 MHz, RAM 80K, 256x192 pixel con 16 colori). L'MSX ha avuto come successore l'MSX2 una macchina molto valida ancora sostenuta da diversi programmatori

www.msx.org
www.nic.funet.fi/pub/msx/
www.xydirect.com/msx/



Il Philips VG8020

DA NON PERDERE

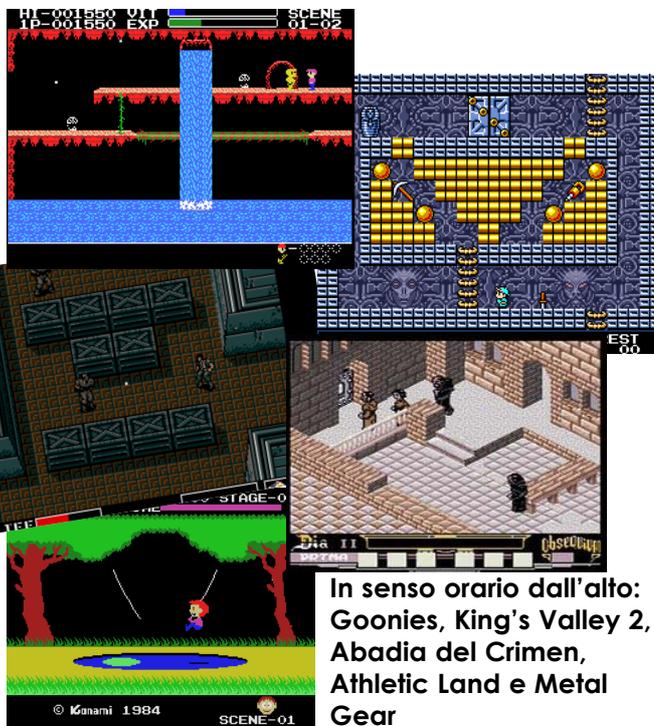
Athletic Land (1984 Konami) un gioco semplice e divertente, il protagonista è un bambino che fa di tutto per divertirsi

Goonies (1986 Konami) dovete liberare tutti i bambini imprigionati nelle caverne, bello.

Abadia del Crimen (Color Remake for MSX2 2000 Opera Soft) un grande classico in questa versione a 16 colori.

King's Valley 2 (1988 Konami) il mio preferito :)

Metal Gear (1987 Konami) uno dei migliori giochi per MSX.



In senso orario dall'alto: **Goonies, King's Valley 2, Abadia del Crimen, Athletic Land e Metal Gear**

EMULATORI

Amiga
fMSX students.silab.dsi.unimi.it/~cs390447/afmsx.html
AmiMSX www.arrakis.es/~joanant/amsx.html

Mac OS
fmsx, freeM e Zodiac emulation.net/msx/

Linux
fMS fms.komkon.org/fMSX/
OpenMSX openmsx.sourceforge.net/

FreeBSD
fMSX
www.FreeBSD.org/cgi/cvsweb.cgi/ports/emulators/fmsx

BeOS
Per quanto riguarda l'emulazione dell'msx sotto BeOS vi consiglio di usare fMSX/SDL, questo emulatore richiede le librerie SDL ed è dotato di una interfaccia per lanciare i giochi fMSXLauncher; per meglio controllare l'emulatore vi consiglio di creare dei comodi script per lanciare i singoli giochi, di seguito un esempio di script per lanciare "Abadia del Crimen" (Color Remake for MSX2), i comandi che potete usare vengono spiegati nel file readme.txt che accompagna l'emulatore, per utilizzare una rom togliete il comando "-disk a" dallo script.

```
#!/bin/sh
cd /boot/emulazione/fmsx-sdl/
fmsx -filter 3 -msx2+ -diska /boot/emulazione/msx_giochi/abadia.dsk
```

fMSX
beemulated3.bei.t-online.de/fmsx-sdl-2.6.0.40beos.zip

SDL
diana2.bei.t-online.de/SDLGameLibs.zip

REBOL - parte I

di **Davide Gessi** [davidegessi@tin.it]

Scripting language o framework per applicazioni distribuite?

Il mio primo approccio alla programmazione risale a più di 20 anni fa, quando quasi per caso mi capitò in mano una rivista in cui era trattata la costruzione di una scheda a microprocessore venduta in scatola di montaggio, il linguaggio in questione era l'assembler del processore 6502 (quello che sarebbe poi stato il cuore del VIC 20). Quando capii il primo esempio trattato (non era grande cosa in realtà, era la banale somma di due byte..), mi si aprì un mondo fantastico davanti, gli opcode esadecimali avevano preso il posto dei mattoncini Lego.

In seguito mi sono sempre occupato di informatica, prima per hobby, poi per studio e in seguito per lavoro; da allora di acqua sotto i ponti ne è passata, ed ho avuto l'occasione di utilizzare molte piattaforme di sviluppo; tre sono i linguaggi di programmazione che mi hanno fatto rivivere l'ebbrezza della prima addizione: il primo fu il Prolog, il secondo fu Smalltalk ed il terzo manco a dirlo è stato, in tempi più recenti, il Rebol.

Rebol è un linguaggio interpretato, ideato, sviluppato da Carl Sassenrath, uno degli sviluppatori di AMIGA ed ora fondatore di *Rebol Technologies* (<http://www.rebol.com>).

Si tratta di un linguaggio multipiattaforma, gira su una quarantina di sistemi operativi diversi, partendo da Linux per arrivare ad Amiga, passando per Solaris, MacOS, Windows e last but not least BeOS; pur essendo molto leggero come dimensioni, non richiede librerie aggiuntive per il suo utilizzo. Brilla per semplicità ed eleganza formale ed è possibile sviluppare applicazioni di una certa complessità scrivendo veramente poco codice. Una delle sue caratteristiche salienti è la gestione nativa dei vari protocolli che sono usati in internet (quelle sigle strane del tipo http, ftp, dns) e questo permette di sviluppare applicazioni distribuite, sistemi client-server, messenger per il peer2peer in maniera naturale senza ricorrere a protocolli oscuri e pesanti.

Questa vuole essere la prima di una serie di articoli dedicati alla programmazione in REBOL, ci si accosterà a questo linguaggio mettendone in evidenza i punti di forza, ma con un occhio critico alle sue limitazioni.

INSTALLAZIONE

La procedura di installazione è semplice e non dà problemi di sorta. Una volta scaricato il software dal sito della *Rebol* (www.rebol.com/downloads/view-pro052.tar.gz per Intel e www.rebol.com/downloads/view-pro051.tar.gz per PowerPC), occorre decomprimerlo in una directory temporanea e lanciare l'eseguibile `rebol`.

L'installazione parte chiedendo la cartella in cui verrà effettuata l'installazione, (`/boot/apps/rebol` potrebbe essere una buona scelta) e prosegue mostrando una maschera in cui occorre inserire alcune impostazioni fondamentali, quali nome dell'utente, server pop3 e smtp ed eventualmente il proxy che utilizziamo per accedere ad internet. Una volta salvate le impostazioni siamo pronti per partire ! Anzi

non ancora... vi consiglio di creare subito un link all'eseguibile `rebol` che posizioneremo in `/bin` così da averlo sempre a portata di mano: per i gli smalzati della bash è sufficiente un'istruzione del tipo:

```
ln -s /boot/apps/rebol/rebol /bin/rebol
```

seguita da

```
chmod 744 /bin/rebol
```

mentre per gli altri basta trascinare con il tasto destro l'eseguibile `rebol` in `/bin` e creare il link, per poi impostare il permesso di esecuzione dalla maschera richiamando il `get-info`.

LA CONSOLE

La versione di Rebol che abbiamo scaricato è denominata "View" che, a differenza della versione "Core", è dotata di una libreria grafica che permette di costruire interfacce, pulsanti, caselle di testo e tanti altri widget; visto e considerato che il nostro fine ultimo è quello di riempire BeBits di una serie di nuove utility ed applicazioni questa è proprio la versione che fa per noi.

Aprondo il terminale e digitando "rebol" la versione View parte mostrando il Desktop, caratterizzato da una serie di icone alla sinistra e centralmente le informazioni sulla versione dell'interprete (vedi **FIGURA 1**). Per questi primi approcci con il nuovo giocattolo abbandoniamo il desktop, anche se un pò a malincuore, cliccando l'icona console sulla sinistra ritornando così al terminale che ospita la console `rebol` che ci attende fiduciosa con il prompt ">>".

Se scriviamo qualcosa del tipo:

```
view layout [text "hello world!" button
              "Close" [unview]]
```

l'interprete appena installato emetterà il suo primo vagito.

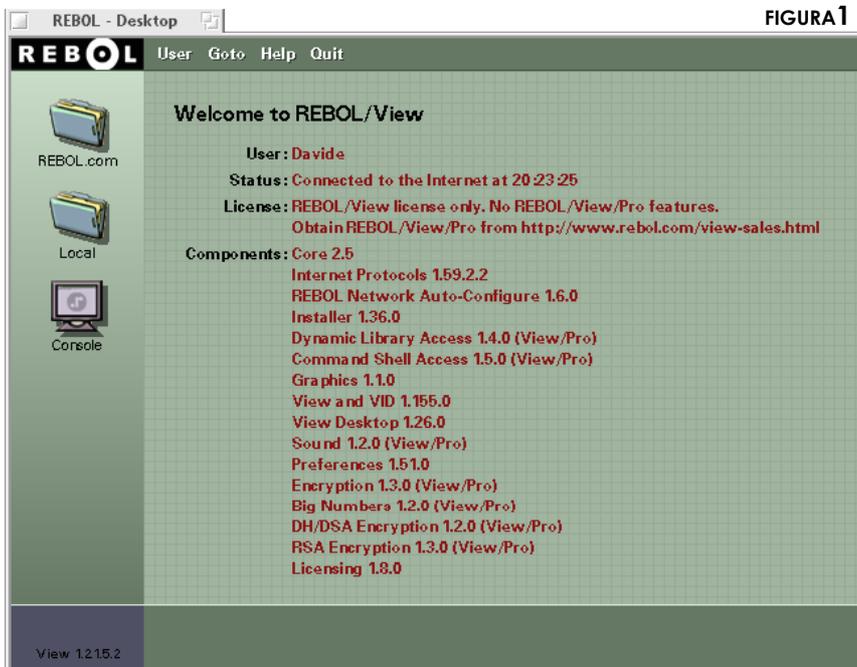


FIGURA 1

UN'OCCHIATA AL LINGUAGGIO

Lungi da me voler replicare i tutorial che si trovano sul sito di rebol ai quali vi rimando per approfondire i vari punti che tratteremo, piuttosto vorrei sottolineare le peculiarità del linguaggio per solleticare la curiosità dei miei attenti lettori. Ogni esempio riportato può essere copiato direttamente nella console per una prova immediata.

Rebol è essenzialmente privo di sintassi, non esistono terminatori di linea, le parole sono separate da spazi e i blocchi sono delimitati da parentesi quadre, non vi è una vera distinzione tra blocchi di codice e blocchi di dati.

L'assegnamento di valori a una variabile avviene usando i due punti ":" come negli esempi seguenti:

```
a: 123
nome: "davide"
email: davidegessi@tin.it
sito: http://www.bebits.com
frutta: ["mele" "pere" "banane" "arance"]
ora-attuale: now/time
```

Pur non essendoci una dichiarazione esplicita del tipo delle variabili, questo linguaggio è fortemente tipizzato, esistono molti tipi di dati oltre ai canonici, intero, decimale stringa troviamo il tipo "email", il tipo "url", il tipo "pair" e così via...

Vorrei soffermarmi sull'ultimo assegnamento in cui viene richiamata la funzione "now", la scrittura che viene dopo "/time" è chiamato "refinement" della funzione, e in questo caso permette di specificare che vogliamo solo l'ora attuale e non anche la data.

I *refinement* sono un concetto molto utilizzato in rebol facili da capire e da ricordare... permettono di specificare parametri aggiuntivi alle funzioni o di modificarne il comportamento normale. Quasi tutte le funzioni in rebol possiedono refinement, come ad esempio l'istruzione "request" che ha il compito di mostrare una dialog-box all'utente con tre pulsanti di

default "yes", "no", "cancel", in questo esempio la scelta dell'utente viene memorizzata nella variabile "salvare":

```
salvare: request "Vuoi salvare
il file ?"
```

è possibile specificare il refinement "/confirm" per ottenere una maschera con soli due pulsanti:

```
salvare: request/confirm "Vuoi
salvare il file ?"
```

nel caso in cui un refinement abbia bisogno di un parametro aggiuntivo questo deve essere aggiunto in coda all'istruzione. Per ogni istruzione è presente un aiuto in linea tramite il comando "help" o "?" in cui vi è una breve descrizione del comando e dei parametri e degli eventuali refinement:

```
help now
help request
```

FILE LOCALI E FILE REMOTI

Read e write sono le istruzioni fondamentali per interagire con i file. Diversamente da altri linguaggi come C++ o Visual Basic in cui è necessario aprire uno stream di lettura/scrittura, trasferire i dati e chiudere lo stream, in Rebol le cose sono semplificate, per leggere un file locale si opera nel modo seguente:

```
txt: read %file.txt
imm: read/binary %image.jpg
```

La prima linea legge l'intero file file.txt e lo pone nella variabile txt, il secondo legge un'immagine per cui è necessario specificare il refinement binary. Il carattere percentuale in Rebol identifica il tipo "file".

L'istruzione write prende due parametri, il primo la destinazione dei dati e il secondo i dati da scrivere:

```
write/append %log.txt now/precise
;aggiungo l'ora corrente al file di log
write %copia.txt read %file.txt
;copia del file
```

Il carattere ; viene usato per inserire commenti nel codice, ogni carattere tra il punto e virgola e la fine della linea viene ignorato.

Vorrei far notare che il secondo esempio non solo effettua una copia di file di testo, ma converte anche i terminatori di linea per la piattaforma corrente, quante volte vi è capitato, aprendo un file che è stato creato con il blocco note di Windows usando StyleEdit, di vedere dei quadratini in corrispondenza della fine di ogni linea? Beh in questa maniera si risolve il problema semplicemente...

La lettura/scrittura di file remoti risulta ugualmente semplice, solo che al posto di un parametro di tipo file occorre specificare un parametro di tipo "url":

```
home-page: read http://www.bebits.com
; lettura della home-page di bebits
messaggi: read nntp://news.tin.it/it.comp.lang
; recupero messaggi dal NG
```

Una volta recuperata la home page di Bebits è possibile ed esempio spedirla per mail tramite l'istruzione "send":

```
send marco@test.it home-page
```

Oppure è possibile stampare tutti i titoli dei post del news group come spiegato in **TABELLA 1**.

CONCLUSIONI

Direi che per questo primo assaggio di Rebol è tutto, vi lascio con il vostro nuovo strumento di sviluppo consigliandovi di provare la demo del linguaggio in **TABELLA 2**.

Per ogni commento, idea e critica, vi invito ad utilizzare il piccolo script di **TABELLA 3** che genera un modulo per il feedback. ■

```
forall messaggi [
  ;per ogni messaggio recuperato
  ;effettuo il parsing di ogni messaggio recuperando l'oggetto del messaggio
  parse messaggi/1 [thru "Subject: " copy e to newline]
  ; stampo l'indice del messaggio ed il suo oggetto
  print reduce [index? messaggi tab e]
]
```

TABELLA 1

```
view layout [
  backdrop black
  across h1 red "Alcuni demo per "
  image http://www.itbug.org/images/topics/beos.gif
  return below
  button "Gel" [do http://www.rebol.com/view/demos/gel.r]
  button "Mines" [do http://www.rebol.com/view/demos/mines.r]
  button "Rebtris" [do http://www.rebol.com/view/demos/rebtris.r]
]
```

TABELLA 2

```
view center-face layout [
  style tx label 100x24 right
  across
  tx "Categoria:"
  cat: choice 196x24 "Suggerimenti/Idee" "Bug report" "Commenti"
  return
  tx "Descrizione:"
  desc: area wrap 400x72
  return
  pad 106
  button "Invia" [
    out: rejoin [
      "Beyond feedback: " cat/text newline newline desc/text
    ]
    alert either not error? try [send davidegessi@tin.it out] [
      unview
      "Email spedita, grazie !"
    ] [
      "Errore durante l'invio, sei connesso ?"
    ]
  ]
  button "Annulla" [unview]
]
```

TABELLA 3

OZONE il client

BeShare per Windows

di Marco Begliardo [marco@marcobegliardo.com]

BeShare è il famosissimo network di file-sharing per BeOS. Oggi disponibile anche per Windows

Non volete rinunciare a condividere i vostri file e chattare su BeShare anche quando utilizzate Windows? La soluzione c'è: Ozone.

Ozone (<http://www.ozone-o3.net>) è un client BeShare (il sistema di condivisione file e chat per gli utenti BeOS) per Windows, realizzato da Vitaliy Mikitchenko e rilasciato con licenza freeware.

L'installazione dura pochi secondi, dopo i quali compare un'icona sul desktop rappresentante una molecola (di Ozono?).

Appena lanciato il programma, compare la finestra principale, con un look molto simile al BeShare per BeOS: in alto l'elenco dei server, lo spazio per la scelta del nickname e dello status; subito sotto la finestra principale che visualizza i messaggi del server e degli altri utenti connessi; a sinistra l'elenco degli utenti, e in basso lo spazio per digitare i propri interventi.

Per lanciare la connessione è sufficiente scegliere nella lista un server (per esempio beshare.tycomsystems.com), impostare il proprio nick e scegliere dal menu *File* la voce *Connect*.

L'interfaccia, come per il cugino BeShare, è del tutto intuitiva e molto semplice da comprendere; digitando nell'apposito spazio in basso e premendo *INVIO* è possibile chattare con gli utenti "in

pubblico", cliccando col tasto destro su un nickname nella lista si può invece entrare in chat privata, nonché ottenere la lista dei files condivisi dall'utente: dalla finestra di ricerca si possono scegliere quelli da scaricare, che verranno aggiunti nella finestrella "File transfers".

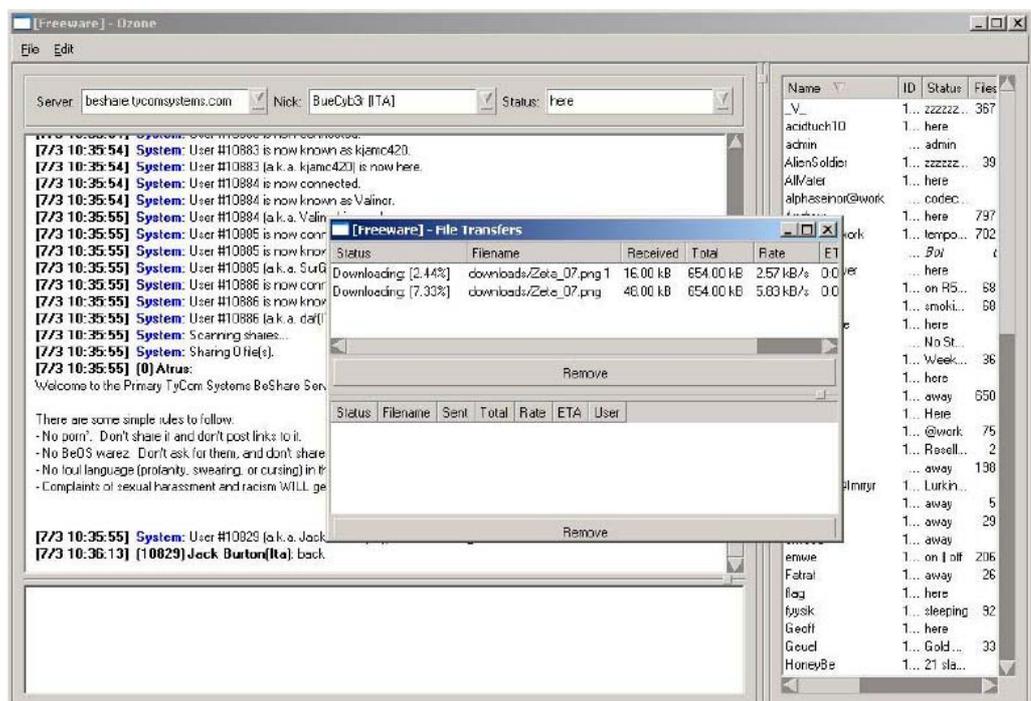
Se si desidera cercare un file su tutti gli utenti, basta scegliere il comando *File > Search* e digitare ciò che vi interessa; per condividere i propri file è sufficiente copiarli nella cartella che si aprirà cliccando su *Open shared folder* del menu *File*.

Da notare che su tutte le finestre del programma campeggia la scritta [Freeware], a rimarcare le intenzioni del programmatore.

Un ultimo accenno alle personalizzazioni: dal menu *Edit->Preferences* è possibile andare a configurare il programma in parecchi aspetti, dall'autologin ai colori dei messaggi; da segnalare il pannello "Style" che permette di cambiare il tema delle finestre del programma, secondo diversi stili: Windows, SGI, Motif...

Da alcuni giorni il sito ufficiale del programma annuncia che Ozone 2 è in fase di sviluppo (quasi giornalmente vengono aggiunti nuovi screenshots), oltre a sollecitare i visitatori a... contribuire alla dotazione hardware di **Vitaliy Mikitchenko!** ■

Il client Ozone in azione mentre sta scaricando alcuni file



Giochi d'estate

di daf [daf73@interfree.it]

Grandi classici, puzzle game, platform e 3D. Con l'arrivo delle vacanze non abbiamo scuse, è ora di divertirsi!



Il mondo dei videogiochi è uno di quelli in cui maggiormente si avverte il bisogno di grandi case software che possono investire in sviluppo e ricerca. Infatti, se scrivere un'applicazione per catalogare i propri mp3 o connettersi alla rete di ICQ può essere un piacevole esercizio per far trascorrere il tempo libero di uno dei tanti buoni programmatori che utilizzano BeOS, creare da zero un gioco, un vero gioco, uno di quelli che ti fanno passare le notti insonni alla ricerca della verità, è un compito che può essere affrontato solo da un team di diverse persone dedicate in maniera completa.

E non è un caso se i migliori giochi per BeOS che a tutt'oggi si ricordano sono quei titoli che la **Wilcard Design** è riuscita a trasferire sulla nostra piattaforma. La cosa bella dei lavori provenienti dalla casa di John Fehrè il fatto che non erano semplici versioni che "giravano" anche su BeOS, ma veri e propri adattamenti che sfruttavano le caratteristiche uniche del sistema per migliorare il gioco in questione, dalla scelta del workspace in cui lanciare il gioco al readme che usavano gli attributi per rendere più leggibile il file di semplice testo, mentre un'immagine di filesystem BE veniva creata al volo e montata all'apertura per contenere tutti i dati necessari allo svolgimento della partita! Il gioco più famoso che la WD ci ha regalato è senz'altro *Civilization: Call to ThePower* mentre quello che probabilmente ne sarebbe stato il capolavoro, *Worms Armageddon*, non è riuscito ad arrivare alla pubblicazione prima che la coraggiosa azienda dovesse arrendersi alla dura legge del mercato.

E una simile situazione si è verificata nel mondo Linux con la **Lokigames** che ha portato diversi titoli famosi sul sistema operativo del pinguino prima di doversi ritirare.

Questo significa forse che l'unico sistema operativo che avrà giochi nativi sarà sempre e soltanto Windows? La risposta è no, e questa risposta viene dalle stesse case produttrici di videogiochi che hanno iniziato a rilasciare versioni Linux dei propri titoli. E non sono passatempo con le carte... parliamo di giochi come **Never Winter Nights** della **Atari** e **Enemy Territory** della **Activision**. 3D

all'ennesimo pixel!

Per capire come questa sia la strada auspicabile faccio un semplice esempio: la 3D Realms ha rilasciato ad aprile l'ormai vetusto codice di **Duke Nuken 3D**. Quattro (4) giorni dopo era già stata compilata la versione per Linux e la settimana dopo potevamo scaricare la nostra da BeBits... Quanto tempo ha perso la 3D Realms che poteva produrlo a costo praticamente zero ed offrircelo quando ancora era un gioco attuale? Ma per convincere le grandi case a tornare a guardare con interesse al nostro

Kobo Deluxe v0.4pre8



Rollemap (x86)



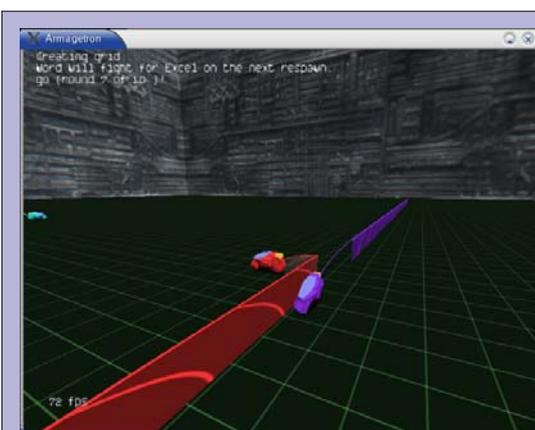
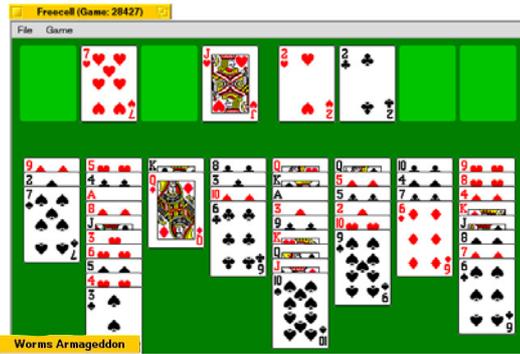
sistema abbiamo bisogno di due cose: un'azienda/associazione che garantisca lo sviluppo costante e l'accelerazione 3D hardware. Non credo che ci sia bisogno di aggiungere che questo dovrà essere compito di Zeta/OBOS... e mi sento di dire che siamo in buone mani.

Ok. Questo il passato ed il futuro. Ma per adesso? Con cosa ci distraiamo tra un film, una sessione di programmazione e un'oretta di Besharing? Un

buon aiuto ci viene dato, come sempre, dal mondo dell'opensource. La presenza sulla nostra piattaforma di Perl, Python, librerie SDL (oltre al nativo C++) ci ha permesso di compilare su BeOS molti giochi creati da gruppi più o meno nutriti di appassionati del codice libero.

Se il divertente flipper *Rollemup* e *LBreakout*, ennesima (forse la migliore) reincarnazione della pallina spacca-muri, sono ormai dei classici del passatempo Beossiano, nuovi titoli fanno spesso capolino su bebits. E infatti è da poco uscito *KrystalDrop*, ispirato al gioco da bar *Magical Drop*, dotato di una grafica veramente accattivante ed una buona colonna sonora.

Io personalmente ho passato diverse ore all'inseguimento del record di mio fratello nel velocissimo *Kobo DeLuxe*. Purtroppo ogni volta che riesco ad aver ragione di una nuova orda di asteroidi-bolle esplosive-astronavi nemiche scopro che lui si è già portato tre quadri più la... che dieci anni di differenza comincino a farsi sentire?



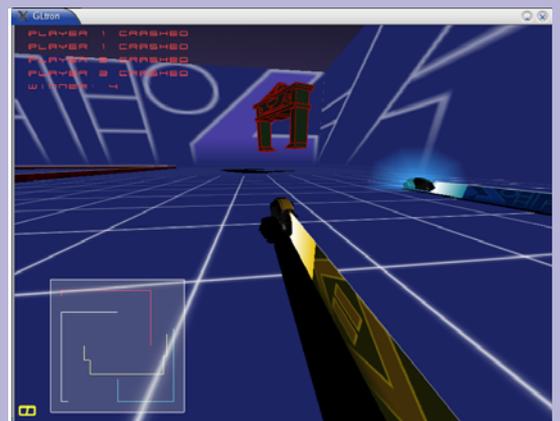
GIOCANDO CON TUX

Gli appassionati di informatica che erano ragazzi negli anni '80 non possono essersi dimenticati del film di fantascienza "Tron" prodotto dalla Walt Disney, il quale in maniera inspiegabile viene raramente riproposto sulla TV italiana.

Le molteplici avventure vissute dai protagonisti hanno da sempre ispirato numerosi videogiochi, ma adesso l'avvento del 3D ci porta veramente nel vivo dell'azione!

Diversi i progetti opensource su questo tema degni di nota, ma due mi hanno veramente stupito: gltron e armagetron.

Se avete una Linux-Box con una decente scheda video a portata di mano non aspettate altro: andate, scaricate e... all'incrocio non date la precedenza a chi viene da destra! ;)



Avventura a Udine

AMIGA Alpe Adria 2003

di Mattia Tristo [mattia_tristo@libero.it]

I nostri prodi inviati Mattia Tristo e Andrea Bernardi si sono ritrovati in quel di Udine per assistere ad uno dei raduni italiani per utenti Amiga... e non solo

Sabato 5 Giugno 2003, arrivati nella calorosa città di Udine, ci apprestiamo ad individuare la sede della Cloanto. Chi di voi non ha mai sentito parlare di questa nota software house friulana, quella del famoso word processor Text Craft per Amiga? Sì, proprio lei. Con le dettagliate istruzioni reperite nel sito della Cloanto stessa, trovare la via non è stato così difficile e in men che non si dica ci posizionavamo nel banco a noi riservato con due PC, pronti a dare dimostrazione delle capacità del sistema operativo BeOS. Appena accese le macchine, una nutrita schiera di visitatori si radunava attorno a noi, incuriositi dalle caratteristiche multimediali del SO, che stavamo cercando di evidenziare eseguendo più filmati compressi in contemporanea e allo stesso tempo riproducendo alcuni remake in mp3 dei SID del glorioso C64, che ben si prestavano all'occasione visto il luogo ed il pubblico.

Il tempo scorreva velocemente e, tra una spiegazione e l'altra, cercavamo di trovare lo spazio per ammirare il nuovo Amiga OS 4.0 su hardware PowerPC A4000 e AmigaOne, il sistema operativo MorphOS su hardware PowerPC Pegasos e il sistema riconfigurabile C-One (compatibile Commodore 64 e altri computer 8 bit).

Trascorreva così la mattina, accompagnati dalla bellissima presentazione dell'AmigaOS di Jürgen Schober. Nel primo pomeriggio giungeva il nostro turno e ci veniva comunicato che il tempo a nostra disposizione sarebbe stato ridotto a 30-45 minuti a causa della necessità di programmazione delle altre



Dall'alto Mattia e Andrea, Jens Schönfeld presenta il C-One, un nutrito gruppo di curiosi e in basso un Amiga PPC e due motherboard AmigaOne. Nella pagina accanto: ancora Mattia e Andrea che presentano la distribuzione BeOS++ di Ken, Linux su hardware AmigaOne e un momento della presentazione di AmigaOS 4.0.

SPECIALE

presentazioni, ma subito dopo aver iniziato, i presenti si dimostravano veramente interessati e le brevi spiegazioni seguite da una dimostrazione su grande schermo della velocità e delle capacità di BeOS come piattaforma veramente multitasking e multithreading, riscuotevano un degno successo. Abbiamo così cercato di evidenziare come il SO sia veloce, facendo il boot in meno di 10 secondi e lo shutdown in minor tempo ancora, sottolineando la facilità di installazione sia di una distribuzione che delle applicazioni, in quanto quest'ultime operazioni si risolvono in una semplice copia di file. Abbiamo evidenziato inoltre la stabilità del sistema, mostrando in pratica che anche in caso di crash di un'applicazione o addirittura del desktop, è possibile continuare a lavorare con il resto delle applicazioni ed in poche mosse tornare ad essere operativi al 100%, ripristinando l'intero sistema senza bisogno di riavviare. La prova lampante del successo riscontrato da BeOS veniva nel finale della presentazione, quando giungeva l'ora delle domande da parte degli spettatori. Lieti di rispondere, abbiamo continuato a mostrare caratteristiche pratiche del SO, a parlare del passato e del futuro di BeOS, evidenziando il fatto che, con la prossima uscita di Zeta della YellowTab e con il proseguirsi dello sviluppo di OpenBeOS, le migliorie e le innovazioni sarebbero state notevoli, andando ad incrementare ancora di più il fascino di questo sistema operativo. Abbiamo inoltre fatto notare che il supporto hardware verrà presto ampliato, andando così a parlare anche oltre il tempo in precedenza stabilito e occupando circa un'ora invece dei 30-45 minuti a noi riservati.

La nostra opinione si è quindi rafforzata, vedendo tanto interesse e curiosità attorno a questo stupendo sistema operativo: BeOS ed il suo successore Zeta sono una chiara risposta a quelle che sono le esigenze concrete dell'utilizzatore di personal computer, cioè velocità (di boot, di shutdown, di esecuzione), stabilità, capacità multimediali, facilità di installazione, utilizzo e manutenzione. Siamo sicuri che Zeta e in futuro OpenBeOS faranno centro, cogliendo e rispondendo a tutte quelle necessità che sorgono dall'uso di altri e ben più noti sistemi operativi e che questi, tuttavia, non riescono a colmare, creando anzi insoddisfazione.

Una nota conclusiva; lo stupore nasce dal fatto che BeOS non sia più stato "ufficialmente" supportato e migliorato dalla fine del 1999 e, malgrado ciò, riesca ancora a stupire e cogliere l'attenzione dell'utente, restando competitivo nello scenario dei sistemi operativi, segno di una struttura interna ed una organizzazione del SO veramente eccezionale, un capolavoro dell'ingegno creato da quei veri e propri geni che formavano la Be Inc.

Queste non sono soltanto le parole di due entusiasti di BeOS, ma corrispondono anche a realtà, vedere per credere [<http://www.yellowtab.com>]. Ringraziamo la Cloanto per averci permesso di partecipare a questo evento.

[<http://cloanto.com/>]. ■



Le foto pubblicate in questo servizio sono opera di:

Andreas Bornmann
Claudio Marro Filosa
Daniel Beorchia
Jens Schönfeld

Per un vasto reportage fotografico visitate il sito:
<http://www.cloanto.com/events/aaa/gallery.html>



MAGAZINE BEYOND
beos - amiga - osx - linux - freebsd

free magazine per utenti di sistemi BeOS - Amiga - OSX - Linux - FreeBSD
Distribuzione gratuita

beyond.itbug.org - beyond@itbug.org

BEYOND

© 2003 ITBug - Italian BeOS User Group
All rights reserved worldwide

Tutti gli articoli pubblicati e le loro traduzioni sono © di BEYOND e dei rispettivi autori
È VIETATA LA RIPRODUZIONE ANCHE PARZIALE SENZA L'AUTORIZZAZIONE DI BEYOND

Tutti i marchi e i nomi sono © dei rispettivi proprietari
