

Spring, la primavera di Solaris

Si parla molto di quest'ultima realizzazione di SunSoft. Al di là del nome e del titolo, comunque, si tratta solo di uno studio interno per verificare gli effetti della coesistenza di più ambienti, un territorio ben caro all'azienda. Nulla di commerciale, quindi, e nulla che possa portare in tempi brevi ad una versione microkernel di Solaris

di Leo Sorge

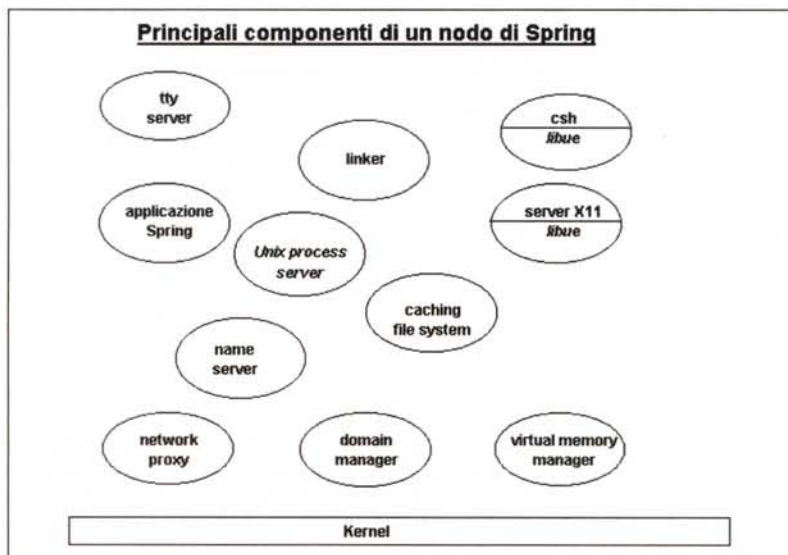
Nel tempo le funzioni dei sistemi operativi sono andate aumentando, così come la loro complessità, impostata su livelli successivi tra loro comunicanti solo se contigui (teoricamente, visto che nella pratica erano numerose le scorcioie, brutte ma funzionali, imposte dalla compatibilità con il passato, dalla necessità di prestazioni particolari o più semplicemente dall'esiguità del budget a disposizione del capoprogetto). Le enormi dimensioni crearono dei mostri difficili da installare ed amministrare, per cui si rendeva necessario un nuovo approccio: qualche anno fa uscì fuori un nuovo indirizzo, il microkernel, che sembrava interessante. Lo spunto venne dall'osservazione dell'andamento della potenza dell'hardware in rapporto alla inamovibilità del software di base. I sistemi operativi tradizionali sono basati su unità di elaborazione pensate quando la potenza degli attuali processori non era disponibile. Ma la tecnologia attuale ha sensibilmente ridotto il tempo di esecuzione, quindi queste unità risultano troppo grosse non solo per i moderni server, ma spesso anche per le workstation. D'altro canto anche la polverizzazione dei dati in rete, e le elaborazioni diverse ancorché integrate alle quali assistiamo oggi, non erano certo prevedibili allorché dal Multics venne estratta la costola Unix.

La tradizionale unità di elaborazione di Unix è il processo, definito in modo assai ricco per essere veloce al momento di passare da una esecuzione alla successiva. Dovendo velocizzarne l'esecuzione si potevano fare due scelte: accelerare il sistema (dal punto di vista software) oppure spezzare il processo. Nel primo caso abbiamo avuto pro-

poste per nuovi nuclei di sistema operativo, detti microkernel in quanto riducono al minimo i servizi offerti dal sistema per portare tutto il resto al di fuori del nucleo (kernel), nello spazio dei normali programmi utente; nel secondo caso il processo è stato frammentato in più agili unità dette thread, generando sistemi operativi e di gestione di dati detti multithread.

Se ormai tutti i sistemi moderni reclamano un'identità multithread (OSF/1, Windows NT, Windows 95...) non tutti sono ancora pronti all'approccio microkernel, non foss'altro che la sua messa a punto si è mostrata piuttosto difficoltosa e comunque penalizzante dal punto di vista delle singole presta-

zioni. Infatti il prefisso micro, nel nostro caso, trae in inganno, in quanto le piccole dimensioni richiamano subito una maggiore velocità di esecuzione. In realtà è esattamente il contrario: un microkernel è più grande e più lento d'un sistema monolitico. Pensandoci bene è presto detto: per portare nello spazio utente delle componenti tradizionalmente eseguite in modo supervisore c'è bisogno di un ulteriore straterello di software che decodifica la situazione. Inoltre le componenti così isolate hanno tutte del codice in più, non foss'altro che per le comunicazioni tra processi. La convenienza è però altrove: le singole componenti possono essere installate su macchine diverse o anche essere

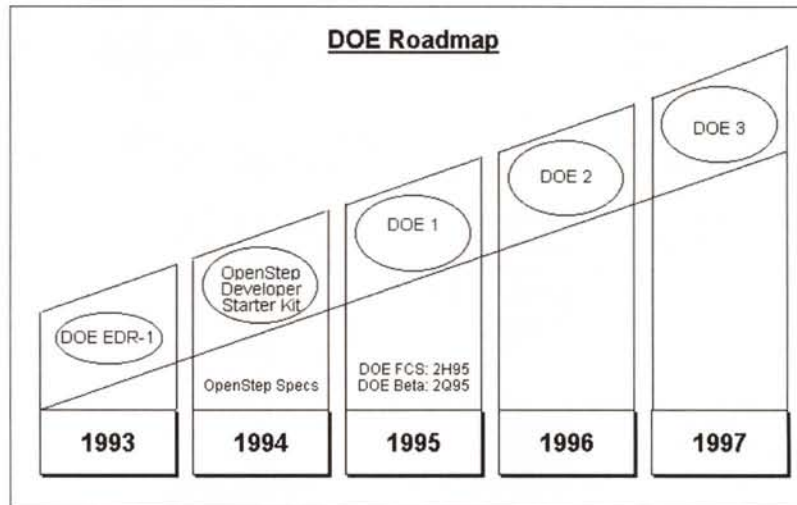


riprogettate a seconda delle esigenze, evitando i colli di bottiglia e permettendo un corretto dimensionamento del sistema, che inoltre adesso è facilmente gestibile e modellabile sulla dinamica delle necessità. Inoltre il microkernel riduce il tempo di esecuzione in modalità supervisore, permettendo un migliore sfruttamento delle risorse. In ultima analisi è vero che il microkernel, se se ne personalizzano le componenti, è più veloce in quanto sfrutta meglio un hardware comunque più ricco.

I nomi in ballo

Questo approccio ha quindi pagato nella gestione di sistemi distribuiti ed eventualmente su hardware specializzato a seconda delle componenti, ognuna delle quali, grazie alla rete, può essere allocata dove più fa comodo sia agli utenti che ai manutentori. Finora i nomi relativi a tecnologie Unix di tipo microkernel erano tre: l'americana Mach, la francese Chorus e la joint venture Taligent. L'approccio americano, pragmatico ma teoricamente non documentabile, alla fine della fiera è piaciuto solo ad OSF (e quindi a Digital) e a Next, l'azienda di Steve Jobs che lo ha usato per l'ambiente NextStep. Apparentemente maggiore il successo di Chorus, che oltre a grandi clienti nel settore di origine, le telecomunicazioni, ha firmato grandi contratti con SCO e Novell per il futuro dei rispettivi sistemi. Dal canto suo Taligent, nata da Apple ed IBM per costruire il sistema operativo del futuro, sembra in un momento di impasse: dopo l'entrata in gioco di Hewlett Packard con alcune tecnologie ed il 15% del capitale sociale le voci riferiscono del rilascio d'un sistema di sviluppo e non d'un sistema operativo.

Fatto sta che tra i grandi dell'ambiente Unix l'unica rimasta al palo per quanto riguarda il microkernel sembrava Sun, da anni decisa a sviluppare in proprio questa tecnologia. Ebbene finalmente se ne sa qualcosa di più: il nome di riferimento è Spring, una tecnologia completa che non si prevede frutti dei prodotti commerciali, ma dovrebbe essere usata come magazzino d'esperienze per i futuri progetti. Un'affermazione di questo tipo sembra strana in un mondo nel quale non si fa nulla che non venga venduto, ma in questo caso c'è una ragione: Spring è basato su un subset delle chiamate di sistema di SunOS 4.1, ed è quindi ben lontano da una commercializzazione. Ricordiamo infatti che l'attuale versione di SunOS disponibile su Solaris 2.4 è la 5, compatibile



con System V versione 4. Ciononostante le applicazioni interne sembrano molte: non bisogna dimenticare che SunSoft ha acquistato i diritti di NextStep e ne sta facendo una versione commerciale per i suoi sistemi operativi a base Solaris, quindi ha bisogno di affrontare sul tavolo l'integrazione con un ambiente nato sostanzialmente sul microkernel di Mach. Anche se sia Mach che Solaris nascono da BSD l'ambiente di SunSoft è stato del tutto appoggiato su System V versione 4.

Spring

Sono sei anni che si lavora intorno a questo progetto, che in qualche modo è la prima immagine disponibile dei futuri sistemi operativi di Sun, pensati direttamente per spazi d'indirizzamento a 64 bit. Se finora si è parlato di oggetti a partire dai dati per arrivare alle applicazioni, l'intento di Spring è di spingere la ricostruzione fino al nucleo di un sistema operativo, approfittando dell'approccio distribuito reso possibile sia dagli oggetti che dal microkernel. Contemporaneamente a questo va costruita una perfetta compatibilità con le precedenti versioni di Unix nella forma tradizionale, il che è probabilmente la parte più complessa del lavoro.

Come tutti i microkernel, anche Spring spezza la tradizionale unità di Unix in un nucleo e i server specifici. Il nucleo ha quattro funzioni di base: potente architettura di naming, un file system distribuito e coerente, un sistema di memoria virtuale piuttosto flessibile e il supporto di un certo numero di periferiche. Per il funzionamento del nodo si implementano degli altri servizi: il ge-

store dei domini, quello delle librerie dinamiche, un tty server e un proxy di rete. Anche le funzioni di base del nucleo vengono invocate da un gestore o manager.

Concettualmente il nucleo di Spring si appoggia su tre elementi: i domini, i thread e il meccanismo di comunicazione interprocesso. Il dominio somiglia molto ad un tradizionale processo Unix, però caratterizzato attraverso lo spazio d'indirizzamento al quale fanno riferimento tutti i thread nei quali è frammentato. A seconda degli oggetti un dominio può essere server o client. Se l'implementazione e il cliente non risiedono nello stesso dominio la descrizione dell'oggetto comprende una porta (door o handle) che identifica il dominio del server. I servizi di sistema sono accessibili attraverso una classica interfaccia IDL, del quale Spring usa la versione per DOE. La generazione dei meccanismi di interfaccia degli oggetti è quello consueto: dalla parte del client vengono generati gli stub (talloncini) dai quali il server desume lo skeleton (scheletro), ovvero l'interfaccia chiamata.

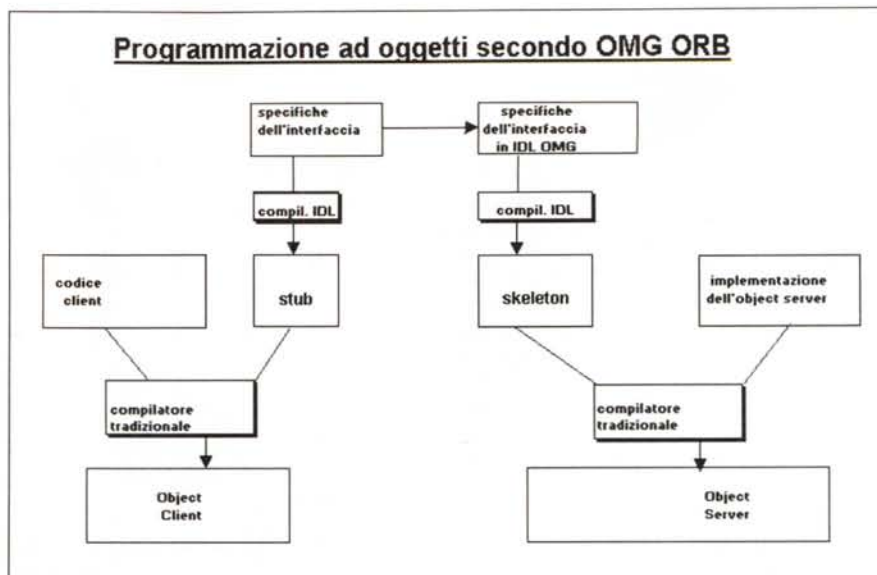
Un kernel Spring non è a conoscenza dell'esistenza di altri kernel, in quanto tutte le chiamate remote vengono gestite da un proxy di rete. Il file system ha dei meccanismi di caching per la coerenza.

La caratteristica di essere distribuito rende necessaria la gestione degli accessi e degli indirizzi, quindi Spring contiene delle componenti che svolgono queste funzioni, in particolare le ACL e il name server.

È stata fatta anche un'implementazione di Unix su Spring. Questa si appoggia sul kernel, senza però limitare

l'esecuzione. Inoltre è stata fatta particolare attenzione all'infrazione di eventuali copyright, una cosa da tenere in considerazione viste le pastoie derivate a BSDI nell'uso di codice ritenuto libero da diritti, per cui tutto il codice usato è stato generato specificamente per l'occasione. Le componenti principali dell'implementazione sono due, la libreria libue.so e UPS, lo Unix process server. UPS serve principalmente a tenere le relazioni padre-figlio, seguire gli id dei processi e dei gruppi, dare pipe e socket e instradare segnali (signals). Libue viene collegata dinamicamente ad ogni eseguibile Unix, ad esempio X11 o la shell.

Il progetto resta quindi un laboratorio le cui possibili linee evolutive comprendono l'implementazione delle chiamate di sistema mancanti e l'accesso ad oggetti non Unix. La direzione commercialmente più importante, vale a dire una migrazione verso SVR4 ed una sua completa implementazione microkernel sul modello di Chorus, non sembra verosimile. SunSoft ha già sofferto molto la migrazione da Solaris 1 a Solaris 2, che le ha portato incertezze e ritardi, che difficilmente s'imbarcherà in un'altra avventura analoga. Resta quindi da chiedersi un paio di cose: quando Sun-



Soft disporrà d'un microkernel commerciale; a cosa serve Spring.

DOE, oggetti ovunque

Pensando a NextStep, l'ambiente ad oggetti acquistato da Next e ribattezzato OpenStep, la risposta alla seconda domanda è semplice, ed ha un nome: si chiama DOE, Distributed Objects Everywhere, il meccanismo di rete implicita proposto da Sun. Il paradigma

completo comprende il sistema operativo Solaris, il brokeraggio degli oggetti attraverso Corba, i servizi di Coss, gli oggetti di OpenStep ed infine le applicazioni per utenti finali e per le aziende. In particolare gli oggetti di OpenStep sono gli strumenti di creazione delle interfacce grafiche (finestre, bottoni, etc.), il testo in formato internazionale, font, stampanti, accesso ai database e interfacce IDL.

I servizi di Coss, standardizzati nel 1984, sono quelli tradizionali dei sistemi distribuiti: naming, events, object lifecycle, transazioni e sicurezza, ai quali è stata aggiunta la gestione diretta dei database ad oggetti secondo le specifiche stilate già nel 1993 dall'Object Database Management Group interno all'OMG.

Dal punto di vista dell'utente quindi OpenStep non è altro che uno dei tipi di schermo che possono essere aperti, gli altri essendo Wabi, Mae ed ovviamente CDE ed OpenLook. Ovviamente ci si avvia alla coesistenza dei due ambienti Unix, CDE ed OpenStep: laddove il primo si indirizzerà al vasto pubblico già esistente e alle metodologie tradizionali, il secondo sarà scelto non solo per l'orientamento agli oggetti ma anche per la maggior produttività implicita e per la più alta aderenza alle necessità dell'industria.

Una riflessione sulla tecnologia ad oggetti porta delle conseguenze anche per quanto riguarda un microkernel commerciale da Sun. L'affermarsi di questo paradigma potrebbe isolare gli utenti dal sistema, in qualche modo dando per la trasformazione un po' più di tempo. Non tanto, però, visto che la concorrenza si sta già attrezzando.

Leo Sorce è raggiungibile su MC-link all'indirizzo MC6750 e su Internet all'indirizzo leo.sorce@mclink.it

Glossario

CORBA, Common Object Request Broker Architecture: interfaccia di richiesta d'informazioni ad oggetti.

DCE, Distributed Computing Environment: servizi fondamentali dell'ambiente distribuito di OSF, comprendenti gestione di eventi, stampe, distribuzione e licencing del software, sottoreti.

DOMF, Distributed Object Management Facility: proposta di SunSoft ed HP accettata come intermediario (broker) dell'architettura Corba 1 di OMG.

MAE, Macintosh Application Environment: il sistema operativo Apple montato sul file-system Unix e con emulazione di altri microprocessori.

OLE, Object Linking and Embedding: tecnologia per la multimedialità proposta da Microsoft. Nella versione 1 si trova su Windows 3.1 e su Lotus Notes 3.1, mentre la versione 2 sarà su Chicago ed NT.

OMF, Object Management Framework: gestione di rete del DME che si appoggia sull'architettura ad oggetti CORBA dell'OMG.

OpenDOC: architettura multimediale e multiplatforma Apple per documenti composti con testo, grafica e video. È l'analoga di Microsoft OLE 2.0.

OSF, Open Software Foundation: associazione senza scopo di lucro nata nel 1988 per promuovere i sistemi aperti con un processo decisionale altrettanto aperto.

QuickTime: gestione del video all'interno di documenti multimediali nell'architettura OpenDoc.

RPC, Remote Procedure Calls: chiamate a procedure remote, ovvero localizzate non sulla macchina chiamante ma su un'altra in rete.

System V: lo Unix prima di At&t, poi di Novell ora confluito in X/Open CAE.

Taligent: joint venture tra Apple ed IBM, cui poi si è aggiunta HP, tesa allo sviluppo d'un sistema operativo ad oggetti distribuiti.

WABI, Windows Application Binary Interface: il prodotto di SunSelect per emulare Windows 3.1 sotto Unix.

Win16: l'insieme di regole di accesso alle risorse di Windows 3.1. La loro emulazione sotto X-Windows può consentire l'esecuzione di applicativi Windows sotto sistemi operativi diversi dall'Ms/Dos, purché non si acceda direttamente all'hardware.

Win32: l'insieme di regole di accesso alle risorse di Windows 95 ed NT.

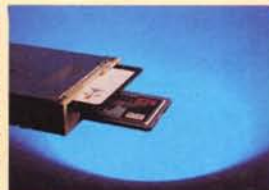
PCMCIA Kernel. La via maestra al Plug-N-Play.



Mass Storage



Hard Disk
Dischi rigidi di semplice utilizzo disponibili in varie capacità da 130Mb a 260Mb senza Stacker.



PCMCIA Drive
Lettori PCMCIA interni ed esterni per sistemi desktop. È disponibile anche un drive con uno slot e un drive per floppy.



Plug-N-Scan
Scanner manuale a 256 livelli di grigio e risoluzione da 100 a 400 dpi. Non necessita di alimentazione esterna.



SCSI Card
Interfaccia SCSI certificata Corel. La card è dotata di connettore SCSI II ed è fornita di adattatore Centronics.



Speech Card
Card di riconoscimento vocale. Consente la dettatura e la navigazione in ambiente Windows. Dizionari legale, medico, etc.



Multimedia



Type I-Type III Adapter
Adattatore per utilizzare una card di tipo III anche su personal che dispongono del solo slot di tipo I.



Sound Card
Card stereo 16 bit, AdLib e Games compatibile. È disponibile anche la versione Combo con interfaccia SCSI.



Video Capture Card
Scheda per acquisire singoli frame o sequenze. Gestione stereo dell'audio. Richiede SVGA 640x480, 256 colori.



VVL Minicamera
Telecamera B/N miniaturizzata alimentata direttamente dal portatile. È fornita con un flessibile kit di sviluppo.



Communication & Connectivity



Combo Card
Card multifunzione EthernetModem 14.4 e 28.8 anche con supporto cellulare e EthernetFlash da 2MB.



Emulation Card
Linea completa di card per la connessione locale o remota a Mainframe, AS 400 e Grandi Sistemi.



Ethernet Card
Card di connessione di rete 10 Base T, 10 Base 2 e Combo. Tutte le card sono certificate Novell.



Cellular Cables
Cavetti di connessione diretta con i nostri Fax/Modem 14.4 e 28.8 per i più diffusi modelli di telefono cellulare.



Fax/Modem Card
Una linea completa per tutte le esigenze: 24/96 BPS, 14.4 BPS e 28.8 BPS V34 con o senza supporto cellulare.



GPS Card
Sistema di posizionamento satellitare con approssimazioni inferiori ai 100 m. La card è fornita completa di antenna.



Serial & Parallel Card
Seriali RS 232 o RS 422/485 sia in versione singola che doppia ed interfacce parallele.



ISDN Card
Card tipo II che consente la connessione a qualsiasi access point ISDN con standard S0 o ad una rete X25.

Nome _____ Cognome _____
 Azienda _____
 Indirizzo _____
 Città _____
 Tel _____ Fax _____

Avere altre informazioni sulle PCMCIA Kernel è facile. Basta barrare le card che interessano e presentare questa pagina nel Punto Affiliato più vicino, oppure inviarla per posta o fax a:
KERNEL Consulting
 via di S. Giovanni in Laterano, 262
 00184 Roma Tel. 06/77207000 Fax 06/77205480



Le card vengono fornite complete di software e cavi di connessione e sono tutte garantite a vita franco Ns uffici, eccetto gli hard disk che sono garantiti un anno. Tutti i marchi citati sono registrati dai rispettivi proprietari.

PIEMONTE - Omega, 011/882701 TORINO • **LOMBARDIA** - Supertronic, 02/27208200 MILANO • **CRC Sistemi**, 0383/49078 VOGHERA (PV) • **FRIULI V. GIULIA** - Murrisoft, 040/369441 TRIESTE • **VENETO** - Cartocontabile, 0423/497333 CASTELFRANCO V. (Tv) • **LIGURIA** - CIDI Office, 010/584279 GENOVA • **Comunicazione & Sistemi**, 010/5705876 • **TOSCANA** - Elettronica Centostelle, 055/610251 FIRENZE • **Elettronica Centostelle**, 055/432694



FIRENZE • **LAZIO** - Gam Computer, 06/78347334 ROMA • **MA.NA.** Elaboratori, 06/44244714 ROMA • **Redwood Microlink**, 06/88642132 ROMA • **P.S. Italia**, 06/9352524 S.M. DELLE MOLE (Rm) • **CAMPANIA** - Informatica Meridionale, 081/7611130 NAPOLI • **General Computer**, 089/251903 SALERNO • **PUGLIA** - Progetti Informatici, 099/350602 TARANTO • **SICILIA** - Data Studio, 091/901785 BAGHERIA (Pa) • **Zuleima**, 0924/505318 ALCAMO (Tp)