

Appunti di programmazione

La programmazione del Macintosh

Come forse ricorderete, la volta scorsa abbiamo parlato della disponibilità sul mercato di pacchetti capaci di creare ambienti di programmazione orientati all'oggetto efficienti e di grande utilità. Tra di essi lo stato dell'arte è rappresentato da MPW, MPW C e Symantec C, ambienti, tutti e tre, capaci di fornire un adeguato supporto per la costruzione di applicazioni efficienti e ben disegnate in ottica dell'interfaccia Mac. Probabilmente si tratta del meglio attualmente disponibile sul mercato e, fino a poco tempo fa, ognuno di essi poteva dare punti a tutto quanto disponibile in MS-DOS. Ma sono davvero equivalenti, o ognuno di essi ha qualità (e applicabilità) particolari?

Le caratteristiche degli ambienti a confronto

La scelta dell'ambiente è condizionata, prima di tutto, dalla struttura del gruppo di programmazione che utilizzerà l'ambiente stesso. Se sulla stessa applicazione lavorerà più di una persona, MPW C è la scelta obbligata; Think C sotto questo punto di vista è molto difficoltoso da utilizzare. Se invece si tratta del lavoro di una sola persona, o, al massimo, di due che lavorano in parallelo, Think C è preferibile.

Ma non è questa l'unica differenza; nel caso di costruzione di applicazioni fortemente interattive, MPW C mette a disposizione tool più efficienti, ma è più difficile da maneggiare di Think, che è più «user friend». Quest'ultimo ha una velocità di compilazione estremamente alta (fino a 100.000 linee al minuto) ma questa sua facoltà perde molta efficacia, ovviamente, se il codice sorgente presenta pochi bug. Al contrario MPW C ha a disposizione diverse funzioni di controllo del codice, che, inevitabilmente, si trasformano in un lieve rallentamento della procedura di compilazione stessa.

A proposito di compilatori

Ambedue i produttori affermano che i loro compilatori C non sono conformi agli standard ANSI, e tutti e due forniscono una documentata tabella delle differenze. Accanto a ciò c'è l'estrema limitatezza delle librerie, piuttosto ridotte in ambedue i casi (mancano diverse funzioni addirittura considerate standard per un linguaggio C). Questo può portare a diversi problemi in quanto una libreria potrebbe non funzionare nel modo che uno si aspetterebbe. Inoltre, cosa questa comune a moltissimi linguaggi, occorre avere sempre ben in mente i limiti della notazione in virgola mobile, specie nei casi in cui è necessaria una

elevata precisione numerica (esemplare a tal proposito una calcolatrice, si fa per dire, dei linguaggi, lo ZBasic, che può raggiungere una precisione di ben 20.000 cifre! Ancora, ambedue hanno notazioni numeriche di tipo [int] e [double] piuttosto scadenti e, soprattutto, differenti tra loro. Think C usa una notazione a 16 bit per gli interi, mentre MPW usa 32 bit; il primo, inoltre, offre una nuova notazione [extended] per i dati del tipo [long double]; come se non bastasse esiste anche una differenza, in fatto di precisione, per i tipi di dati a precisione più ampia, in funzione della presenza della cosiddetta unità di precisione in virgola mobile [FPPU]. Questa usa un formato esteso di rappresentazione a 96 bit per strutturare la possibilità del microprocessore 68020 di accedere alle cosiddette longword. Per ambedue i compilatori, comunque, occorre settare delle specifiche opzioni di compilazione per usare la FPPU; in altri termini la scelta non è automatica, sebbene il sistema operativo Mac abbia la possibilità, autonoma, di riconoscere la presenza dell'FPPU.

Esiste ancora una differenza riguardo alle tecniche di valutazione delle espressioni; Think C converte sempre tutti i numeri in doppia precisione prima di eseguire i calcoli (salvo poi ridimensionarli alla grandezza primitiva) mentre MPW converte solo alla più alta delle precisioni che coinvolgono i numeri. Questo può portare, per programmi «number cruncher» a risultati leggermente diversi.

Altri problemi, sebbene di più ridotta entità, possono sorgere dalla manipolazione degli array. Ambedue gli ambienti supportano una completa architettura a memoria segmentata, usando offset a 16 bit. Il risultato si traduce nella non portabilità di una applicazione sviluppata in memoria virtuale che usa grosse array statiche, direttamente e senza alcuna modifica (ognuno ricorderà quanto

Indirizzi utili**Bowers Development**

AppMaker
P.O. Box 9, Lincoln Center, MA 01773,
(508)369-8175

Holder, Egan & Company

MacInterface
4148 Spring Hill Road, Midland, MI
48640,
(800)782-9976

Masters Publishing

Professional Programmer's Extender
3571 Pheasant Run, #7, Ann Arbor,
MI 48108-2460, (313)971-1118.

MMC AD Systems

McCLint; McCPrint; C Programmer's
Toolbox
P.O. Box 360845, Milpitas, CA 95036,
(415)770-0858

SmetherBarnes

Prototyper
520 SW Harrison, Suite 435,
Portland, OR 97201, (503)274-7179

Symantec Corporation

Think C
10201 Torre Avenue, Cupertino, CA
95014, (800) 441-7234,2.

problematica fosse la manipolazione degli array fino alla invenzione delle architetture, appunto, a memoria segmentata).

Il problema è, però più nominale che reale; per superarlo basta avere l'accuratezza, quando si trasporta codice tra architetture diverse, di ricordare che, generalmente, passare all'ambiente Mac significa migliorare e non peggiorare, quanto a tool di utilizzo. Il caso più frequente è il trasferimento di codice da Unix a Mac, che, solo col toolbox, dispone di routine, funzioni e librerie ben più articolate e complete.

Cosa non nuova nella famiglia informatica, anche in questo caso succede che, una volta scelto un ambiente il programmatore è portato a considerare l'altro come estraneo al suo modo e schema di sviluppo. Effettivamente esistono grosse differenze nel file [include], nei calcoli in virgola mobile, e nelle chiamate al sistema operativo. Uno dei bug, ad esempio, più interessanti in MPW (solo fino alla versione 3.0) era uno strano blocco del compilatore (con generazione di messaggio d'errore) in corrispondenza degli operatori [++] e [--], peraltro standard anche nel più scassato dei linguaggi. Sebbene Apple abbia risolto il problema nella versione 3.1 (che peraltro supera una serie di ulteriori fastidiosi errori), questo portò spesso l'utente a scartare i prodotti sviluppati con la versione 3; il motivo è presto detto; il programmatore, per superare l'impasse dell'errore del compilatore, utilizzava un codice alternativo per la simulazione degli operatori incriminati, dimenticando che gli operatori in parola sono tra i più veloci ed efficienti a disposizione.

Think C comprende infine una sezione (con relativi operatori) dedicata alla programmazione orientata all'oggetto. Questa versione di C è strettamente conforme agli standard del C++. Esso dispone, inoltre di un set di librerie davvero raffinato e completo, tanto da rap-

presentare, oggi, un vero cavallo di battaglia nella programmazione OO.

Per entrare nell'ambiente Mac

Se si è davvero interessati a una seria programmazione del Mac, è il caso di iscriversi all'APDA, Apple Programmer Development Association, attraverso

cui è possibile acquistare MPW e MPW C, usufruire di una continua serie di informazioni e conoscere prodotti e software realizzati da terze parti.

Se si desidera divenire un programmatore Mac, è necessario immediatamente saper realizzare una completa ed efficiente interfaccia Mac. La cosa non è facile, anzi è estremamente difficile; ma non disperiamoci per questo. Certo, se si desidera realizzare tutto da sé, l'impresa risulta quasi impossibile, vista l'immane fatica occorrente per la costruzione di menu, finestre, dialoghi, e così via. Per fortuna sono disponibili numerose librerie linkabili, che forniscono (spesso anche a livello di sorgente) la maggior parte del codice di cui si ha bisogno. Esempi del genere sono Professional Programmer's Extender (395\$) o Mac Interface (un centinaio di dollari in meno), che mettono immediatamente a portata di mano menu, finestre, routine di stampa e un agevole interfacciamento con Multifinder.

Prototyper e AppMaker vanno ancora più avanti e rendono il lavoro di gene-

Confronto tra le caratteristiche di rappresentazione interna di MPW e Think**MPW C**

	Bit	Minimo	Massimo
int	32	2.14E-9	2.14E+9
float	32	1.5E-45	3.4E+38
double	64	5.0E-324	1.7E+308
double ext	80*	1.9E-4951	1.1E+4932

THINK C

	Bit	Minimo	Massimo
int	16	-32,768	32,767
float	32	1.5E-45	3.4E+38
double	64	5.0E-324	1.7E+308
double ext	80*	1.9E-4951	1.1E+4932

*: diviene a 96 bit se viene utilizzata la notazione FPUU.

razione dei programmi un vero piacere, attraverso una interfaccia grafica eccellentemente realizzata. Questi CASE possono poi generare automaticamente un codice sorgente C necessario per creare e sviluppare una interfaccia utente personalizzata.

È probabile che a questo punto tutto sia pronto per l'avvio del lavoro. Prima di selezionare un metodo, comunque, per realizzare la propria interfaccia, occorre aver ben chiaro in mente il layout dell'interfaccia stessa, il suo sviluppo operativo, e quali primitive d'interfaccia saranno necessarie. Il Mac ha un certo numero di primitive già realizzate, tra cui:

- layout di schermo, con finestre, dialoghi, e messaggi di avviso.
- Controlli, come bottoni, edit-text (testo cliccabile, editabile e rifelezionabile), icone (sia di sistema che definite dall'utente), figure, linee rettangolari (a spigolo acuto e arrotondato, palette, scrollbar, controllo di risorse, tool direttamente definiti dall'utente e altro).
- Menu, del tipo semplice, gerarchico, autosrotolantesi, ecc.

Per poter mettere ordine nel prevedibile marasma che si genererebbe da una disponibilità di tal fatta, Apple ha stabilito una serie di regole che si suppone ogni implementatore debba seguire. D'altro canto è proprio il fatto stesso di adottare le stesse regole che ha determinato la fortuna dell'interfaccia Mac. Il set principale di queste regole è contenuto in *Inside Macintosh* e, comunque, in una serie di pubblicazioni piuttosto diffuse. È infine prassi comune, una volta realizzata una interfaccia Mac, di compararla con quella delle applicazioni più diffuse (generalmente lo standard è rappresentato da MacWrite-MacPaint-MacDraw. Come dicevamo in precedenza la mancanza di una interfaccia nota è un grave freno alla utilizzabilità di una applicazione.

Alcuni criteri sulle modalità di sviluppo di una applicazione

Come in ogni abbinata macchina-linguaggio, ma soprattutto utilizzando il Mac, ci si rende conto fin dall'inizio che ci sono diverse maniere per affrontare un problema.

La notevole flessibilità di programmazione della macchina potrebbe portare alla realizzazione di una interfaccia che funziona bene per un certo problema, ma si tratta in ogni caso di un rischio, visto che questa interfaccia particolare, e fuori standard, potrebbe non essere più sufficiente e adeguata per una futura release della stessa applicazione (confes-

so di essere caduto vergognosamente nello stesso errore); ridisegnare l'interfaccia daccapo mi è costata una fatica immane, tanto che alla fine decisi di lasciar perdere tutto e di ricostruire daccapo la finestra completa. Inoltre occorre tener presente che la modifica dell'ambiente o anche solo del compilatore potrebbe creare interferenze e problemi con la particolare interfaccia costruita; se invece si adottano regole standard di costruzione, è garantito che ogni nuova release del linguaggio-ambiente di sviluppo supporterà quello precedente (nessun costruttore sarebbe tanto folle da abbandonare la struttura di base che ha adottato per il suo linguaggio).

D'altro canto, oggi, a distanza di sette anni dalla nascita della macchina, si ha a disposizione una bibliografia enorme, e una messe di materiale, anche a livello di codice sorgente, estremamente articolata. Molto materiale si ritrova sulle riviste specializzate (Mac Tutor, Mac Week, Macintosh Today, MacWorld, ecc), in letteratura (vedi riferimenti) o ricavandolo dai bulletin board, materiale che molto spesso è possibile compilare tal quale o con modeste modifiche; l'analisi di queste pubblicazioni rappresenta, inoltre, una palestra programmatoria raffinata, che, oltre a mettere a disposizione codice compatto e ben fatto, insegna regole di vita (programmatoria) molto utili. Occorre comunque tener presente che lo standard nel campo dei codici sorgente è rappresentato da quello compilabile dalle applicazioni precedentemente nominate (MPW e Think generalmente sono piuttosto compatibili in termini di compilazione; Aztec C, di gran lunga l'ambiente più potente include quasi tutto compreso nei due precedenti; altri ambienti potrebbero essere parzialmente incompatibili); almeno all'inizio, non porsi grandi progetti; come consigliava una volta Guy Kawasaki sulle pagine di Mac User, volare basso per non prendere grandi cadute.

Infine un consiglio personale, matura-

Bibliografia

- *Macintosh Development Tools & Languages Guidebook, 1991 Edition*, e
- *Apple Engineering-Scientific Solutions Guide. Winter 1989/90*. Pubblicati da Apple Computer, Inc.
- *Inside Macintosh. Volumes I-VI* (Addison-Wesley Publishing Company).
- *The Macintosh Bible*, Arthur Naiman, Editor (Goldstein & Blair).
- *MacTutor*. Rivista di programmazione mensile dedicata a Macintosh, pubblicata in diverse lingue (714)7771255.
- *Macintosh Buyer's Guide o Macintosh Product Registry*. Catalogo quadrimestrale pubblicato da Redgate Communications Corporation.
- *Apple's Programmer's Development Association*; (800)2822732.

to in tanti anni di uso e programmazione del Mac; inutile illudersi di poter portare codici scritti per altre macchine in maniera facile e veloce; in barba a tutta la decantata portabilità permessa da certi linguaggi, è proprio la presenza dell'interfaccia Mac a costringere a lavoro talora estenuante per ottenere risultati appena accettabili; molto meglio utilizzare, del vecchio programma, il solo algoritmo risolutivo e affrontare, ex novo, la completa programmazione della struttura di interfaccia (una delle più complesse operazioni di programmazione, sembrerà strano, è proprio quella della realizzazione di una finestra di input veloce e facilmente gestibile).

Per concludere

La realizzazione di una interfaccia veramente efficiente, in Mac, non è né semplice né veloce, qualunque sia il linguaggio adottato; abbiamo più volte nominato in queste note il Think C e l'MPW in quanto per una serie di circostanze e per il buon supporto che offrono i costruttori, sono oggi divenuti lo standard de facto dell'ars programmatoria su Mac. Scomparso ormai dalla scena il Pascal di Borland (che peraltro non aveva mai brillato), ci sono ancora diversi linguaggi da adottare, tra cui il Prototyper (più un ambiente che un linguaggio; comunque molto divertente da esplorare), SmallTalk/V, Spinnaker Plus, ProGraph e l'eccellente ZBasic. Ma la cosa più importante, quando si decide di intraprendere un progetto di una applicazione è una, forse ovvia: conoscere quello che offre la concorrenza. È inutile scrivere anche una sola linea di programma, se non si è prima ben sicuri di cosa si potrà dare di più rispetto a quello che offre il mercato. Se si è capaci di creare un prodotto significativamente migliore, è ben difficile che la nostra applicazione possa avere successo sul mercato!

MR

MASTER ELETTRONICA

Via Valentini, 96/B/C - 50047 PRATO (FI)
Tel. 0574/34352 - Fax 0574/36652

VENDITA AL MINUTO E PER CORRISPONDENZA - CONTATTATECI GARANTIAMO QUALITA' CORTESIA COMPETENZA.
TUTTI I NOSTRI PRODOTTI SI INTENDONO CON GARANZIA ORIGINALE IVA INCLUSA - ORARIO 9,00-12,30 / 15,30-19,30
- SABATO APERTO - TUTTI I PRODOTTI SONO DISPONIBILI PRESSO I NOSTRI MAGAZZINI

COMMODORE

A500	Lit.	7 19.000
A500 APPETIZER	Lit.	735.000
A500 + ESP 512 Kb	Lit.	810.000
A590 HD 20 Mb	Lit.	720.000
A2000	Lit.	1.550.000
A2000 + 2 DRIVER	Lit.	1.650.000
A2000 + A 2058	Lit.	2.250.000
SCHEDA JANUS XT	Lit.	550.000
SCHEDA JANUS AT	Lit.	1.350.000
SCHEDA PROCESSORE A2630	Lit.	2.900.000
SCHEDA VIDEO	Lit.	130.000
SCHEDA GENLOCK A2300	Lit.	285.000
SCHEDA A2320 Flicker-fixer per Amiga 2000	Lit.	400.000
C 64 NEW	Lit.	260.000
DRIVER 1541 II	Lit.	270.000

ACCESSORI PER AMIGA

ESPANSIONE 512 KB+CLOCK+INT	Lit.	110.000
ESPANSIONE 2-8 Mb per A500	Lit.	620.000
ESPANSIONE 2 Mb per A1000	Lit.	640.000
ESPANSIONE 2-8 Mb per A2000	Lit.	600.000
ESPANSIONE A 2058 per A2000	Lit.	760.000
DRIVE 3,5 MECC NEC+DISCONNET	Lit.	180.000
DRIVER 3,5 DISCONNET	Lit.	160.000
MIDI DI NOSTRA PROD.	Lit.	90.000
DIGITALIZZATORE AUDIO STEREO	Lit.	170.000
DIGI-VIEW GOLD 4.0	Lit.	300.000
DE LUXE-VIEW	Lit.	350.000
FATTER AGNUS 8372/A	Lit.	180.000
GENLOCK PNL 20	Lit.	590.000
GENLOCK Y - C PAL	Lit.	940.000
FILTRO ELETTRONICO "SPLITTER"	Lit.	450.000

MONITOR

BM 7513 RGB TTL F.V	Lit.	175.000
BM 7502 CVBS F.V	Lit.	180.000
BM 7923 FSQ RGB TTL. F. AMBRA	Lit.	195.000
CM 11342 / 1DG	Lit.	500.000
COMMODORE 1084S	Lit.	490.000
MULTISYNCH IIID NEC	Lit.	1.300.000

HARD-DISK A 2091/A

AUTOBOOTING PER AMIGA 2000 Lit. 295.000

50 Mb CCSI Quantum	Lit.	680.000
80 Mb SCSI Quantum	Lit.	850.000
105 Mb SCSI Quantum	Lit.	1.117.000
210 Mb SCSI Quantum	Lit.	1.890.000
Espansione 2 MB SCSI e A2991A o A590	Lit.	180.000

AMIGA 3000

AMIGA 3000 25 Mhz 40 mb	Lit.	6.030.000
AMIGA 3000 25 Mhz 100 mb	Lit.	6.850.000

STAMPANTI

STAMP. PHILIPS NMS 1433	Lit.	400.000
STAMP. PHILIPS NMS 1461	Lit.	790.000
STAMP. PHILIPS NMS 1467	Lit.	1.000.000
NEC P20 PLUS	Lit.	750.000
NEC P30 PLUS	Lit.	1.011.000
NEC P60 PLUS	Lit.	1.250.000
NEC P70 PLUS	Lit.	1.580.000
MPS 1230	Lit.	320.000
MPS 1270	Lit.	360.000

OFFERTA PHILIPS

NMS 9115 XT 8088 4.7-8 Mhz, 768 Kb Ram, 1 Fdd 3,5"
720 Kb, 1 HD 20 Mb, Scheda Video ATI-CGA-HERCULES-
MDA, Tastiera avanzata 101 tasti, Seriale, Parallela, Ms-
Dos 3.30. Lit. 1.199.000

PARTI STACCATE

Tastiera avanzata 102 tasti	Lit.	90.000
Tastiera avanzata 102 tasti "Cherry"	Lit.	115.000
Case Desktop, Display, alim. 200 watt "Lux"	Lit.	215.000
Case Desktop, Display, alim. 200 watt "Bey"	Lit.	230.000
Case MiniTower, Display, alim. 200 watt	Lit.	252.000
Case Tower, Display, alim. 200 watt	Lit.	390.000
Mainboard 80286 - 16 Mhz - 0 Mb Ram	Lit.	250.000
Mainboard 80386sx - 16 Mhz - 0 Mb Ram	Lit.	780.000
Mainboard 80386sx - 20 Mhz - 0 Mb Ram	Lit.	900.000
Mainboard 80386 - 25 Mhz - 32Ram Cache 0 Mb Ram	Lit.	1.350.000
Mainboard 80386 - 33 Mhz - 64 Ram Cache 0 Mb Ram	Lit.	1.600.000
Floppy Drive da 5,25 360 Kb	Lit.	118.000
Floppy Drive da 5,25 1,2 Kb	Lit.	128.000
Floppy Drive da 3,5 720 Kb	Lit.	125.000
Floppy Drive da 3,5 1,44 Mb	Lit.	128.000
Controller per driver XT e ATFD 360/1.2/720/1.44	Lit.	45.000
Controller per HD XT MFM	Lit.	78.000
Controller per 286/386 interlave 2:1 HDC+FDC	Lit.	115.000
Controller per 286/386 interlave 1:1 HDC+FDC	Lit.	128.000
Controller per 286/386 ESDI HDC+FDC	Lit.	222.000
Controller per 286/386 Multi/IO interface HD ATBUS	Lit.	85.000
FDC+HDC+2 RS232 Com1 e Com2, 1 Paral, 1 Game	Lit.	350.000
HD Segate St 125 28 ms 20 Mb 3,5	Lit.	425.000
HD Segate St 157/A 28 ms 42 Mb Atbus	Lit.	930.000
HD Fujitsu 90 Mb ATBUS	Lit.	1.300.000
HD Fujitsu 136 Mb ATBUS	Lit.	1.630.000
HD Fujitsu 182 Mb ATBUS	Lit.	390.000
HD Conner 40 Mb ATBUS	Lit.	630.000
HD Quantum 52 Mb ATBUS	Lit.	690.000
HD Maxtor 80 Mb ATBUS	Lit.	840.000
HD Conner 120 Mb ATBUS	Lit.	1.900.000
HD Quantum 210 Mb ATBUS	Lit.	1.900.000

**TELEFONATE E RICHIEDETE
IL NOSTRO LISTINO**