



# FREEPASCAL

Claudiu Soroiu

**Istoria sistemelor de calcul se rescrie pe zi ce trece; o dată cu aceasta se modifică și istoria limbajelor de programare. Multe dintre acestea se scufundă în marea adâncă a tehnologiei, iar altele abia acum se ridică...**

## Introducere

Tehnologia de realizare a sistemelor de calcul se schimbă destul de repede în zilele noastre, ceea ce duce, în paralel, la o dezvoltare a limbajelor de programare. Posibilitățile oferite de limbajul *Pascal*, unul dintre primele limbaje de programare de nivel înalt, sunt limitate și, nu de puține ori, programatorii care folosesc mediul *Borland Pascal* au întâmpinat dificultăți în implementarea codului sursă.

Pentru a ține pasul cu standardele impuse de nivelul ridicat al dezvoltării tehnologice, o echipă de programatori a creat o nouă implementare, numită *Freepascal*. Acest produs vine în sprijinul celor care programează în *Borland Pascal* și care au nevoie de mai multe resurse și unelte ușor de folosit pentru crearea de programe într-o perioadă mai scurtă de timp.

## Prezentare

Gama de instrumente oferite de acest produs este foarte variată. Principalele caracteristici ale acestui limbaj sunt portabilitatea pe diferite sisteme de operare cum ar fi *MS-Dos*, *Windows*, *OS/2*, *Linux*, *Amiga*, *Atari* și compatibilitatea aproape totală cu *Borland Pascal*.

Pachetele cu care este livrat produsul sunt dependente de sistemul de operare pentru care vor fi folosite. Cele mai importante unelte ale unui pachet sunt: un compilator pentru surse (*ppc386*), un compilator pentru proiecte (*make*), un *debug-er* (*gdb*) și, în funcție de pachetul descărcat, acesta include și un mediu de programare (*fp*).

## Facilități

Prima noutate apare chiar la compilator, care, spre deosebire de *Borland Pascal*, afișează lista cu toate erorile care apar în codul sursă scris de programator, al doilea afișând numai prima eroare găsită.

Programele generate de compilatorul *Freepascal* vor lucra pe 32 de biți în mod protejat, și nu pe 16 biți ca în cazul compilatorului oferit de *Borland*.

Limbajul *Freepascal* oferă programatorilor posibilitatea de a folosi aproape toate resursele sistemului. Dacă în *Borland Pascal* nu se putea crea un tablou cu dimensiunea mai mare de 64 KB, acum mărimea tabloului pe care îl utilizați este limitată doar de capacitatea memoriei sistem și a memoriei virtuale create de mediul de operare.

După cum ați observat, în alte limbaje de programare cum ar fi *C/C++* există posibilitatea supraîncărcării operatorilor și funcțiilor, lucruri imposibil de realizat în *Borland Pascal*, dar realizabile în *Freepascal*.

După cum știți, supraîncărcarea operatorilor și a funcțiilor ușurează foarte mult munca programatorului. Acesta nu mai este acum nevoit să creeze câte o funcție cu nume diferit pentru diversele tipuri de date pe care aceasta le primește ca parametri. Dacă utilizați compilatorul *ppc386*, funcțiile pot avea nu numai tipuri ordinale, reale și *pointer-i*, ci pot returna și tablouri, înregistrări sau obiecte.

Acum, numele funcțiilor pot apărea, în cadrul definiției lor, în partea dreaptă a expresiilor, fără să se apeleze recursiv. Pentru ca funcția să nu fie apelată recursiv, trebuie să nu conțină parametri în partea dreaptă a identificatorului, ci să apară numai numele funcției. În cazul în care aveți o funcție fără parametri și doriți ca aceasta să fie apelată recursiv, în partea dreaptă a expresiei, după numele funcției, trebuie să apară o paranteză deschisă "(" și o paranteză închisă ")". Printre altele, apar și tipuri noi de date cum ar fi tipul *cardinal*, număr pe 32 de biți fără semn. Variabilele de tip **string** pot avea dimensiuni de peste 255 de caractere.

## Resurse online

Informații suplimentare și documentații referitoare la acest limbaj de programare puteți găsi la <http://www.freepascal.org>, site-ul oficial dedicat limbajului *Freepascal*.

Claudiu Soroiu este redactor GInfo și poate fi contactat prin e-mail la adresa [csoroiu@yahoo.com](mailto:csoroiu@yahoo.com).