

Índice general

4. Compiladores del Lenguaje Pascal	1
4.1. Pascal User's Group (PUG)	1
4.1.1. Historia	1
4.2. Pascal-P (The Portable Pascal Compiler)	1
4.2.1. Historia Pascal CDC 6000	2
4.2.2. Historia Pascal-P	2
4.3. UCSD Pascal	3
4.3.1. Historia	3
4.4. Pascaline	5
4.4.1. IP Pascal	5
4.5. Borland Pascal	5
4.5.1. Historia	5
4.5.2. Valores internos para datos numéricos simples	6
4.5.3. Biblioteca estándar	6
4.6. GNU Pascal Compiler (GPC)	8
4.6.1. ¿Qué es GPC?	8
4.6.2. Estructura de GPC	8
4.7. FreePascal	9
4.7.1. ¿Qué es FreePascal?	9
4.7.2. Historia	9
4.7.2.1. Versiones	9
4.7.3. Estructura de FreePascal	10
Notas del capítulo	13

Índice de figuras

4.1. Evolución de Portable Pascal.	3
4.2. Evolución de compiladores para Pascal.	4
4.3. Arquitectura de GPC.	9
4.4. Arquitectura de FPC.	11

Índice de cuadros

4.1. Versiones de Pascal-P.	2
4.2. Relación entre la Biblioteca Estándar de Pascal y el Cálculo Matemático.	7
4.3. Comparativa entre compiladores de Pascal.	8

Capítulo 4

Compiladores del Lenguaje Pascal

4.1. Pascal User's Group (PUG)

*“This is the first issue of a newsletter sent to users and other interested parties about the programming language PASCAL. Its purpose is to keep the PASCAL community informed about the efforts of individuals to implement PASCAL on different computers and to report extensions made o the language. It will be published at infrequent intervals due to the limited manpower...”*¹

George H. Richmond. 1974 (Newsletter #1)

4.1.1. Historia

La Revista PUG fue publicada entre Enero de 1974 y Noviembre de 1983. Durante su actividad resultó un importante soporte para la evolución del Lenguaje Pascal.

Entre otros aspectos, se trató la estandarización de Pascal, la generación de compiladores base como P4 y aspectos de la evolución que sufría la computación en la década de los setenta.

Resalta el hecho, de que es sus últimas publicaciones se nombra un nuevo Lenguaje en desarrollo, ADA.

UCSD Pascal fue duramente criticado dado que era un proyecto que se ajustaba a las bases “de facto” de PUG.

El estándar propio de Pascal (propuesto por Tony Addyman) resultó ser la ultima gran disputa entre PUG y los institutos ANSI e ISO.

4.2. Pascal-P (The Portable Pascal Compiler)

El equipo de Wirth en la Universidad de Zurich creo dos familias de compiladores:

- i. CDC 6000: Código nativo para las propias máquinas CDC 6000. Se trataban de compiladores de una pasada que traducían el código fuente a código máquina directamente. Usaban o “Full Pascal”.
- ii. Pascal-P: Enfocado a la portabilidad y compatibilidad. Su idea era crear compilador/intérprete capaz de generar código intermedio para que después, sobre una arquitectura en concreto, se generase el ejecutable.

4.2.1. Historia Pascal CDC 6000

Fue implementado en Scallop (Lenguaje propio de las máquina CDC) entre los años 1970 y 1971. Hubo también un intento de desarrollar el mismo compilador en Lenguaje Fortran pero debido al uso de que hacía el Lenguaje Pascal de estructuras recursivas, hizo imposible la tarea.

En el año 1972 Wirth y su equipo trabajan en una revisión del Lenguaje Pascal, un subconjunto del original ya que, se trabajaba intensamente en la idea de un compilador independiente de una arquitectura en concreto. La primera versión de Pascal Portable, P1 usaba la máquina “Stack” o pseudo-machine. Se trato de un prototipo que convivió con las versiones de CDC 6000.

4.2.2. Historia Pascal-P

Versión	Origen	Año	Hito
Pascal P1	Zurich	1973	Concepto de portabilidad entre arquitecturas.
Pascal P2	Zurich	1974	Implementación de “Full Pascal” y primer Compilador.
Pascal P3	Zurich	1976	Paso previo entre P2 y P4.
Pascal P4	Zurich	1976	“Estándar de facto” y base para UCSD Pascal.
Pascal P5	San Jose	2009	Compatible con ISO 7185 .
Pascal P6	Comunidad	En Desarrollo	Implementación de ISO 10206 y Pascaline.

Cuadro 4.1: Versiones de Pascal-P.

- I. Pascal P2: Publicado en 1974, se trataba de una versión real del lenguaje. Fue acompañada de una revisión integra de “Full Pascal” o también llamado Pascal 1971. Sobre P2 se derivaron importantes compiladores como: UCDS Pascal a la vez que sirvió de prototipo para Borland Turbo Pascal.
- II. Pascal P3 y P4: Está versión es la más importante de toda la familia dado que, aún hoy día sobrevive y es matriz para desarrollar nuevos compiladores. Data de 1976 aunque ha sido plenamente adaptada al estándar ISO Pascal 7185:1990. Decir que fue acompañada de una versión P3 que trataba de ser un intermediario entre P2 y P4, fue una implementación hipotética debido a que P4 se convirtió en el “estándar de facto”.
- III. Pascal P5: Dado los problemas de memoria sobre los que se desarrolló la versiones previas, en 2009 se propuso una revisión de Pascal-P4 que tiene como objetivo principal (sigue en desarrollo) mejorar el rendimiento y ser plenamente compatible con ISO 7185.
- IV. Revisiones:
 - i. Pascal P6: Pretendía implementar la versión extendida de Pascal. Finalmente se decidió desarrollar como una versión de Pascaline que añade a la ISO mecanismo de la Programación Paralela y Distribuida.
 - ii. Pascal P7: Hipotética versión exclusiva para Pascal Extendido. No se ha llegado a codificar debido a que dicha ISO es parte del proyecto P6.

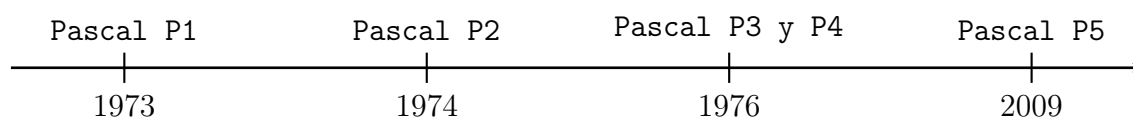


Figura 4.1: Evolución de Portable Pascal.

4.3. UCSD Pascal

UCSD Pascal o también, University of California, San Diego Pascal se trata de una revisión de Pascal-P2 que supuso una importante evolución conceptual en los lenguaje de programación.

Su característica más destacada era que uso instrucciones **p-code** con el propósito de ser multiplataforma, idea que era realidad a finales de los años setenta y que es parte hoy día de importantes Lenguajes de Programación como Java.

4.3.1. Historia

La idea original es de Kenneth Bowles quien en 1974 se percato de la gran cantidad de arquitecturas que existían y la incompatibilidad entre ellas. La síntesis de su idea era crear un dialecto de Pascal-P2 para que generase en la compilación el p-code que era fácilmente portable entre distintas arquitecturas con base en **p-code Operating Systems**.

La disputa surge por IBM y su política de instalaciones base, en concreto se ofrecía UCSD p-System, PC-DOS y CP/M-86 pero el rendimiento era muy distinto para los modelos de Hardware de la época. Por ello se ideó UCSD Pascal basado en una arquitectura p-code. El sistema se pasó a llamar The UCSD Pascal p-Machine que ya en sus orígenes era compatible para distintas máquinas.

Su estructura de compilación puso de base la necesidad de unidades de código (UNITS) y el uso de cadenas (STRING).

UCSD Pascal ha tenido cuatro versiones:

- I. Versión 1.0: Primer Software Base que fue distribuido junto al código fuente. Esta versión fue mejorada por los propios usuarios y derivaron en gran cantidad de mejoras.
- II. Versión 2.0: Revisión que trajo consigo compatibilidad con numerosas arquitecturas como: Apple II, DEC PDP-11, Zilog, MOS 6502, Motorola 68000 y primeros IBM-PC.
- III. Versión 3.0: Escrita desde Western Digital era parte de Pascal MicroEngine.
- IV. Versión 4.0: Desarrollada por SofTech, era una versión comercial orientada a la industria del desarrollo. Finalmente y tras ser improductiva paso a manos de Pecan? Systems que a su vez estaba formada por entusiastas de p-System.

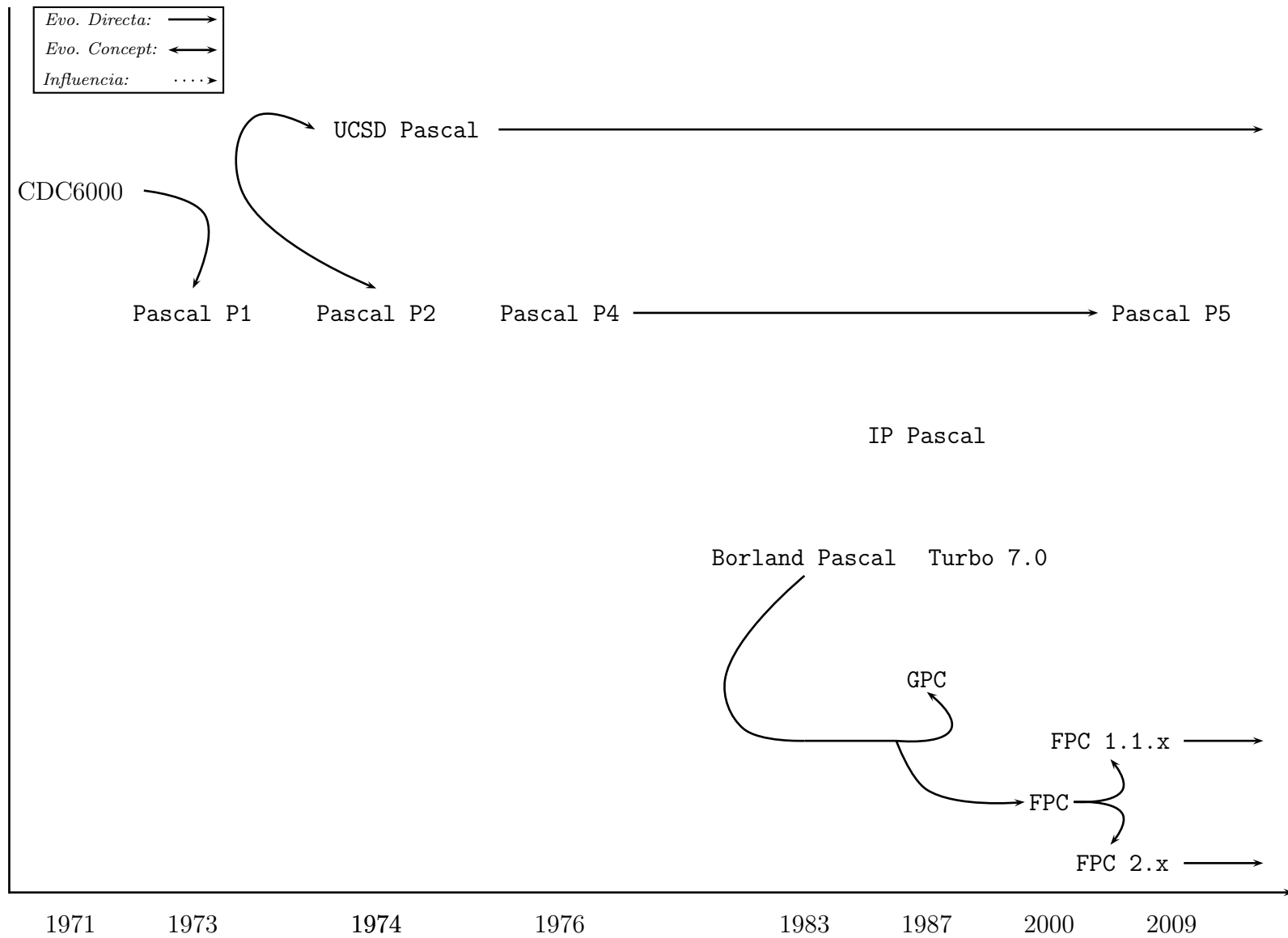


Figura 4.2: Evolución de compiladores para Pascal.

4.4. Pascaline

El dialecto Pascaline (Calculadora de Pascal) implementa el estándar ISO 7185 además de incorporar importantes funcionalidades como: Conceptos de Programación Orientada a Objetos, Arrays dinámicos o Monitores.

4.4.1. IP Pascal

IP Pascal Se trata un conjuntos de programas: IDE (Entorno de Desarrollo), compilador y codificador.

A lo largo de su desarrollo ha sufrido importantes mejoras y usando distintas plataformas de ejecución como:

- i. Z80: La implementación original (1980) fue escrita en Lenguaje Ensamblador de la propia máquina Z80. En 1985, IP Pascal fue completamente transcrito al propio Lenguaje Pascal. Ya en 1987, sufrió un importante cambio estructural tomando como base el Lenguaje C, dado que se estaba preparando la versión i386.
- ii. i386: Evolucionó a lo largo de las distintas versiones, donde originalmente se usaba código intermedio para IBM-PC. En 1994 se añadieron las funcionalidades de **Extended ISO 7185 Pascal**.
- iii. GNU/Linux: Creada en el año 2000 por la empresa Red Hat para su uso exclusivo a través de línea de comandos. En su diseño se utilizó un sistema GNU (glibc) y Syscalls para núcleo Linux.

4.5. Borland Pascal

Turbo Pascal se trata de un paquete Software compuesto por un compilador y un entorno de desarrollo (IDE).

El compilador fue desarrollado por la empresa Borland y que, gozó de gran popularidad a principios de los noventa del siglo XX dado su precio y compatibilidad con MS-DOS.

4.5.1. Historia

El desarrollo de Turbo Pascal estuvo liderado por Philippe Kahn, quien sentó su bases de su diseño. Entre sus hitos, destacan el de integra el proceso de: edición, compilación y enlazado. Por aquella época, era el propio programador y de manera explícita el que realizaba estas tareas. El concepto de “Kit de Desarrollo” unido a su precio de venta fueron los factores determinantes en su popularidad a los largo de mediados de los años ochenta y años noventa del siglo XX.

Su primera versión se basó Blue Label Pascal². Turbo Pascal se lanzó al mercado como Compas Pascal para CP/M con otras arquitecturas desarrolladas como: Apple II, máquinas DEC, CP/M-86 y MS-DOS. Su precio de mercado era de 49.99 USD. Hablamos del año 1983, por aquel entonces el Software y en particular los compiladores tenían precios mucho más elevados. Otro hito importante es que poco después fue lanzado la computadora personal IBM PC, dónde el propio compilador ofrecía resultados sorprendentes de rendimiento para estas máquinas tan limitadas.

Las versiones 2 y 3 del compilador ofrecieron cambios discretos, haciendo énfasis en la gestión de la memoria.

Por contra la versión 4 lanzada en 1987, fue prácticamente reescrita desde cero. Las versiones de 5 a 7 siguieron en la línea de añadir nuevos complementos al Software.

4.5.2. Valores internos para datos numéricos simples

I. Tipo Entero:

- i. SHORTINT: $[-128, 127]$ (1 Byte)
- ii. INTEGER: $[-32768, 32767]$ (2 Bytes)
- iii. LONGINT: $[-2147483648, 2147483647]$ (4 Bytes)
- iv. BYTE: $[0, 255]$ (1 Bytes)
- v. WORD: $[0, 65535]$ (2 Bytes)

II. Tipo Real:

- i. REAL: $[2.9 \cdot 10^{-39}, 1.7 \cdot 10^{38}]$ (de 11 a 12 dígitos representables, 6 Bytes)
- ii. SINGLE: $[1.5 \cdot 10^{-45}, 3.4 \cdot 10^{38}]$ (de 7 a 8 dígitos representables, 4 Bytes)
- iii. DOUBLE: $[5.0 \cdot 10^{-324}, 1.7 \cdot 10^{308}]$ (de 15 a 16 dígitos representables, 8 Bytes)
- iv. EXTENDED: $[1.9 \cdot 10^{-4851}, 1.1 \cdot 10^{4932}]$ (de 19 a 20 dígitos representables, 10 Bytes)
- v. COMP: $[-9.2 \cdot 10^{18}, 9.2 \cdot 10^{18}]$ (de 19 a 20 dígitos representables, 8 Bytes)

4.5.3. Biblioteca estándar

I. Procedimientos Estándar de Turbo Pascal (Descritas en el apartado ??):

- i. PROCEDURE APPEND(*var F:Text*); → Abre el archivo determinado como parámetro (*var F:Text*) para escribir a partir del final del fichero.
- ii. PROCEDURE DISPOSE(*var P:Pointer*); → Se encarga de liberar la memoria asignada al puntero (*var P:Pointer*).
- iii. PROCEDURE NEW(*var P:Pointer*); → Reserva memoria para el puntero (*var P:Pointer*).
- iv. PROCEDURE READ(*var F: tipodeFichero; {lista de variables}*); → *idem*.
- v. PROCEDURE READ([*var F: tipodeFichero;*] *{lista de variables}*); → *idem*.
- vi. PROCEDURE READLN([*var F: ficherodeTexto;*] *{lista de variables}*); → *idem* para la utilización de parámetros con el procedimiento anterior, con la salvedad de que se lee toda una línea del fichero, con el consiguiente avance del puntero de lectura.
- vii. PROCEDURE RESET(*var F: tipodeFichero*); → *idem*.
- viii. PROCEDURE REWRITE(*var F: tipodeFichero*); → *idem*.
- ix. PROCEDURE WRITE(*var F: tipodeFichero; {lista de variables}*); → *idem*
- x. PROCEDURE WRITE([*var F: tipodeFichero;*] *{lista de variables}*); → *idem*

- xi. PROCEDURE WRITELN(*[var F: fichero de Texto;] {lista de variables}*); $\rightarrow idem$ para la utilización de parámetros con el procedimiento anterior, con la salvedad de que se escribe toda una línea del fichero, con la consiguiente marca del puntero de escritura.

Función	Simbología
FUNCTION ABS	$ x $
FUNCTION ARCTAN	$arctg(x)$
FUNCTION COS	$cos(x)$
FUNCTION EXP	e^x
FUNCTION LN	$Ln x$
FUNCTION SIN	$sen(x)$
FUNCTION SQR	x^2
FUNCTION SQRT	\sqrt{x}
FUNCTION TRUNC	$TRUNC(a, b) = a$

Cuadro 4.2: Relación entre la Biblioteca Estándar de Pascal y el Cálculo Matemático.

II. Funciones Estándar de Turbo Pascal:

- i. FUNCTION ABS(*x: tipo*): *tipo*; $\rightarrow idem$
- ii. FUNCTION ARCTAN(*x: REAL*): *REAL*; $\rightarrow idem$
- iii. FUNCTION CHR(*x: BYTE*): *CHAR*; $\rightarrow idem$
- iv. FUNCTION COS(*x: REAL*): *REAL*; $\rightarrow idem$
- v. FUNCTION EOF(*var F: tipo de Fichero*): *BOOLEAN*; $\rightarrow idem$
- vi. FUNCTION EOLN(*var F: tipo de Fichero*): *BOOLEAN*; $\rightarrow idem$
- vii. FUNCTION EXP(*x: REAL*): *REAL*; $\rightarrow idem$
- viii. FUNCTION LN(*x: REAL*): *REAL*; $\rightarrow idem$
- ix. FUNCTION ODD(*x: LONGINT*): *BOOLEAN*; $\rightarrow idem$
- x. FUNCTION ORD(*x: tipo Ordinal*): *LONGINT*; $\rightarrow idem$
- xi. FUNCTION PRED(*x: tipo Ordinal*): *tipo Ordinal*; $\rightarrow idem$
- xii. FUNCTION ROUND(*x: REAL*): *LONGINT*; $\rightarrow idem$
- xiii. FUNCTION SIN(*x: REAL*): *REAL*; $\rightarrow idem$
- xiv. FUNCTION SQR(*x: tipo*): *tipo*; $\rightarrow idem$
- xv. FUNCTION SQRT(*x: REAL*): *REAL*; $\rightarrow idem$
- xvi. FUNCTION SUCC(*x: tipo Ordinal*): *tipo Ordinal*; $\rightarrow idem$
- xvii. FUNCTION TRUNC(*x: REAL*): *LONGINT*; $\rightarrow idem$

Nombre	Instrucciones	Código Binario	Distribuido	IDE	Multiplataforma
CDC 6000	Full	Si	No	No	No
Pascal P1	1971	Ø	Ø	Ø	Ø
Pascal P2	1971	No	No	No	Si
Pascal P3	1971	Ø	Ø	Ø	Ø
Pascal P4	1971	No	No	No	Si
Pascal P5	ISO 7185	No	No	No	Si
Pascal P6	ISO 10206	Ø	Ø	Ø	Ø
UCSD Pascal	1971	No	No	No	Si
Pascaline	ISO 7185	Si	Si	Ø	Si
IP Pascal	ISO 7185	Si	Si	No	Si
Borland Pascal	Borland	Si	No	Si	Si
GPC	ISO 7185	No	No	No	Si
FPC	FPC	Si	Si	Si	Si

Cuadro 4.3: Comparativa entre compiladores de Pascal.

4.6. GNU Pascal Compiler (GPC)

4.6.1. ¿Qué es GPC?

GPC (GNU Pascal Compiler) se trata de un compilador del lenguaje de programación Pascal perteneciente a la familia de compiladores de GNU GCC. Su primeras versiones datan de 1987. El compilador GPC se presenta como un autómata portable, rápido y flexible.

Es compatible con la ISO 7185 de Pascal e incorpora soporte para la ISO 10206 de Pascal Extendido.

Durante el año 2010 el grupo de desarrolladores debatió en torno al hipotético futuro de compilador y su integración con el “Front-End”³ de GCC. Finalmente en Julio de 2013 se congeló su desarrollo.

4.6.2. Estructura de GPC

Su portabilidad se basa en su estructura motor, es decir, en las herramientas con las que se ha creado.

- i. Flex: Determina utilizando expresiones regulares la pertenencia o no de a palabra al alfabeto (ver Apartado ??) Σ^4 .
- ii. Bison: Trata la sintaxis en base a las especificaciones BNF⁵.
- iii. Interfaz GAS: GNU Assembler (más conocido como GAS) se trata del ensamblador oficial del proyecto GNU. Es el “Back-End” para GCC. Se distribuye en el metapaquete Software Binutils. Actualmente tiene la licencia GPL v.3.0.

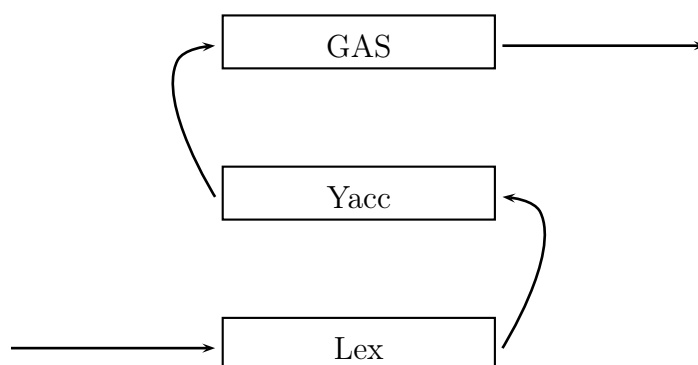


Figura 4.3: Arquitectura de GPC.

4.7. FreePascal

4.7.1. ¿Qué es FreePascal?

Inicialmente se conocía como FPC (Florian Paul Klämpf) acrónimo del propio autor.

Actualmente FPC está compuesto por: el propio Compilador, un conjunto de Bibliotecas y un IDE (Lazarus).

4.7.2. Historia

Su desarrollo comienza tras el anuncio de Borland en relación al abandono de su su familia de compiladores Borland Pascal (su sucesor natural sería Delphi).

Las primeras versiones fueron escritas por Florian Paul Klämpf en el propio dialecto de Borland Pascal. Del mismo modo, sus primeros ejecutables fueron para MS-DOS de 16 bits aunque, dos años después soportaba distintas arquitecturas de 32 bits.

4.7.2.1. Versiones

- I. Rama 0.x: La versión de 32 bits fue distribuida a través de Internet. Se hizo compatible con GNU/Linux y OS/2. 0.88.5 se convirtió en el primer producto estable de PFC. A pesar de esto, la mejora posterior (0.99.8) se hizo plenamente compatible con Win32 y añadía gran parte de las Bibliotecas de de Delphi.
- II. Rama 1.x: La primera versión estable de esta rama fue lanzada en Julio del año 2000 a la que siguió la 1.0.10 de Julio de 2003 donde se insistió en la corrección de errores. La misma se hizo compatible con procesadores de 68K, hecho que dejó palpables las notables deficiencias en el diseño del propio compilador.

Por ello se tomó la decisión de la reescritura del mismo con el claro objetivo de la limpieza del código y la idea de ser plenamente compatible con distintas plataformas.

Entre Noviembre de 2003 y principios de 2003 el nuevo diseño fue tomando forma y finalmente fue presentada como FPC 1.9.0 compatible para:

- i. Por Arquitectura: x86 y amd64 , Porwer-PC, ARM y Sparc v.8 y v.9.
 - ii. Por Sistema Operativo Base: Win2K y MS-DOS, GNU/Linux, Mac OS X, FreeBSD, OS/2.
- III. Rama 2.2.x: La motivación de está versión venía dada por que Lazarus necesitaba soporte pleno para: Win64, Windows CE y Mac OS X en x86. La primera versión estable se publicó en Septiembre de 2007 (2.2.2). Además se incorporó en lo sucesivo soporte para Active X/COM y OLE que lo convertía en un producto maduro para plataformas Win2k.
- IV. Rama 2.4.x: La versión 2.4 de FPC trajo consigo importantes cambios en el diseño del compilador. De nuevo la portabilidad fue el aspecto más relevante y en el que mayor esfuerzo realizó el equipo de desarrolladores- Las nuevas plataformas soportadas fueron:
- i. Mac PowerPC 64 y x86-amd64.
 - ii. iPhone.
 - iii. ARM.
- Se añadió también soporte para Delphi y se reescribió “Unit System”.
- V. Rama 2.6 y 2.7: El lanzamiento en Enero de 2014 de PFC 2.6 aportó el soporte pleno del compilador en Mac OS X. La revisión 2.7 (actualmente en desarrollo) incorpora gran cantidad de cambio en el núcleo del compilador:
- i. Soporte para ISO 7185 y capacidad de compilar código de P4.
 - ii. Soporte para Delphi (aspectos avanzados de POO).
 - iii. Soporte para las arquitecturas y SSOO: MIPS, NetBSD, OpenBSD, AmigaOS (m68k) y JVM (algunas primitivas).

4.7.3. Estructura de FreePascal

El compilador FPC se divide en tres partes bien diferenciadas (siguiendo el esquema propio de un compilador):

- I. Analizador Léxico (*Scanner/Tokenizer*): El escáner (LEX) analiza el flujo de entrada de datos y prepara la lista de tokens que a su vez será utilizado por el *Paser*. Es el estado donde se analizan las directivas del Preprocesador. A su vez se divide en las siguientes unidades:
 - i. Flujo de Entrada (*Input Stream*): Se encarga de normalizar el método de entrada (I/O) al fichero llavero `file.pas`
 - ii. Preprocesador: El escáner resuelve todas las directivas del preprocesador en el código fuente del programa. Se encarga de transformar dichas operaciones en sentencias de Pascal.

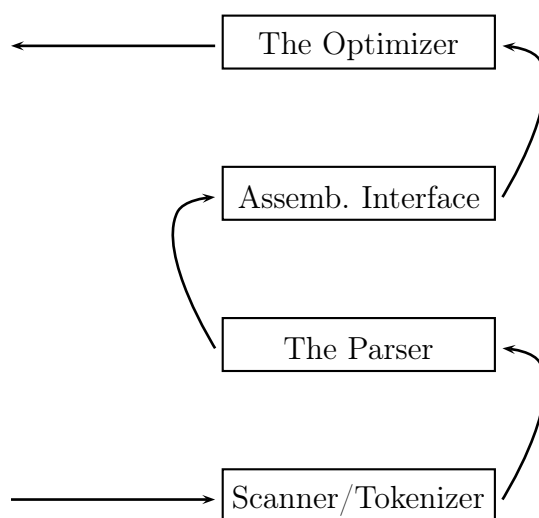


Figura 4.4: Arquitectura de FPC.

- II. Árbol Sintáctico (*The Parse Tree*): El árbol es la base del compilador. Tras el desglose de las sentencias y bloques, el código es traducido a una estructura de datos tipo Árbol *The Parse Tree*.
- III. Analizador Sintáctico (*Parser*): La tarea del *Parser* (Yacc) es la de analizar el flujo de tokens generado por el (*Scanner*) y comprobar que tiene un orden lógico es decir, que se ajustan a la sintaxis del lenguaje.

El mismo *Parser* utiliza una tabla de símbolos y genera un árbol de nodos para la interfaz de *Assembler*.

- IV. Generador de Código (*The Code Generator*): La interfaz *The Code Generator* es la encargada de generar el Output para el Lenguaje Ensamblador y posteriormente el enlace con las bibliotecas del SSOO.

En la versión 1.0 de FPC establecía código intermedio por cada nodo tras el primer análisis. A su vez se asociaba con las rutinas en código ensamblador tras la “segunda pasada” y finalmente generaba las instrucciones en Ensamblador.

Notas del capítulo

¹“Este es el primer número de un boletín enviado a los usuarios y otras partes interesadas sobre el lenguaje de programación PASCAL. Su propósito es mantener a la comunidad informada sobre PASCAL los esfuerzos de las personas para poner en práctica PASCAL en equipos diferentes y que informe extensiones hechas o el idioma. Se publicará a intervalos poco frecuentes debido a la mano de obra limitada...”

²Desarrollado por NasSys.

³El problema y motivo de la discusión era por la reimplementación del código intermedio, en este caso C++ a C.

⁴Para la versión: <http://www.gnu-pascal.de/alpha/gpc-20060325.tar.bz2> el fichero `pascal-lex.l`

⁵Para la versión: <http://www.gnu-pascal.de/alpha/gpc-20060325.tar.bz2> el fichero `parse.y`

Índice alfabético

A

ADA, 1

B

Back-End, 8

Borland, 5

Borland Turbo Pascal, 2

C

Calculadora de Pascal, 5

CDC 6000, 1

E

Entorno de Desarrollo, 5

F

Front-End, 8

Full Pascal, 1

Funciones Estándar, 7

G

GNU Assembler, 8

GNU Pascal Compiler, 8

GNU/Linux, 5

I

i386, 5

IBM-PC, 3

IP Pascal, 5

ISO 10206, 8

ISO 7185, 8

ISO Pascal 7185:1990, 2

L

Lenguaje Fortran, 2

Lenguaje Pascal, 2

M

Motorola 68000, 3

MS-DOS, 5

P

p-code, 3

p-code Operating Systems, 3

Pascal 1971, 1

Pascal P2, 2

Pascal P3 y P4, 2

Pascal P5, 2

Pascal P6, 2

Pascal P7, 2

Pascal-P, 1

Pascaline, 5

PC-DOS, 3

Procedimientos Estándar, 6

pseudo-machine, 2

R

Revista PUG, 1

T

Tipo Entero, 6

Tipo Real, 6

Turbo Pascal, 5

U

UCSD Pascal, 1

Universidad de Zurich, 1

University of California, San Diego Pascal, 3

Z

Z80, 5