

Pascal: preparazione di alcuni compilatori comuni

Pascal-to-C	953
Librerie e compilazione	953
Uso di Pascal-to-C	955
GPC	956
Free-Pascal	957

Sono disponibili diversi compilatori per il linguaggio Pascal; in questo capitolo ne vengono descritti alcuni.

Pascal-to-C

Pascal-to-C,¹ è una sorta di compilatore che permette di convertire un sorgente Pascal in un sorgente C. I problemi che possono sorgere da questo tipo di conversione sono nella definizione precisa del tipo di dialetto Pascal e del tipo di dialetto C. Utilizzando Pascal-to-C con GNU/Linux, non si dovrebbero avere difficoltà con il compilatore C. Quello che resta da sistemare è la definizione del dialetto Pascal che si vuole usare, dal momento che ne esistono di diversi, che alle volte sono incompatibili.

Questi dettagli possono essere controllati e configurati; quello che conta è esserne consapevoli e approfondire l'uso di Pascal-to-C attraverso lo studio della documentazione originale, quando se ne presenta la necessità, ovvero quando si intende programmare seriamente attraverso questo strumento.

Il nome di Pascal-to-C è indicato dal suo autore come P2c. Tuttavia, P2C è anche il nome di un altro compilatore analogo, realizzato per sistemi speciali: <http://www.geocities.com/SiliconValley/Network/3656/rocket/>. In questo secondo caso, oltre alla particolarità del compilatore stesso, c'è da considerare il fatto che non si tratta di software libero.

Librerie e compilazione

Il codice C generato da Pascal-to-C contiene sempre l'inclusione del file `'p2c/p2c.h'`, che poi, a sua volta, provvede a includere il solito `'stdio.h'`.

Il *link* del file generato dalla compilazione del sorgente C che si ottiene, deve essere fatto includendo la libreria `'libp2c.a'`, cosa che si traduce generalmente nell'uso dell'opzione `'-lp2c'`.

In pratica, le fasi necessarie a ottenere un programma eseguibile si riassumono nei due comandi seguenti.

```
p2c sorgente_pascal
```

```
cc -lp2c sorgente_c
```

L'eseguibile che si ottiene, richiede la presenza della libreria dinamica `'libp2c.so'`.

Il funzionamento predefinito di `'p2c'` può essere configurato attraverso una serie di file di configurazione:

1. `'/usr/lib/p2c/p2csrc', '$P2CRC'`
2. `'~/p2csrc'`
3. `'~/p2csrc'`

Il primo file dell'elenco è quello usato per definire la configurazione generale. Eventualmente, si può usare la variabile di ambiente `'P2CRC'`, contenente il percorso assoluto per raggiungere un file analogo, sostituendosi in tal modo a quello generale.

Dopo il file di configurazione generale, viene cercato il file 'p2crc' nella directory personale dell'utente, oppure, in sua mancanza, il file '.p2crc'. Questo file serve a definire una personalizzazione della configurazione di 'p2c'.

Le direttive di questo file di configurazione sono rappresentate da assegnamenti, espressi in una delle due forme seguenti.

```
nome = valore
```

```
nome valore
```

I commenti si rappresentano come di consueto facendoli precedere dal simbolo '#', dove le righe vuote o bianche vengono semplicemente ignorate.

Il file di configurazione che accompagna Pascal-to-C, cioè '/usr/lib/p2c/p2crc', contiene l'elenco completo di tutte le direttive utilizzabili, tutte impostate nel modo più conveniente per l'uso normale e tutte debitamente commentate in modo da sapere come può essere modificato ogni valore.

L'esempio seguente definisce l'utilizzo di un sorgente TURBO Pascal:

```
Language Turbo
```

Le direttive di configurazione possono anche essere incorporate all'interno dello stesso sorgente Pascal, permettendo così una definizione dinamica, riferita a porzioni di codice. Per farlo, si utilizza una forma speciale dei commenti Pascal (le parentesi graffe fanno parte della direttiva).

```
{ nome=valore }
```

In tal caso, come si può vedere, il simbolo '=' è obbligatorio e l'uso di spazi bianchi è generalmente inammissibile. È possibile l'utilizzo di commenti anche all'interno di direttive espresse in questo modo. Per farlo, occorre usare la sequenza '##'.

La configurazione dinamica all'interno del sorgente, permette di utilizzare anche altre modalità di assegnamento e di eliminazione automatica delle definizioni alla fine del sorgente. Per approfondirle, conviene consultare la documentazione originale, cosa che si riduce in pratica alla lettura di p2c(1).

Segue la descrizione di alcuni esempi.

```
{ Language=Turbo }
```

Definisce l'utilizzo di un sorgente TURBO Pascal.

```
{ Language=Turbo ## utilizza una codifica TURBO Pascal }
```

Definisce l'utilizzo di un sorgente TURBO Pascal e vi aggiunge un commento interno.

Le direttive della configurazione di Pascal-to-C sono numerose; anche se l'impostazione predefinita si adatta alle situazioni più comuni, potrebbe essere conveniente modificarne alcune, già le prime volte che si utilizza Pascal-to-C.

```
AnsiC [ 0 | 1 ]
```

Permette di definire il tipo di dialetto C da utilizzare. Se si attiva la modalità, utilizzando il valore uno, si fa in modo di generare codice C ANSI; se invece non si inserisce, o si utilizza il valore zero, si ottiene un codice compatibile con il C K&R originale.

Come accennato, se non si definisce diversamente, si ottiene un codice C tradizionale, mentre potrebbe essere desiderabile di generare codice C ANSI.

```
Language [ HP | HP-UX | Turbo | UCSD | VAX | Oregon | Berk | Modula ]
```

Permette di definire il dialetto Pascal utilizzato come sorgente per la conversione. Le varie parole chiave usate per distinguere i dialetti hanno il valore seguente:

'HP'	È la codifica usata in modo predefinito e si riferisce precisamente al Pascal HP, compatibile con il Pascal dello standard ISO.
'HP-UX'	È il Pascal HP del sistema HP-UX, praticamente identico al Pascal HP normale.
'Turbo'	TURBO Pascal 5.0, quello usato con il Dos. La differenza rispetto al tipo HP è minima, tanto che generalmente non è necessario richiedere esplicitamente questo tipo di codifica, quando si usano sorgenti TURBO Pascal.
'UCSD'	UCSD Pascal. Si tratta di un dialetto molto simile al TURBO Pascal.
'MPW'	Macintosh Programmer's Workshop Pascal 2.0, senza le estensioni Object Pascal.
'VAX'	VAX/VMS Pascal 3.5. Non tutte le funzionalità sono disponibili.
'Oregon'	Oregon Software Pascal/2.
'Berk'	Berkeley Pascal con le estensioni Sun.
'Modula'	Modula-2. Basato sul libro <i>Programming in Modula-2</i> di Wirth, terza edizione. La conversione in C a partire da questo formato non è ancora completa.

```
ShortOpt [ 0 | 1 ]
```

Permette di definire il modo con cui devono essere valutate le espressioni logiche: uno abilita il «cortocircuito» attraverso cui si valutano effettivamente solo le condizioni strettamente necessarie a determinare il risultato finale; zero lo disabilita, in modo che tutte le condizioni vengano valutate in ogni caso.

Uso di Pascal-to-C

La conversione del sorgente Pascal in linguaggio C avviene per mezzo del programma 'p2c', configurato come descritto nelle sezioni precedenti.

'p2c' è effettivamente un compilatore, il cui risultato è un programma C. Questo significa che genera da solo la segnalazione di errori di sintassi nel sorgente Pascal e, alla fine, il sorgente C che si ottiene dovrebbe essere corretto (dal punto di vista del C).

```
p2c [ opzioni ] [ file ]
```

'p2c' legge il file indicato come argomento, oppure lo standard input in sua mancanza. In base alle opzioni e alla configurazione definita, genera da quel file una trasformazione in linguaggio C.

Il nome del file generato si ottiene togliendo l'eventuale estensione precedente e aggiungendo '.c'.

Tabella u118.5. Alcune opzioni di 'p2c'.

Opzione	Descrizione
-o file	Definisce esplicitamente il nome del file del sorgente C da generare.
-c file_di_configurazione	Definisce il nome di un file di configurazione da utilizzare al posto di quelli standard.
-a	Genera codice C ANSI. Questa opzione permette di sostituirsi agevolmente alla configurazione standard secondo cui il sorgente generato dovrebbe essere di tipo tradizionale (K&R).
-l { HP HP-UX ↵ ↵ Turbo UCSD VAX ↵ ↵ Oregon Berk Modula }	Permette di definire il tipo di Pascal nel sorgente. Le caratteristiche abbinata alle varie parole chiave sono state descritte in occasione della descrizione dei file di configurazione.

Segue la descrizione di alcuni esempi.

```
• $ p2c mio_programma.pas [Invio]
```

Genera il file 'mio_programma.c' convertendo il contenuto di

```
'mio_programma.pas'.
```

```
* $ p2c -a mio_programma.pas [Invio]
```

Come nell'esempio precedente, ma generare un programma C secondo lo standard ANSI.

```
* $ p2c -a -o mio.c mio_programma.pas [Invio]
```

Come nell'esempio precedente, ma il file generato è 'mio.c'.

Segue la descrizione di un esempio di compilazione:

```
{
  CiaoMondo.pas
  Programma elementare di visualizzazione di un messaggio
  attraverso lo standard output.
}

program CiaoMondo;

begin
  Writeln('Ciao Mondo!');
end.
```

Se il file si chiama 'CiaoMondo.pas', si può trasformare in C con il comando seguente:

```
$ p2c CiaoMondo.pas [Invio]
```

```
CiaoMondo
Translation completed
```

Si ottiene così il file 'CiaoMondo.c', mostrato di seguito.

```
/* Output from p2c, the Pascal-to-C translator */
/* From input file "CiaoMondo.pas" */

/*
  CiaoMondo.pas
  Programma elementare di visualizzazione di un messaggio
  attraverso lo standard output.
*/

#include <p2c/p2c.h>

main(argc, argv)
int argc;
Char *argv[];
{
  PASCAL_MAIN(argc, argv);
  printf("Ciao Mondo!\n");
  exit(EXIT_SUCCESS);
}

/* End. */
```

Questo file può essere compilato a sua volta.

```
$ cc -lp2c -o CiaoMondo CiaoMondo.c [Invio]
```

Se tutto funziona correttamente, si ottiene il file 'CiaoMondo' eseguibile.

```
$ ./CiaoMondo [Invio]
```

```
Ciao Mondo!
```

Se si desidera generare un sorgente C ANSI, si può usare l'opzione '-a' di 'p2c'. Nel caso dell'esempio, il corpo del programma C sarebbe stato il seguente:

```
main(int argc, Char *argv[])
{
  PASCAL_MAIN(argc, argv);
  printf("Ciao Mondo!\n");
  exit(EXIT_SUCCESS);
}
```

GPC

GPC,² ovvero il Pascal GNU, è un compilatore Pascal che fa parte di GCC. Il suo utilizzo immediato è molto semplice:

```
gpc file_pascal
```

In questo modo, l'eseguibile 'gpc' compila il sorgente indicato come argomento e genera il file 'a.out', che poi può essere avviato. In

alternativa, per specificare il nome del file eseguibile da generare con la compilazione, si usa l'opzione '-o':

```
gpc -o file_eseguibile file_pascal
```

Per esempio, per compilare il programma seguente, contenuto nel file 'CiaoMondo.pas', si può procedere con il comando mostrato subito dopo:

```
{
  CiaoMondo.pas
  Programma elementare di visualizzazione di un messaggio
  attraverso lo standard output.
}

program CiaoMondo;

begin
  Writeln('Ciao Mondo!');
end.
```

```
$ gpc -o CiaoMondo CiaoMondo.pas [Invio]
```

Dalla compilazione si genera il file eseguibile 'CiaoMondo':

```
$ ./CiaoMondo [Invio]
```

```
Ciao Mondo!
```

GPC è un compilatore Pascal molto sofisticato, accompagnato da una documentazione molto dettagliata. Qui si elencano soltanto le opzioni di uso più comune, per consentirne l'uso a scopo didattico, con gli esempi che vengono descritti nei capitoli successivi di questa parte dedicata al Pascal.

Tabella u18.13. Alcune opzioni per l'uso del compilatore GPC.

Opzione	Descrizione
-o file	Specifica il nome del file eseguibile o del file oggetto da generare.
-Wall	Richiede di mostrare tutte le informazioni diagnostiche (<i>warning</i>) che possono essere utili a migliorare il sorgente Pascal.
-Werror	Fa in modo che ogni piccolo difetto del sorgente sia considerato alla stregua di un errore che impedisce il buon esito della compilazione.
-static	Se possibile, fa in modo di incorporare le librerie nell'eseguibile che viene generato.
--short-circuit	Abilita o disabilita il «cortocircuito», ovvero la valutazione effettiva delle sole condizioni strettamente necessarie a determinare il risultato finale.
--no-short-circuit	

Free-Pascal

Free-Pascal³ è un altro compilatore Pascal. La compilazione è svolta dall'eseguibile 'fpc' che si avvale anche di un file di configurazione, '/etc/fpc.cfg', che può contenere le stesse opzioni della riga di comando.

In condizioni normali, quando si installa Free-Pascal da un pacchetto già pronto per la propria distribuzione GNU, il file di configurazione dovrebbe essere già adatto alle caratteristiche del proprio sistema, senza richiedere altri interventi. Così, di solito è sufficiente compilare un programma in questo modo:

```
fpc file_pascal
```

Di solito, se il nome del file sorgente Pascal ha un'estensione del tipo '.pas', si ottiene un file eseguibile con la stessa radice e senza estensione. Per esempio, compilando il file seguente, denominato 'CiaoMondo.pas', si ottiene il file eseguibile 'ciao mondo' nella stessa directory:

```
{
  CiaoMondo.pas
  Programma elementare di visualizzazione di un messaggio
  attraverso lo standard output.
}

program CiaoMondo;

begin
  Writeln('Ciao Mondo!');
end.
```

\$ **fpc** CiaoMondo.pas [Invio]

Si osservi che il nome dell'eseguibile che si ottiene contiene solo lettere minuscole:

\$ **./ciao mondo** [Invio]

```
Ciao Mondo!
```

¹ **Pascal-to-C** GNU GPL

² **GPC** GNU GPL

³ **Free-Pascal** GNU LGPL