Glibc 2 HOWTO

Eric Green, ejg3@cornell.edu

v1.6, 22 June 1998

Questo HOWTO, glibc 2, riguarda l'installazione e l'uso della versione 2 della libreria GNU C (libc 6) sui sistemi Linux. Traduzione a cura di Andrea Girotto (andrea at andreagirotto.it) e revisione a cura di Marcello Seri (marcello.seri at email.it).

Indice

1 Introduzione.

1.1 Note su glibc 2.

Glibc 2 è l'ultima versione della libreria GNU C. Attualmente può essere utilizzata senza modifiche su sistemi GNU Hurd e Linux i386, m68k, e sistemi alpha. Nella versione 2.1 saranno supportate anche le macchine Linux PowerPC, MIPS, Sparc, Sparc 64 ed Arm. In futuro il supporto verrà esteso ad altre architetture e sistemi operativi.

Su Linux, glibc 2 è usata come libc con versione principale 6, successiva di Linux libc 5. Con questo si intende comunicare agli sviluppatori di sostituire libc 5. A partire dalla versione 2.0.6, glibc è considerata qualità di produzione. La versione 2.1 (attesa in un prossimo futuro) sarà pronta per un diffuso uso con l'aggiunta di molti adattamenti e caratteristiche.

Ci sono tre aggiunte opzionali disponibili per glibc 2:

Crypt

Il pacchetto UFC-crypt. È separato a causa di restrizioni per l'esportazione.

LinuxThreads

Implementazione dell'interfaccia Posix 1003.1c pthread

Locale data

Contiene i dati necessari per costruire i file dati locali da usare per le caratteristiche di internazionalizzazione di glibc.

Si raccomandano caldamente i pacchetti crypt e LinuxThreads... non usandoli c'è il rischio di essere incompatibili con le librerie di altri sistemi. (Se si desidera non usarli, è necessario aggiungere l'opzione –disable-sanity-checks durante la configurazione.)

1.2 Note su questo documento.

Questo HOWTO descrive come installare glibc 2 su un sistema Linux esistente. È pensato per utenti di sistemi Intel che usino libc 5; in ogni caso, utenti di altri sistemi e librerie diverse (come glibc 1), dovrebbero essere in grado di utilizzare queste informazioni sostituendo i nomi di file corretti e i nomi delle architetture dove necessario.

La copia più recente (in inglese) di questo HOWTO può essere trovata come parte di *Linux Documentation Project* `http://www.tldp.org`

oppure da `http://www.imaxx.net/~thrytis/glibc/Glibc2-HOWTO.html` . Per la rispettiva versione italiana si veda: *Italian Linux Documentation Project* `http://it.tldp.org` .

1.3 Modifiche recenti a questo documento.

Differenze tra le versioni 1.6 ed 1.5:

- Corrette le istruzioni di installazione del package binario di glibc

Differenze tra le versioni 1.5 ed 1.4:

- Indice aggiunto da Ed Bailey.
- Cambiato l'indirizzo email dell'autore.

2 Scegliere il proprio metodo di installazione.

È possibile installare glibc in diversi modi. Le librerie possono essere installate come test, usando quelle già esistenti come predefinite ma lasciando la possibilità di provare le nuove utilizzando opzioni diverse durante la compilazione dei programmi. Questo tipo di installazione rende facile una futura rimozione di glibc (tuttavia qualsiasi programma collegato con glibc non sarà più in grado di funzionare se le librerie dovessero essere eliminate). Usare glibc come libreria di test richiede la compilazione dai sorgenti. Non ci sono distribuzioni in formato binario per questo tipo di configurazione. L'installazione è descritta in 4 (Installare come libreria di test).

L'altro tipo di installazione descritto in questo documento è per l'uso di glibc come libreria primaria. Tutti i nuovi programmi compilati sul sistema useranno glibc, tuttavia sarà sempre possibile collegare programmi alla versione precedente usando differenti opzioni di compilazione. È possibile sia installare le librerie precompilate, sia compilare le librerie personalmente. Se si desidera cambiare opzioni di ottimizzazione o di configurazione, o usare aggiunte che non siano distribuite come pacchetto binario, è necessario avere la distribuzione in codice sorgente per la compilazione. Questa procedura di installazione è descritta in 5 (Installare come libreria C primaria).

Frodo Looijaard descrive un diverso modo di installare glibc. Questo metodo implica l'installazione di glibc come libreria secondaria e la configurazione come compilatore incrociato per l'uso di glibc. La procedura è più complicata che l'installazione della libreria di test descritta in questo documento, ma consente una compilazione più semplice quando si collega glibc. Il metodo è descritto nel suo documento *Installing glibc-2 on Linux* `http://huizen.dds.nl/~frodol/glibc/` .

Se si sta utilizzando la distribuzione Debian 1.3 ma non si desidera aggiornare ad una versione instabile di Debian per usare glibc, il documento *Debian libc5 to libc6 Mini-HOWTO* `http://www.gate.net/~storm/FAQ/libc5-libc6-Mini-HOWTO.html` descrive come usare i pacchetti Debian per il proprio sistema.

Se si installa glibc 2 su un sistema importante, si potrebbe usare l'installazione di test. Anche se non ci sono errori, alcuni programmi potrebbero aver bisogno di qualche modifica prima che possano essere compilati, come conseguenza di alterazioni dei prototipi alle funzioni e dei tipi.

3 Ottenere la libreria.

La libreria glibc 2 consiste nel pacchetto glibc e tre altri opzionali, LinuxThreads, Locale, e Crypt. Il sorgente può essere trovato presso

- `ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.6.tar.gz`
- `ftp://prep.ai.mit.edu/pub/gnu/glibc-linuxthreads-2.0.6.tar.gz`
- `ftp://prep.ai.mit.edu/pub/gnu/glibc-localedata-2.0.6.tar.gz`
- `ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.6.tar.gz`

Saranno necessari 150 MB di spazio su disco per la compilazione completa e l'installazione. La libreria base occupa circa 50 MB.

Pacchetti binari per 2.0.6 non sono disponibili. Le versioni binarie dei pacchetti 2.0.4 per i386 e m68k, e le versioni 2.0.1 per alpha sono disponibili:

- Intel x86:
  - `ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.4.bin.i386.tar.gz`
  - `ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.4.bin.i386.tar.gz`
- Alpha:
  - `ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.1.bin.alpha-linux.tar.gz`
  - `ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.1.bin.alpha-linux.tar.gz`
- m68k:
  - `ftp://prep.ai.mit.edu/pub/gnu/glibc-2.0.4-m68k-linux.bin.tar.gz`
  - `ftp://prep.ai.mit.edu/pub/gnu/glibc-crypt-2.0.4-m68k-linux.bin.tar.gz`

Ci sono restrizioni di esportazione per le aggiunte crittografiche. Utenti non Statunitensi possono recuperare una versione presso `ftp://ftp.ifi.uio.no/pub/gnu` .

Se si utilizza una distribuzione Red Hat, è possibile avere gli rpm per glibc 2 da `ftp://ftp.redhat.com/pub/redhat/` . Glibc 2 è la libreria C primaria per la nuova distribuzione Red Hat 5.0.

Se si utilizza una distribuzione Debian, è possibile avere i pacchetti per glibc 2 da `ftp://ftp.debian.org/debian/dists/unstable/main/` . I file sono nominati libc6. Glibc 2 ora è parte del pacchetto base per la versione hamm di Debian, e sarà la libc primaria quando Debian 2.0 sarà rilasciata.

4 Installare come libreria di test.

Questa sezione descrive come installare glibc 2 come libreria di test. Qualsiasi cosa compilata sarà collegata alle librerie esistenti a meno che non si usino alcuni parametri extra per il collegamento alle nuove librerie. Pare che i percorsi siano compilati in pochi file, quindi probabilmente è necessario installare la libreria dalla versione sorgente.

4.1 Compilare ed installare.

4.1.1 Prerequisiti.

- Circa 150 MB di spazio libero
- GNU make 3.75
- gcc >= 2.7.2 (meglio 2.7.2.1)
- binutils 2.8.1 (per alpha è necessario uno snapshot)
- bash 2.0
- autoconf 2.12 (se si cambia configure.in)
- texinfo 3.11

Su un i586@133 con 64 MB di RAM, sono necessarie circa 3 ore per compilare le librerie complete con le aggiunte. Su un i686@200 carico, è necessaria circa mezz'ora.

4.1.2 Estrarre i sorgenti.

È necessario estrarre i sorgenti dagli archivi così che si possano compilare. Il modo migliore per farlo è:

```
tar xzf glibc-2.0.6.tar.gz
cd glibc-2.0.6
tar xzf ../glibc-linuxthreads-2.0.6.tar.gz
tar xzf ../glibc-crypt-2.0.6.tar.gz
tar xzf ../glibc-localedata-2.0.6.tar.gz
```

Questi comandi estrarranno le directory linuxthreads, crypt e localedata in glibc-2.0.6 dove configure potrà trovare le rispettive aggiunte.

4.1.3 Configurazione.

Nella directory glibc-2.0.6, se ne crea un'altra chiamata compile, e la si rende corrente. Tutto il lavoro sarà fatto in questa directory, cosa che semplificherà la pulizia. (Gli sviluppatori non sono ancora stati in grado di realizzare un 'make clean' perfetto.)

```
mkdir compile
cd compile
```

Si esegue `../configure`. Per usare i pacchetti aggiuntivi, è necessario specificarli con –enable-add-ons, ad esempio –enable-add-ons=linuxthreads,crypt,localedata. È anche necessario scegliere una directory di installazione. /usr/i486-linuxglibc2 è una buona scelta. Il comando per configure quindi sarà:

```
../configure --enable-add-ons=linuxthreads,crypt,localedata --prefix=/usr/i486-linuxglibc2
```

4.1.4 Compilare ed installare.

Per compilare e verificare, si esegue:

```
make
make check
```

Se il comando 'make check' ha successo, installare la libreria:

```
make install
```

4.2 Aggiornare il loader dinamico.

1. Creare un collegamento dal nuovo `ld.so` a `/lib/ld-linux.so.2`:

   ```
   ln -s /usr/i486-linuxglibc2/lib/ld-linux.so.2 /lib/ld-linux.so.2
   ```

   Questa è la sola libreria la cui locazione è prefissata una volta che un programma sia stato collegato e l'uso di un riferimento in `/lib` faciliterà l'aggiornamento a glibc come libreria C primaria, una volta che la versione stabile sia stata rilasciata.

2. Modificare `/etc/ld.so.conf`. È necessario aggiungere un percorso alla directory lib dove le nuove librerie risiedono, che sarà <`prefix`>`/lib`, come `/usr/i486-linuxglibc2/lib` per la scelta precedente. Dopo aver corretto `/etc/ld.so.conf`, si esegue

   ```
   ldconfig -v
   ```

4.3 Configurare gcc.

L'ultimo passo dell'installazione è aggiornare `/usr/lib/gcc-lib` in modo che gcc sappia come usare le nuove librerie. Per prima cosa è necessario duplicare la configurazione esistente. Per verificare la configurazione corrente, si usa l'opzione -v:

```
% gcc -v
Reading specs from /usr/lib/gcc-lib/i486-unknown-linux/2.7.2.2/specs
gcc version 2.7.2.2
```

In questo caso, i486-unknown-linux è il sistema e 2.7.2.2 è la versione. È necessario copiare `/usr/lib/gcc-lib/`<`system`> nella nuova directory di test per il sistema:

```
cd /usr/lib/gcc-lib/
cp -r i486-unknown-linux i486-linuxglibc2
```

Si rende corrente la nuova directory test di sistema e la directory di versione

```
cd /usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2
```

e si corregge il file `specs` presente. In questo file, si cambia `/lib/ld-linux.so.1` a `/lib/ld-linux.so.2`. Inoltre è necessario anche eliminare tutte le espressioni `%{...:-lgmon}` nel file, dal momento che glibc non usa la libreria gmon per il profiling. Un esempio di file specs può essere trovato nella sezione 10 (Esempio file specs).

4.4 Aggiornare i collegamenti per header file.

È necessario creare collegamenti nella nuova directory di link alle altre directory include:

```
cd /usr/i486-linuxglibc2/include
ln -s /usr/src/linux/include/linux
ln -s /usr/src/linux/include/asm
ln -s /usr/X11R6/include/X11
```

Alcune librerie come ncurses necessitano di avere i propri file posti in questa directory. Si dovrà copiare o collegare i file in questione da `/usr/include`. (Probabilmente sarà necessario ricompilarne alcune con glibc2 per farle funzionare. In questi casi, basta compilare ed installare il pacchetto in `/usr/i486-linuxglibc2`.)

4.5 Verificare la propria installazione.

Per verificare l'installazione, si crei il seguente programma in un file glibc.c:

```
#include <stdio.h>

main()
{
    printf("hello world!\n");
}
```

e si compili con l'opzione "-b <directory base> -nostdinc -I<directory installazione>/include -I/usr/lib/gcc-lib/<nuova directory di sistema>/<gcc version>/include":

```
% gcc -b i486-linuxglibc2 -nostdinc -I/usr/i486-linuxglibc2/include
-I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include glibc.c -o glibc
```

Si usi ldd per verificare che il programma è stato collegato con glibc2 e non la vecchia libc:

```
% ldd glibc
libc.so.6 => /usr/i486-linuxglibc2/lib/libc-2.0.6.so (0x4000d000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Se compila, i collegamenti sono verificati e l'output del programma quando eseguito è "hello world!" l'installazione ha avuto successo.

5 Installazione come libreria C primaria

Questa sezione descrive l'installazione di glibc 2 come libreria primaria. Qualsiasi nuovo programma compilato sarà collegato con questa libreria, a meno che non si usino particolari opzioni per qualche altra versione.

Se si dispone di una distribuzione Redhat o Debian, dopo aver prelevato i file rpm o deb appropriati, consultare le istruzioni per l'installazione. In questo caso si può saltare la sezione che segue.

5.1 Compilare la libreria dai sorgenti.

Questa sezione spiega come compilare glibc 2 e le aggiunte a partire dai sorgenti. È necessario compilare la libreria se si desidera cambiare opzioni di ottimizzazione o configurazione o usare un pacchetto di cui non si dispongano i binari.

5.1.1 Prerequisiti.

- Circa 150 MB di spazio libero su disco
- GNU make 3.75
- gcc >= 2.7.2 (meglio 2.7.2.1)
- binutils 2.8.1 (per alpha è necessario a ?)
- bash 2.0
- autoconf 2.12 (se si modifica configure.in)
- texinfo 3.11

Su un i586@133 con 64 MB di RAM, sono necessarie circa 3 ore per compilare completamente librerie e aggiunte. Su un i686@200 carico, è necessaria circa mezzora.

5.1.2 Estrarre i sorgenti.

È necessario estrarre i sorgenti dagli archivi in modo da poterli compilare. Il modo migliore per farlo è:

```
tar xzf glibc-2.0.6.tar.gz
cd glibc-2.0.6
tar xzf ../glibc-linuxthreads-2.0.6.tar.gz
tar xzf ../glibc-crypt-2.0.6.tar.gz
tar xzf ../glibc-localedata-2.0.6.tar.gz
```

Questa sequenza di comandi dispone le directory linuxthreads, crypt, e localedata in glibc-2.0.6 dove è possibile configurare queste aggiunte.

5.1.3 Configurazione.

Si crei una directory di nome compile in `glibc-2.0.6`, e la si renda directory corrente. Tutto il lavoro sarà fatto in questa directory, cosa che semplificherà la pulizia. (Gli sviluppatori non sono ancora stati in grado di realizzare un 'make clean' perfetto.)

```
mkdir compile
cd compile
```

Si esegua `../configure`. Per usare i pacchetti aggiuntivi, è necessario specificarli con –enable-add-ons, come in –enable-add-ons=linuxthreads,crypt,localedata. Probabilmente si desidererà anche specificare i percorsi dove debbano essere installati. Per adeguarsi alle distribuzioni standard di Linux, si specifichi –prefix=/usr. (Quando un prefisso di `/usr` è specificato su un sistema linux, configure sa come sistemare i percorsi rimanenti per disporre libc.so e le altre librerie importanti in `/lib`.) La linea completa di configurazione dovrebbe essere:

```
../configure --enable-add-ons=linuxthreads,crypt,localedata --prefix=/usr
```

5.1.4 Compilazione.

Per compilare e verificare, si esegua:

```
make
make check
```

5.2 Prepararsi per l'installazione.

Ora è necessario spostare alcuni file per prepararsi per la nuova libreria, a seconda che si stia installando dai sorgenti o dai binari. Ogni nuovo programma compilato sarà collegato alla glibc, ma programmi vecchi che non sono collegati staticamente dipenderanno ancora dal libc 5, per questo non è possibile semplicemente sovrascrivere la vecchia versione.

1. Creare una nuova directory per contenere i vecchi file:

   ```
   mkdir -p /usr/i486-linuxlibc5/lib
   ```

2. I vecchi file di header devono essere rimossi da `/usr/include`:

   ```
   mv /usr/include /usr/i486-linuxlibc5/include
   ```

3. Creare una nuova directory per i file include ed assegnare i collegamenti per includere altre directory:

   ```
   mkdir /usr/include

   ln -s /usr/src/linux/include/linux /usr/include/linux
   ln -s /usr/src/linux/include/asm /usr/include/asm
   ln -s /usr/X11R6/include/X11 /usr/include/X11
   ln -s /usr/lib/g++-include /usr/include/g++
   ```

   I collegamenti potrebbero essere diversi a seconda della distribuzione di cui si dispone. Ad esempio Slackware pone gli header g++ in `/usr/local/g++-include`, mentre Debian in `/usr/include/g++`, e collega `/usr/lib/g++-include` a `/usr/include/g++`. Nell'ultimo caso, probabilmente si desidererà spostare la directory originale di nuovo in `/usr/include`.

4. Ripristinare ogni file di header o collegamento extra. Alcune librerie non standard come ncurses pongono i propri file in `/usr/include` oppure creano un link alla loro directory di include in `/usr/include`. È necessario ripristinare questi file e collegamenti al fine di poter usare correttamente le librerie extra.

5. Aggiungere la nuova directory della libreria (ad esempio `/usr/i486-linuxlibc5/lib`) *all'inizio* del proprio file `/etc/ld.so.conf`. È necessario avere ld.so 1.8.8 (o più recente) per evitare alcuni strani messaggi una volta che glibc sia installata.

6. Spostare/copiare tutte le librerie C vecchie nella nuova directory.

```
mv /usr/lib/libbsd.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libc.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libgmon.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libm.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libmcheck.a /usr/i486-linuxlibc5/lib
mv /usr/lib/libc.so /usr/i486-linuxlibc5/lib
mv /usr/lib/libm.so /usr/i486-linuxlibc5/lib
cp /lib/libm.so.5.* /usr/i486-linuxlibc5/lib
cp /lib/libc.so.5.* /usr/i486-linuxlibc5/lib
```

   `libm.so.5` e `libc.so.5` dovrebbero essere copiate e non spostate se `/usr` è una partizione separata da `/`, in quanto sono necessarie per i programmi usati per far partire linux e devono risiedere nella partizione di root.

7. Spostare i file `/usr/lib/*.o` nella nuova directory.

```
mv /usr/lib/crt1.o /usr/i486-linuxlibc5/lib
mv /usr/lib/crti.o /usr/i486-linuxlibc5/lib
mv /usr/lib/crtn.o /usr/i486-linuxlibc5/lib
mv /usr/lib/gcrt1.o /usr/i486-linuxlibc5/lib
```

8. Aggiornare la propria cache dopo lo spostamento delle librerie

```
ldconfig -v
```

5.3 Installare dal pacchetto binario.

Se si installa glibc da binari precompilati, per prima cosa si vorrà verificare cosa ci sia nel package prima di installare i binari:

```
tar -tzvvf glibc-2.0.bin.i386.tar.gz
tar -tzvvf glibc-crypt-2.0.bin.i386.tar.gz
```

Se si sta installando glibc da codice binario precompilato, si deve:

```
cd /
gzip -dc glibc-2.0.bin.i386.tar.gz | tar tvvf -
gzip -dc glibc-crypt-2.0.bin.i386.tar.gz | tar tvvf -
ldconfig -v
```

Con differenti architettura o versione, sostituire i nomi di file adeguati.

La versione di glibc più recente non è generalmente disponibile come package binario, e si raccomanda caldamente che sia eseguita la versione più recente per evitare errori. Se non è possibile compilare da sé la libreria, è sempre possibile prelevarla da una delle distribuzioni basate su glibc (ad esempio RedHat) ed installare questa.

5.4 Installare dai sorgenti.

Per installare la libreria dai sorgenti, si esegua:

```
make install
ldconfig -v
```

5.5 Aggiornare i file specs di gcc.

Il passo finale dell'installazione (sia nel caso di codice binario che sorgente) è di aggiornare il file `specs` di gcc, in modo da poter collegare correttamente i propri programmi. Per determinare quale file specs sia quello usato da gcc, si usi:

```
% gcc -v
reading specs from /usr/lib/gcc-lib/i486-unknown-linux/2.7.2.2/specs
gcc version 2.7.2.2
```

In questo caso i486-unknown-linux è il sistema, e 2.7.2.2 la versione. È necessario copiare il file `/usr/lib/gcc-lib/<system>` nella vecchia directory di sistema

```
cd /usr/lib/gcc-lib/
cp -r i486-unknown-linux i486-linuxlibc5
```

Andare nella directory originale e quella di versione

```
cd /usr/lib/gcc-lib/i486-unknown-linux/2.7.2.2
```

per modificare il file `specs` presente in questa directory. Nel file, si cambi `/lib/ld-linux.so.1` in `/lib/ld-linux.so.2`. Inoltre è necessario rimuovere tutte le espressioni `%{...:-lgmon}` presenti nel file, dal momento che glibc non usa la libreria gmon per il profiling. Un esempio di file specs può essere trovato nella sezione 10 (Esempio di file specs).

5.6 Controllare la propria installazione.

Per verificare l'installazione, si crei il seguente programma in un file glibc.c:

```
#include <stdio.h>

main()
{
    printf("hello world!\n");
}
```

e si compili il programma.

```
% gcc glibc.c -o glibc
```

Usando ldd è possibile verificare che il programma è stato collegato con glibc2 e non la vecchia libc:

```
% ldd glibc
libc.so.6 => /lib/libc.so.6 (0x4000e000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Se il programma si compila e genera "hello world!" quando eseguito, l'installazione è corretta.

6 Compilare con la libreria non primaria libc.

Ci sono situazioni in cui si desidera usare una libreria alternativa con cui compilare i propri programmi. Questa sezione spiega come raggiungere questo scopo, usando le directory ed i nomi di installazione degli esempi delle precedenti due sezioni. Si ricordi di correggere i nomi secondo la propria configurazione.

6.1 Avviso per l'uso di libc non primarie.

Prima di compilare un programma che sarà usato nella fase di boot del sistema, si ricordi che se il programma è collegato dinamicamente ed è usato prima che le partizioni non root siano montate, tutte le librerie collegate dovranno essere presenti nella partizione di root. Seguendo il percorso di installazione della precedente sezione su come installare glibc come libreria primaria, la vecchia libreria è lasciata in `/lib`, che sarà nella propria partizione di root. Questo significa che tutti i programmi saranno in grado di funzionare durante la fase di boot. Tuttavia se `/usr` è su una diversa partizione e si installa glibc come libreria di test in `/usr/i486-linuxglibc2`, qualsiasi nuovo programma compilato con glibc non funzionerà fino a quando non si sia montata `/usr`.

6.2 Compilare programma con una glibc di test

Per compilare un programma con una installazione di prova di glibc, è necessario impostare i percorsi di include a quelli di glibc. Specificando "-nostdinc" si negheranno i normali percorsi, e "-I/usr/i486-linuxglibc2/include" farà puntare agli include di glibc. Sarà anche necessario specificare gli include di gcc che si trovano in `/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include` (assumendo che si sia installata la libreria di test in i486-linuxglibc2 con versione di gcc 2.7.2.2).

Per collegare un programma con una libreria di prova glibc, è necessario specificare la configurazione di gcc. Questo si ottiene usando l'opzione "-b i486-linuxglibc2".

Per la maggior parte dei programmi, si possono specificare queste nuove opzioni aggiungendole alle opzioni di makefile `$CFLAGS` e `$LDFLAGS`:

```
CFLAGS = -nostdinc -I/usr/i486-linuxglibc2/include -I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include -b i486-linuxglibc2
LDFLAGS = -b i486-linuxglibc2
```

Se si sta usando uno script di configurazione, si definiscano le variabili d'ambiente `$CFLAGS` e `$LDFLAGS` (usando env/setenv per csh/tcsh, o set/export per sh/bash/etc) prima di eseguire configure. I makefile generati in questo modo dovrebbero avere `$CFLAGS` e `$LDFLAGS` corretti. Non tutti i file di configurazione fanno riferimento a queste variabili, quindi è necessario controllare l'esecuzione di configure e correggere manualmente i makefile se necessario.

Se i programmi che si compilano eseguono solo gcc (e non direttamente cpp o binutils), si può usare lo script che segue per evitare di scrivere tutte le opzioni:

```
#!/bin/bash
/usr/bin/gcc -b i486-linuxglibc2 -nostdinc \
             -I/usr/i486-linuxglibc2/include \
             -I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include "$@"
```

Si può usare questo script invece di "gcc" per la compilazione.

6.3 Compilare programmi con libc 5 quando glibc è la libreria primaria.

Per compilare un programma con le vecchie librerie avendo installato glibc come libreria primaria, è necessario impostare i percorsi di inclusione ai vecchi include. Specificando "-nostdinc" si nega all'accesso al percorso normale, e "-I/usr/i486-linuxlibc5/include" farà puntare agli include di libc. Inoltre è necessario specificare "-I/usr/lib/gcc-lib/i486-linuxlibc5/2.7.2.2/include" per accedere a specifici include gcc. Si ricordi di correggere questi percorsi sulla base dei nomi che si sono dati alle proprie directory e versione di gcc.

Per collegare un programma con la propria vecchia libc, è necessario specificare la configurazione, ed è fatto con: "-b i486-linuxlibc5".

Per la maggior parte dei programmi, si può specificare le nuove opzioni aggiungendole alle opzioni di makefile `$CFLAGS` e `$LDFLAGS`:

```
CFLAGS = -nostdinc -I/usr/i486-linuxlibc5/include -I/usr/lib/gcc-lib/i486-linuxlibc5/2.7.2.2/include -b i486-linuxlibc5
LDFLAGS = -b i486-linuxlibc5
```

Se si sta usando uno script di configurazione, si definiscano le variabili d'ambiente `$CFLAGS` e `$LDFLAGS` (usando env/setenv per csh/tcsh, o set/export per sh/bash/etc) prima di eseguire configure. I makefile generati in questo modo dovrebbero avere `$CFLAGS` e `$LDFLAGS` corretti. Non tutti i file di configurazione fanno riferimento a queste variabili, quindi è necessario controllare l'esecuzione di configure e correggere manualmente i makefile se necessario.

Se i programmi che si compilano eseguono solo gcc (e non direttamente cpp o binutils), si può usare lo script che segue per evitare di scrivere tutte le opzioni:

```
#!/bin/bash
/usr/bin/gcc -b i486-linuxlibc5 -nostdinc \
             -I/usr/i486-linuxlibc5/include \
             -I/usr/lib/gcc-lib/i486-linuxlibc5/2.7.2.2/include "$@"
```

Si può usare questo script invece di "gcc" per la compilazione.

7 Compilare programmi C++.

Libg++ usa alcune parti della libreria matematica, come pure il collegamento a libm. Da momento che la libg++ esistente è stata compilata con la propria precedente libreria, è necessario ricompilare libg++ con glibc per avere una copia binaria. L'ultima versione sorgente per libg++ insieme con la versione binaria compilata con glibc (per x86) può essere trovata presso:

`ftp://ftp.yggdrasil.com/private/hjl/` .

7.1 Installare libg++ per glibc configurata per test.

Avendo installato glibc come libreria di test, è necessario installare i file nella directory dove risiede glibc (che è `/usr/i486-linuxglibc2` per gli esempi delle sezioni precedenti). Installando da un pacchetto in formato binario (cosa che l'autore consiglia in quanto non è mai riuscito a compilare con successo libg++ in questo modo), è necessario estrarre i file in una directory temporanea e spostare tutti i file da `usr/lib/` in `<directory di installazione>/lib/`, da `usr/include/` in `<directory di installazione>/include/` (ricordarsi prima di rimuovere il link `include/g++`!), ed infine da `usr/bin/` in `<directory di installazione>/bin/`.

7.2 Installare libg++ per glibc configurata come primaria.

Avendo installato glibc come libreria primaria e volendo mantenere la possibilità di compilare programmi con la vecchia versione di libc, è necessario spostare - prima della nuova installazione - tutti i file nella directory di libc. Probabilmente il modo più semplice per farlo è di installare una copia di libg++ compilata per libc5 come nella sezione precedente, e poi installare normalmente glibc.

7.3 Compilare programmi C++ con libc non primaria.

Per compilare programmi C++ con una libreria non primaria libc, è necessario specificare la directory di include g++ che negli esempi precedenti è `/usr/i486-linuxglibc2/include/g++` per l'installazione di glibc come libreria di test e `/usr/i486-linuxlibc5/include/g++` per l'installazione di glibc come libreria primaria. Normalmente si raggiunge questo scopo impostando la variabile `$CXXFLAGS`:

```
CXXFLAGS = -nostdinc -I/usr/i486-linuxglibc2/include -I/usr/lib/gcc-lib/i486-linuxglibc2/2.7.2.2/include -I/usr/i486-linuxlibc5/include/g++ -b i486-linuxglibc2
```

8 Problemi.

Il Package glibc contiene una FAQ con informazioni aggiuntive da verificare se ci sono problemi. Una versione online è anche disponibile presso

`http://222.imaxx.net/~thrytis/glibc/glibc-FAQ.html` . Sotto ci sono alcuni suggerimenti per risolvere problemi che non sono coperti dall'FAQ o sono trattati qui in maggiore dettaglio.

8.1 I nomi di Host non vengono risolti

Glibc 2 usa un metodo differente da libc 5 nel cercare i nomi di host. Il codice (NSS) per lo switch del name server di glibc cerca per un file `/etc/nsswitch.conf`. Se i nomi di host non sono risolti usando un'applicazione glibc 3 ed il file `/etc/resolv.conf` è configurato correttamente, verificare che ci sia il file `/etc/nsswitch.conf`. Se non esiste, è possibile crearne uno contenente la linea:

```
hosts: files dns
```

Adesso si cercherà in `/etc/resolv.conf` per trovare i nameserver.

Si consulti alle pagine info della sezione di libc che descrive il file nsswitch.conf in maggiore dettaglio.

9 Segnalare bug.

Se si pensa che la libreria contenga un errore, per prima cosa si consulti il file FAQ. È possibile che altri abbiano avuto lo stesso problema e che ci sia una soluzione facile. Inoltre è necessario controllare la sezione

Strumenti consigliati per l'installazione della libreria GNU C nel file INSTALL dal momento che alcuni sono bug dei tool e non della libreria.

Una volta scoperto un errore è necessario sincerarsi che sia veramente un errore. Un buon modo è confrontare il comportamento della GNU con altre librerie C. Se è lo stesso, probabilmente non si tratta di un bug (ma non è detto), altrimenti una delle librerie quasi certamente è errata.

Passo successivo, consultare il database di bug presso:

http://www-gnats.gnu.org:8080/cgi-bin/wwwgnats.pl `http://www-gnats.gnu.org:8080/cgi-bin/wwwgnats.pl` . Verificare che il problema non sia già stato segnalato. Sarebbe anche il caso di controllare il file `BUGS` distribuito con libc per vedere l'elenco dei bug conosciuti.

Scovato un errore di cui si è certi, si cerchi di restringerlo al caso di test più compatto possibile. Nel caso di una libreria C, dovrebbe essere possibile restringerlo ad una sola chiamata di libreria. Il compito non dovrebbe essere troppo difficile.

Il passo finale consiste nello scrivere un semplice caso di test per segnalare l'errore. Comunicando il problema, si spedisce il test, i risultati ottenuti, i risultati attesi, quale si ritiene essere il problema se si ha una qualche opinione), il tipo di sistema, la versione della libreria GNU C, il compilatore GNU CC, e la versione delle binutils GNU che si sta usando. Si aggiungano anche i file `config.status` e `config.make` che si ottengono eseguendo `configure`; si troveranno nella directory corrente al momento in cui si esegue `configure`.

Tutte le segnalazioni devono essere mandate usando lo script `glibcbug` fornito con la GNU libc all'indirizzo `bugs@gnu.org`

(l'indirizzo più vecchio `bugs@gnu.ai.mit.edu` è ancora operativo), oppure comunicate con l'interfaccia web di GNATS

`http://www-gnats.gnu.org:8080/cgi-bin/wwwgnats.pl` .

Suggerimenti e domande possono essere indirizzati all'indirizzo

`bugs-glibc@prep.ai.mit.edu` . Se non si legge il newsgroup gnu.bug.glibc, è possibile iscriversi alla lista chiedendo a

`bug-glibc-request@prep.ai.mit.edu` .

Per favore NON si comunichino bug all'indirizzo <bug-gcc@prep.ai.mit.edu>. Questa lista è per segnalare problemi per GNU CC. GNU CC e GNU C lib sono entità separate gestite da persone diverse.

10 Esempio di file specs.

Si include un esempio di file spec per glibc 2, usato da gcc per la compilazione ed il collegamento. Dovrebbe essere presente nella directory `/usr/lib/gcc-lib/<nuova directory sistema>/<versione gcc>`. Se si sta usando un sistema x86, probabilmente è possibile copiare esattamente questa sezione nel file.

```
*asm:
%{V} %{v:%{!V:-V}} %{Qy:} %{!Qn:-Qy} %{n} %{T} %{Ym,*} %{Yd,*} %{Wa,*:%*}

*asm_final:
%{pipe:-}

*cpp:
%{fPIC:-D__PIC__ -D__pic__} %{fpic:-D__PIC__ -D__pic__} %{!m386:-D__i486__} %{posix:-D_POSIX_SOURCE} %{pthread:-D_REENTRANT}

*cc1:
%{profile:-p}
```

```
*cc1plus:


*endfile:
%{!shared:crtend.o%s} %{shared:crtendS.o%s} crtn.o%s

*link:
-m elf_i386 %{shared:-shared}   %{!shared:     %{!ibcs:       %{!static:   %{rdynamic:-export-dynamic}      %{!dynamic-linker:-dynamic-linker /lib/ld-linux.so.2}}   %{static:-static}}}

*lib:
%{!shared: %{pthread:-lpthread}  %{profile:-lc_p} %{!profile: -lc}}

*libgcc:
-lgcc

*startfile:
%{!shared:      %{pg:gcrt1.o%s} %{!pg:%{p:gcrt1.o%s}                        %{!p:%{profile:gcrt1.o%s}                    %{!profile:crt1.o%s}}}}    crti.o%s %{!shared:crtbegin.o%s} %{shared:crtbeginS.o%s}

*switches_need_spaces:


*signed_char:
%{funsigned-char:-D__CHAR_UNSIGNED__}

*predefines:
-D__ELF__ -Dunix -Di386 -Dlinux -Asystem(unix) -Asystem(posix) -Acpu(i386) -Amachine(i386)

*cross_compile:
0

*multilib:
. ;
```

11 Miscellanea.

11.1 Altre informazioni.

11.1.1 Pagine web.

- *FSF's GNU C Library Home Page* http://www.gnu.org/software/libc/libc.html
- *Using GNU Libc 2 with Linux* http://www.imaxx.net/~thrytis/glibc/
- *Installing glibc-2 on Linux* http://huizen.dds.nl/~frodol/glibc/ .
- *Debian libc5 to libc6 Mini-HOWTO* http://www.gate.net/~storm/FAQ/libc5-libc6-Mini-HOWTO.html .

11.1.2 Newgroups.

- *comp.os.linux.development.system*

- *comp.os.linux.development.apps*
- *linux.dev.kernel*
- *gnu.bugs.glibc*

11.1.3 Mailing lists.

Lista Linux di discussione Glibc 2

Questa lista è intesa per le discussioni tra utenti Linux che hanno installato glibc 2, le nuove librerie GNU C. Argomenti riguardano problemi di compatibilità e questioni relative la compilazione del codice in ambiente Linux/glibc. Per iscriversi è necessario scrivere a *Majordomo@ricardo.ecn.wfu.edu* includendo come messaggio: subscribe glibc-linux <your email address>.

Gli archivi per questa mailing list possono essere trovati presso

`http://www.progressive-comp.com/Lists/?l=linux-glibc&r=1&w=2#linux-glibc`

11.2 Riconoscimenti

La maggior parte di queste informazioni è stata presa da

GNU Libc web page `http://www.gnu.org/software/libc/libc.html` , dall'annuncio della glibc 2 e relativi commenti di Ulrich Drepper's <drepper@gnu.ai.mit.edu>. Andreas Jaeger <aj@arthur.rhein-neckar.de> ha fornito parte della sezione riguardo la segnalazione di bug.

Le seguenti persone hanno fornito segnalazioni ed informazioni su questo documento:

- Allex <allex@ms2.accmail.com.tw>
- Mark Brown <M.A.Brown-4@sms.ed.ac.uk>
- Ulrich Drepper <drepper@gnu.ai.mit.edu>
- Scott K. Ellis <ellis@valueweb.net>
- Aron Griffis <agriffis@coat.com>
- Andreas Jaeger <aj@arthur.rhein-neckar.de>
- Frodo Looijaard <frodol@dds.nl>
- Ryan McGuire <rmcguire@freenet.columbus.oh.us>
- Shaya Potter <spotter@capaccess.org>
- Les Schaffer <godzilla@futuris.net>
- Andy Sewell <puck@pookhill.demon.co.uk>
- Gary Shea <shea@gtsdesign.com>
- Stephane <sr@adb.fr>
- Jan Vandenbos <jan@imaxx.net>

Traduzioni di questo documento sono fatte da:

- Cinese: Allex <allex@ms2.accmail.com.tw>

- Francese: Olivier Tharan <tharan@int-evry.fr>
- Italiano: Andrea Girotto <andrea@andreagirotto.it>
- Giapponese: Kazuyuki Okamoto <ikko-@pacific.rim.or.jp>

11.3 Segnalazioni.

A parte la scrittura di questo HOWTO, la manutenzione della pagina

glibc 2 for Linux `http://www.imaxx.net/~thrytis/glibc`

e l'uso sulla propria macchina, l'autore non ha nulla a che fare con il progetto glibc. Ogni segnalazione, correzione o proposta sarà ben accetta, scrivendo a *ejg3@cornell.edu* `mailto:ejg3@cornell.edu` .

11.4 Copyright.