

SuperH RISC engineファミリ用C/C++コンパイラパッケージ V.9 ご使用上のお願い

SuperH RISC engine ファミリ用C/C++コンパイラパッケージ V.9の使用上の注意事項2件を連絡します。

- 大量の関数呼び出しをインライン展開している場合の注意事項 (SHC-0081)
- リンカオプションdata_stuffおよびnooptimizeを選択している場合の注意事項 (LNK-0006)

1. 該当製品

SuperH RISC engine ファミリ用C/C++コンパイラパッケージ
V.9.00 Release 00 ~ V.9.03 Release 02

2. C/C++コンパイラ

2.1 大量の関数呼び出しをインライン展開している場合の注意事項 (SHC-0081)

問題 :

大量の関数呼び出しをインライン展開している場合に正しくないコードを生成する場合があります。

注意 : V.9.02 以前ではインライン展開処理時のメモリ使用量制限があるため、V.9.03以降と比較すると発生確率は低くなります。

発生例 :

以下をすべて満たすプログラムにおいて現象を確認しています。

使用コンパイラ : V.9.03 Release 02

ソース行数 : 約12,000行

関数の数 : 約600関数

インライン展開実施回数 : 約150回

発生条件 :

以下のいずれかを満たす場合に発生することがあります。

- (1) inline=数値 オプションを使用し、数値に1以上を指定している。
注意： speed オプションを指定している場合のデフォルトは、
inline=20 です。
nospeed、 size,optimize=0 またはoptimize=debug_only
オプションを指定している場合のデフォルトは、 noinline
(inline=0と同じ) です。
- (2) #pragma inline で宣言されている関数が存在する。

回避策：

以下をすべて実施してください。

- (1) -noinlineオプションを指定する。
- (2) #pragma inlineを使用しない。

なお、インライン展開される関数呼び出しを減らすことで回避できる場合があります。

その方法として以下の5つがあります。

- (1) inline=数値 オプションで指定する数値を小さくする。
ただし、 #pragma inline で宣言されている関数に対しては効果ありません。
- (2) file_inline オプションを使用しない。
- (3) #pragma inline で宣言する関数を減らす。
- (4) noscopeオプションを使用しない。
- (5) ファイル分割して、1ファイルの関数を減らすことでインライン展開される関数を減らす。

3. 最適化リンケージエディタ

3.1 リンカオプションdata_stuffおよびnooptimizeを選択している場合の注意事項 (LNK-0006)

問題：

リンカオプションdata_stuffおよびnooptimizeを選択している場合に複数シンボルが同じアドレスに割りつけられる場合があります。

発生条件：

以下のすべてを満たす場合に発生することがあります。

- (1) CPU種別がSuperHファミリである。
- (2) リンク時にリンカオプションdata_stuffを指定している。
- (3) リンク時にリンカオプションnooptimizeを指定している。
- (4) コンパイル時にコンパイラオプションcode=machinecodeを指定して生成された.objファイルを、最適化リンケージエディタに入力している。
- (5) (4)の.objファイルにはデータセクションが存在する。
- (6) (5)のデータセクションには、以下のサイズのシンボルがすべて含まれている。
 - 奇数バイト
 - 2バイト
 - 4の倍数バイト

- (7) (5)のデータセクション内に、奇数バイトのサイズのシンボルが奇数個連続で割り付けられている。
- (8) (7)の直後に2バイトのサイズのシンボルが割り付けられている。
- (9) (7)の奇数個連続したシンボルよりも小さいアドレス領域で、以下のすべてを満たしている。
 - (9-a) 2バイトのサイズのシンボルが割り付けられている。
 - (9-b) 4の倍数バイトのサイズのシンボルが割り付けられていない。
- (10) (9-a)の2バイトのサイズのシンボルより大きいアドレスに4の倍数バイトのサイズのシンボルを割り付けている。
- (11) (4)の.objファイルの他に複数の.objファイルを最適化リンケージエディタに入力している。
- (12) (11)の.objファイルには(4)と同名のデータセクションがある。

発生例:

ソース

```
-----data.c-----
//発生条件(5)
char char1; //発生条件(6)
char char2;
char char3;
char char4;
short short1; //発生条件(6)および(9-a)
char char5; //発生条件(7)および(9-b)
short short2; //発生条件(8)
char char6;
char char7;
char char8;
long long1; //発生条件(6) および(10)
```

```
-----dummy.c-----
//発生条件(12)
long dummy;
```

コマンド

```
//発生条件(1),(2),(3),(4) および(11)
shc dummy.c data.c
optlnk dummy.obj data.obj -data_stuff -nooptimize
```

実行結果: リンケージリストファイル

;シンボル_char8および_long1がアドレス0x00000010に割り付いている

```
.....  
FILE=data.obj  
00000008 0000000f 8
```

```
.....  
_char7  
0000000f 1 data ,g 0  
_char8  
00000010 1 data ,g 0  
FILE=data.obj  
00000010 00000013 4  
_long1  
00000010 4 data ,g 0  
-----
```

回避策:

以下のいずれかの方法で回避してください。

- (1) リンク時にリンカオプションdata_stuff指定しない。
- (2) 発生条件(4)に該当する.objファイルを、C/C++ソースファイルから以下の手順で生成する。
コンパイル時にコンパイラオプションcode=asmcodeを指定し、アセンブラソースを生成する。
そのアセンブラソースに対しアセンブルを行い、.objファイルを生成する。

4. 恒久対策

SuperH RISC engine ファミリ C/C++コンパイラパッケージV.9.04 Release 00で改修する予定です。

[免責事項]

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。ニュース本文中のURLを予告なしに変更または中止することがありますので、あらかじめご承知ください。