GNU ARM リンカスクリプト

1. リンカスクリプトのコマンド
   1. PROVIDE(3.5.2章)

あるケースでは、リンカスクリプト用のシンボルを定義することは、リンク時にインクルードされる任意のオブジェクトが参照されておりかつ定義されていない場合に限り、定義可能です。

例えば、伝統的なリンカでは’etext’のようなシンボルが定義されていました。

しかし、ANSI Cでは、ユーザーが、エラーを起こすことなく関数名として’etext’を使うことが可能です。PROVIDEキーワードは、’etext’のようなシンボルを、参照されているが、定義されていない場合に限り、シンボルの定義が可能となります。

シンタックスは**PROVIDE**(*symbol* = *expression*). です。

PROVIDEを使用して’etext’を定義する場合の例を示します。

SECTIONS

{

.text :

{

\*(.text)

\_etext = .;

PROVIDE(etext = .);

}

}

この例では、もしプログラム上で(アンダスコアを用いて)’\_etext’を定義している場合、リンカは多重定義エラーを出すでしょう。もし、(アンダスコアなしの)’etext’を定義しているプログラムの場合、リンカはプログラム上の定義を使用します。

もしプログラムが’etext’を参照しているが、定義がない場合リンカはリンカスクリプトの定義を使用します。

* 1. PROVIDE\_HIDDEN

PROVIDEコマンドと同様です。ELFターゲットポートのために、シンボルは隠され、かつエクスポートされることはありません。

* 1. KEEP

リンク時のガベージコレクションを使用(‘--gc-sections’)する場合削除されたくないセクションをマークすることがしばしば行われます。これを行うためにはインプットセクションのワイルドカード表現エントリをKEEP()で囲みます。　例えばKEEP(\*(.init))など

* 1. ASSERT

1. 「3.6.6 Output Section Keywords」

以下は、アウトプットセクションコマンドのためのキーワードです。

CREATE\_OBJECT\_SYMBOLS

このコマンドはリンカに各インプットファイルのためのシンボルを生成することを伝えるためのものです。各シンボルの名前は入力ファイルのシンボル名と一致します。各シンボルのセクションはCREATE\_OBJECTSYMBOLSコマンドが現れるアウトプットセクションにあります。

CONSTRUCTORS

a.out形式のオブジェクトファイルをリンクする際、リンカはC++ globalコンストラクタとデストラクタをサポートするために通常では使用されないコンストラクトセットを使用します。ECOFFやXCOFFなどの任意のセクションにオブジェクトファイルをリンクすることが許されていないファイルフォーマットでは、リンカが自動的に、名前によってC++ globalコンストラクタとデストラクタが認識されます。これらのオブジェクトファイル形式の場合、CONSTRUCTORSコマンドによって、アウトプットセクションにコンストラクタの情報を配置するためにリンカに伝えられます。これらはCONSTRUCTORSコマンドが現れたときに行われます。CONSTRUCTORSコマンドはそれ以外のオブジェクトファイル形式の場合には、無視されます。→ということで、ELFを使用するので詳細は割愛します。