Ambiente de Desenvolvimento

Walter Fetter Lages

w.fetter@ieee.org

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Engenharia Elétrica

ENG04008 Sistemas de Tempo Real

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Tópicos

- Editor de Textos
- Compilador
- Montador
- Linker
  - Linkagem Incremental
  - Linkagem Estática
  - Linkagem Dinâmica
- Make
- Gerenciador de Bibliotecas
- Depurador
- Ambiente Integrado de Desenvolvimento

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Introdução

- Linguagem de máquina
- Assembly
- Montador
- Compilador
- Sistema *Host*
- Sistema *Target*

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Editor de Textos

- Produz o arquivo fonte (`.c`, `.pas`, `.java`, `.cpp`, ...)
  - MS-DOS Editor (`edit`)
  - Windows notepad (`notepad`)
  - Norton Editor (`ne`)
  - Brief (`brief`)
  - Windows Programmer's Editor (`wpe`)
  - Joe's Own Editor (`joe`)
  - Visual Editor (`vi`)
  - VI Improoved (`vim`)
  - KDE Text Editor (`kedit`)
  - Editor of Macros (`emacs`)

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Compilador

- Traduz os módulos fonte para outra linguagem (`.asm`, `.s`) ou para módulos objeto (`.obj`, `.o`)
  - Microsoft C (`cc`)
  - Turbo Pascal (`tpc`)
  - Turbo C++ (`tcc`)
  - Borland C++ (`bcc`)
  - GNU Compiler Collection (`gcc`, `g++`, `g77`, `gjc`, `...`)

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Montador

- Traduz os módulos fonte em Assembly (`.asm`, `.s`) para módulos objeto (`.obj`, `.o`)
  - Microsoft Macro Assembler (`ml`, `masm`)
  - Turbo Assembler (`tasm`)
  - GNU Assembler (`as`)
  - Netwide Assembler (`nasm`)

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Linker

- Liga os diversos módulos objeto (`.obj`, `.o`) para criar:
  - um programa executável, ou
  - um módulo maior, ou
  - uma biblioteca de linkagem dinâmica
- As referências internas são resolvidas
  - Microsoft Linker (`link`)
  - Turbo Linker (`tlink`)
  - GNU Linker (`ld`, `collect2`)

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Gerenciador de Bibliotecas

- Agrupa diversos módulos objeto (`.obj`, `.o`) em um arquivo de biblioteca (`.lib`, `.a`)
- As referências internas **não** são resolvidas
- A unidade de linkagem é o módulo objeto
  - Microsoft Library Manager (`lib`)
  - Turbo Library Manager (`tlib`)
  - GNU Archive (`ar`)

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Depurador

- debug
- Codeview (`cv`)
- Turbo Debugger (`td`)
- GNU Debuger (`gdb`)
  - Front-ends (`xxgdb`, `kdbg`)

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Linkagem Incremental

- Diversos módulos objeto são ligados para formar um módulo objeto maior
- São resolvidas as referências internas

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Linkagem Estática

- Um ou mais módulos são ligados para criar um programa executável (`.exe`) ou imagem binária (`.com`)
- Podem ser ligados módulos em sí ou módulos contidos em bibliotecas estáticas
- Cada programa tem a sua própria copia dos módulos objeto (`.obj`, `.o`)

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Linkagem Dinâmica

- Bibliotecas dinâmicas (`.dll`, `.so`) não são ligadas com o programa pelo linker
- Carregada pelo sistema operacional
- Compartilhada por todos os programas que usam a biblioteca dinâmica

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Make

- Programa para automatizar e otimizar o processo de compilação e linkagem
- Executa apenas os procedimentos que são necessários
- Funciona baseado na comparação das datas e horas dos arquivos
- `Makefile` = Arquivo interpretado pelo `make`
  - Microsoft Program Maintenance Utility (`make`, `nmake`)
  - Borland Make (`make`)
  - GNU Make (`make`)

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

IDE

- Editor, Compilador, Linker, Make, . . . integrados
- Alguns podem ser configurados para utilizar diversos compiladores, linkers, . . .
  - Microsoft Visual C (`msc`)
  - Turbo Pascal (`turbo`)
  - Borland C++ (`bc`)
  - Windows Programmer's Editor (`wpe`)
  - KDE Development Environment (`kdevelop`)

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo 1

- Hello, World! em C
  - hello.c
  - Makefile

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

hello.c

```
#include <stdio.h>

int main(int argc,char *argv[])
{
        printf("Hello, world!\n");
        return 0;
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Makefile

```
CFLAGS=-O2 -Wall -march=i486
CINCLUDE=-I. -I${HOME}/include
CLIBDIR=-L${HOME}/lib
CLIBS=

CPPFLAGS=
CPPINCLUDE=-I${HOME}/include/cpp -I../include
CPPLIBDIR=-L../lib
CPPLIBS=

INCLUDE=${CINCLUDE} ${CPPINCLUDE}
FLAGS= ${CFLAGS} ${CPPFLAGS}
LIBDIR=${CLIBDIR} ${CPPLIBDIR}
LIBS=${CPPLIBS} ${CLIBS}

CMP= gcc
CMPFLAGS= ${FLAGS} ${INCLUDE}
LDFLAGS= ${LIBDIR} ${LIBS}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Makefile – Continuação

```
all: hello

hello: hello.c
	${CMP} ${CMPFLAGS} -o $@ $^ ${LDFLAGS}

clean:
	rm -f *~ *.bak *.o

install:

distclean: clean
	rm -f hello
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo 2

- Hello, World! em C++
  - hello.cpp
  - Makefile

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

hello.cpp

```
#include <iostream>

int main(int argc,char *argv[])
{
        cout << "Hello, world!\n";
        return 0;
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Makefile

```
CFLAGS=-O2 -Wall -march=i486
CINCLUDE=-I. -I${HOME}/include
CLIBDIR=-L${HOME}/lib
CLIBS=

CPPFLAGS=
CPPINCLUDE=-I${HOME}/include/cpp -I../include
CPPLIBDIR=-L../lib
CPPLIBS=

INCLUDE=${CINCLUDE} ${CPPINCLUDE}
FLAGS= ${CFLAGS} ${CPPFLAGS}
LIBDIR=${CLIBDIR} ${CPPLIBDIR}
LIBS=${CPPLIBS} ${CLIBS}

CMP= g++
CMPFLAGS= ${FLAGS} ${INCLUDE}
LDFLAGS= ${LIBDIR} ${LIBS}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Makefile – Continuação

```
all: hello

hello: hello.cpp
	${CMP} ${CMPFLAGS} -o $@ $^ ${LDFLAGS}

clean:
	rm -f *~ *.bak *.o

install:

distclean: clean
	rm -f hello
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo 3

- Hello, World! em Assembly
- Utiliza serviço do Linux para mostrar *string*
- Sintaxe do `gas`
  - hello.s
  - Makefile

hello.s

```
        .intel_syntax noprefix
        .text
        .global _start

_start:

        mov edx,offset len
        mov ecx,offset msg
        mov ebx,1
        mov eax,4
        int 0x80

        xor ebx,ebx
        mov eax,1
        int 0x80

        .data
msg:
        .string "Hello, world!\n"
len=    . - msg
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Makefile

```
ASMFLAGS=--gstabs

INCLUDE=
FLAGS=${ASMFLAGS}
LIBDIR=
LIBS=

LD= ld
CMP= as
CMPFLAGS= ${FLAGS} ${INCLUDE}
LDFLAGS=
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Makefile – Continuação

```
all: hello

hello: hello.o
	${LD} ${LDFLAGS} -o $@ ${LIBS} $^

hello.o: hello.s
	${CMP} ${CMPFLAGS} -o $@ $^

clean:
	rm -f *~ *.bak *.o

install:

distclean: clean
	rm -f hello *.lst *.map .kdbgrc.*
```