

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia de Sistemas Eletrônicos - PSI

PSI-2553- Projeto de Sistemas Integrados

Tutorial: O Processador Plasma (Parte Prática)

M.S. / W.J.C/ M.AR.J./ B.C.S.S. (16)

1. Objetivos

Este tutorial visa a familiarização do estudante com a arquitetura do processador Plasma, que segue a arquitetura MIPS, introduzida nas aulas de teoria. Serão realizados o estudo e a simulação com as instruções do processador com um programa que realiza o cálculo do número de Fibonacci.

2. Compilação e Montagem

A distribuição do Plasma é bastante completa, incluindo uma série de ferramentas para o desenvolvimento de sistemas embutidos, incluindo compilador, montador e simulador. O compilador converte um programa em linguagem de alto nível C para a linguagem *assembly* ou formato binário e o montador realiza a tarefa de conversão do *assembly* para o binário. A explicação a seguir mostra como é o procedimento apresentado na distribuição do Plasma no seu sítio da Internet, sendo que adaptações podem ser feitas, uma vez que as operações são realizadas por compilação de arquivos em linguagem C.

O compilador realiza uma série de tarefas como indicado na Figura 1.

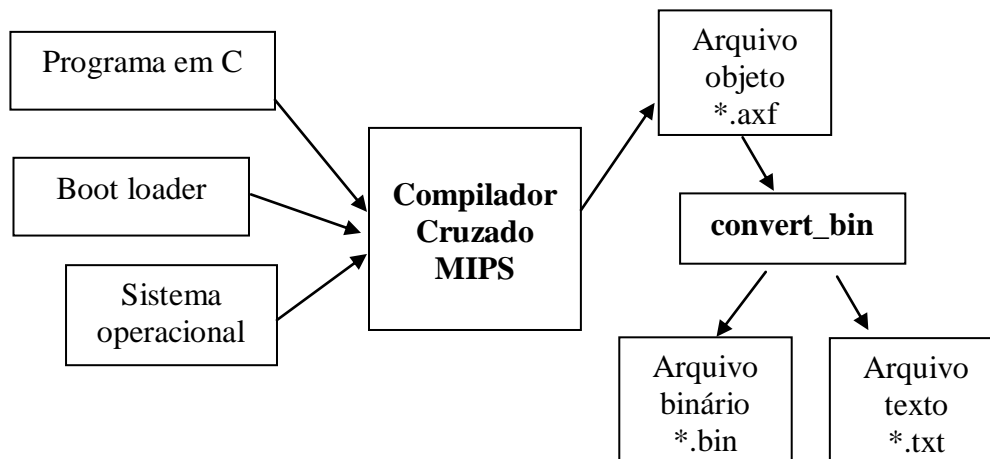


Figura 1. Fluxo de compilação de programa em linguagem C no Plasma

O *boot-loader* contém a inicialização dos registradores do Plasma (alguns não tratados ainda), enquanto o sistema operacional cuida dos periféricos (bloco *ethernet*, UART e *general purpose IO*, entre outros). O arquivo binário corresponde à imagem pronta para ser carregada na memória física, enquanto o arquivo texto permite verificar ao desenvolvedor acompanhar o código *assembly* intermediário gerado, assim como o equivalente hexadecimal da instrução.

Outra opção possível é desenvolver o código *assembly* e a partir dele realizar a montagem. Neste caso, o fluxo para a geração do arquivo binário passa a ser como ilustrado na Figura 2.

3. Iniciando o tutorial

3.1 Aprontando os Arquivos de Síntese, Compiladores e Ferramentas do Plasma

- 1) Criar a pasta **X:\psi2553\tut_mips**.

- 2) Copiar para **X:\psi2553\tut_mips** o arquivo **mips_tools.zip** que está em **Rede=>Newserverlab\psi2553\tut_mips**.
- 3) No subdiretório **X:\psi2553\tut_mips** extraia as pastas comprimidas que estão em **mips_tools.zip**. Com isso duas pastas deverão aparecer: **X:\psi2553\tut_mips\tools** e **X:\psi2553\tut_mips\gccmips_elf**. Atenção: toda compilação e montagem será feita no subdiretório **X:\psi2553\tut_mips\tools**, porém teremos que usar os programas via linhas de comando (diretamente no DOS).

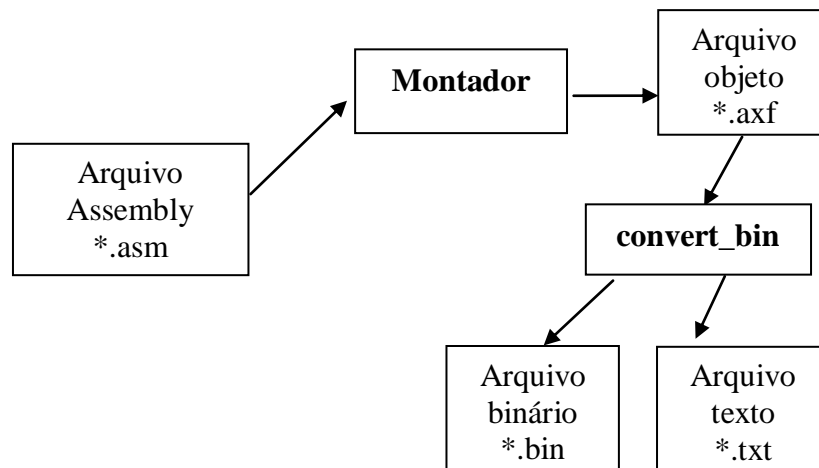


Figura 2. Fluxo partindo de um arquivo assembly.

3.2 Compilando programa em C

- 1) Leia e entenda os seguintes arquivos disponibilizados em **X:\psi2553\tut_mips\tools**:
 - a. fibonacci.c (programa fonte)
 - b. boot.asm (código de boot em assembly que é mesclado ao programa fonte)
 - c. no_os.c (código em C de operações de sistema, em ausência de SO)
 - d. makefile (*script* para compilação programada e incremental)
- 2) No Windows, selecione **Start→All Programs→Accessories→Command Prompt**. Uma janela de terminal (prompt de comando) se abrirá.
- 3) No prompt de comando, dê uma sequência de comandos **cd ..** voltando na hierarquia até alcançar a raiz.
- 4) Tecele **mkdir tmp** para criar o sub-diretório **C:\tmp**.
- 5) Tecele X:
- 6) Digite o comando **cd psi2553\tut_mips\tools** para entrar em **X:\psi2553\tut_mips\tools**
- 7) Tecele **gmake fibonacci**. A compilação e montagem, como descrita na parte teórica, será realizada. (Observação: gmake roda o arquivo makefile)
- 8) Olhe o arquivo **test.map**. Descubra nele quais os endereços do main e da função fibonacci de fibonacci.c.
- 9) Olhe o arquivo **test.lst**. Entenda o código assembly (com auxílio dos itens 1 e 8), e verifique que a função fibonacci é executada.

3.3 Simulando

- 1) Tecle na janela DOS **gmake run** . A simulação será apresentada com algumas opções.
- 2) Inicialmente tecle a opção 1 seguidamente e verifique que o programa assembly é executado passo a passo. Observe como variam os valores dos registradores (apresentados nas 4 linhas iniciais).
- 3) Explore outras possibilidades de depuração.

3.4 Sugestão de realizar a montagem a partir de um código assembly

- 1) A partir do test.lst, edite um código assembly para e chame o arquivo de fibonacci.asm
- 2) Modifique o código do arquivo makefile para poder realizar a montagem diretamente de fibonacci.asm. Para tal, entenda o formato interno do makefile e edite a parte relativa ao make iniciada por **fibonacci:** . Em caso de dúvidas, consulte o manual X:\psi2553\tut_mips\make.html para entender o funcionamento de um makefile.
- 3) Rode o makefile, gerando assim os arquivos objeto de saída e os códigos binários.
- 4) Faça uma nova simulação.