Fusion C no GNU Linux

Se você gosta de pinguim não só nos jogos da Konami, mas também no sistema operacional, seguem aqui as instruções de como configurar a Fusion-C no GNU Linux. Feita essa configuração, você pode continuar a seguir o artigo a partir da seção **Rodando o Programa test.c**.

# Instalação no GNU Linux

Ao extrair o arquivo **MSX\_Fusion-C-V1.2.zip** (outras versões vão nomes parecidos), você terá a pasta **Working Folder**. Nela você vai desenvolver seus programas, e você já encontra o arquivo exemplo **test.c**, além de outras pastas de interesse:

* **dsk**: pasta onde o arquivo **.com** será gerado. Contém uma cópia do MSX DOS2.
  + Se seu programa carregar arquivos, eles ficam aqui.
* **fusion-c**: cabeçalhos e o binário **.lib** da biblioteca, além da pasta **examples**, com os exemplos.

O Toolchain da biblioteca vem com os binários para Windows, que não funcionam no Linux e, pra coroar, a biblioteca tem um **Makefile** incorreto (pode ser que funcione em Mac, mas não sei dizer). Para resolver isso, temos que fazer alguns passos extras. Após descompactar a biblioteca num diretório para o seu usuário (exemplo, **~/MSX\_Fusion-C\_V1.2**), você deve baixar o binário da versão 3.6 ou superior do SDCC diretamente do site oficial em <http://sdcc.sourceforge.net/> (distros Debian ainda usam a versão 2.X), além do código fonte do **Hex2bin-2.5** ou superior de <https://sourceforge.net/projects/hex2bin/>.

Uma vez que você descompacte os arquivos .bz2 (tar xjf <nome do arquivo>), você terá dois diretórios, **sdcc-3.6.0** e **Hex2bin-2.5**. Assumindo que você baixou e descompactou esses arquivos no diretório **Downloads** (padrão de muitos navegadores), você pode instalar as ferramentas com os seguintes comandos (você deve ter o gcc e make instalados):

|  |
| --- |
| sudo cp -r ~/Downloads/sdcc-3.6.0/\* /usr/local  cd Downloads/Hex2bin-2.5  make  sudo make install |

O SDCC vem com versões padrão das funções padrão **putchar**, **getchar** e **printf** que não funcionam no MSX. Para resolver esse problema, a Fusion C utiliza a versão do Konamiman dessas funções. Se você instalou o SDCC como indicado acima, basta você aplicar os seguintes comandos:

|  |
| --- |
| cd /usr/local/share/sdcc/lib/z80  sudo cp z80.lib z80.save  sudo sdar -d z80.lib printf.rel  sudo sdar -d z80.lib sprintf.rel  sudo sdar -d z80.lib vprintf.rel  sudo sdar -d z80.lib putchar.rel  sudo sdar -d z80.lib getchar.rel |

O **sdar** é um comando do SDCC que gerencia as bibliotecas .lib, então ao executar os comandos acima nós removemos essas funções da biblioteca padrão do SDCC.

Instele o **openMSX** e o complemento **openMSX Catapult** (um *launcher* visual) com o gerenciador de pacotes da sua distribuição, mas ele vem só com as máquinas com C-Bios que rodam apenas cartucho. Para usar as que tem drive, copie os arquivos do diretório Tools da SDCC no diretório de configuração do openMSX:

|  |
| --- |
| cp ~/MSX/MSX\_Fusion-C\_V1.2/Working\ Folder/Tools/\_For\ OpenMSX/systemroms/\* ~/.openMSX/share/systemroms |

Por padrão a biblioteca usa um MSX europeu a 50Hz. Vamos mudar para um FS-A1WSX, um MSX 2+ a 60Hz rodando o MSX DOS2, basta modificar o arquivo **Working Folder/openMSX/emul\_start\_config.txt** para:

|  |
| --- |
| machine Panasonic\_FS-A1WSX  ext ASCII\_MSX-DOS2  bind F12 cycle videosource  plug joyporta  diska ./dsk |

Se você prefere usar o MSXDOS com apenas 64kB de memória (próximo a um MSX da época), remova a segunda linha (extensão do MSXDOS2) e adicione em **Working Folder/dsk** os arquivos do MSXDOS (COMMAND.COM e MSXDOS.SYS), que devem ser baixados de um site como o **MSX Pro**.

Inicialmente, o emulador vem configurado por padrão para usar filtros específicos. Execute ele uma vez com o openMSX Catapult e configure o visual como você preferir.

# Modificando o Makefile

Finalmente, para compilar o programa exemplo, devemos modificar o **Makefile**. De acordo com o autor deve funcionar em MacOS, mas certamente não funciona no Linux. Dentro do diretório ~/MSX\_Fusion-C\_V1.2/Working Folder, modifique o arquivo Makefile para o seguinte:

|  |
| --- |
| TARGET=test  DEST := ./dsk/  INCLUDEDIR = ./fusion-c/include/  LIBDIR = ./fusion-c/lib/  crt0 = $(INCLUDEDIR)crt0\_msxdos.rel  # use este crt0 para passar parâmetros  #crt0 = $(INCLUDEDIR)crt0\_msxdos\_advanced.rel  ADDR\_CODE = 0x106  # use este valor se usar o crt0\_msxdos\_advanced  #ADDR\_CODE = 0x180  ADDR\_DATA = 0x0  CCFLAGS = --code-loc $(ADDR\_CODE) --data-loc $(ADDR\_DATA) --disable-warning 196 -mz80 --no-std-crt0 --opt-code-size fusion.lib -L $(LIBDIR) -I $(LIBDIR) $(crt0)  compile:  sdcc $(CCFLAGS) $(TARGET).c  com:  @hex2bin -e com $(TARGET).ihx  @cp $(TARGET).com $(DEST)  bin: compile com  all: bin emulator  clean:  @rm -f $(DEST)/$(TARGET).com \*.asm \*.lst \*.sym \*.bin \*.ihx \*.lk \*.map \*.noi \*.rel  emulator:  openmsx -script ./openMSX/emul\_start\_config.txt & |

# Rodando o Programa Test.c

No Linux, basta rodar em **~/MSX\_Fusion-C\_V1.2/Working Folder** o comando **make all**, que compila e executa o emulador. Caso você queira alterar o programa sem abrir novamente o emulador, basta executar o comando **make bin**, que atualiza o **.com** mas não reinicia o emulador.

|  |
| --- |
| Dica: eu costumo rodar o emulador com o alvo make emulator e deixo ele aberto. Assim podemos recompilar com o alvo make bin e testar direto no emulador, sem precisar passar pela sequência de boot do MSX. |

# Rodando o Exemplo sprites.c

Para compilar o programa exemplo do jogo, mude no Makefile o target para o exemplo do jogo:

|  |
| --- |
| TARGET=sprites |

Assim, ao executar o **make all** ou **make bin**, o arquivo compilado será o do exemplo do jogo.

# Melhoras

Apesar de funcional, a configuração proposta pelos autores da biblioteca não segue o padrão de bibliotecas C. Para melhorar isso, você pode tornar os arquivos da biblioteca (includes e bibliotecas) parte do sistema. Isso vai permitir incluir os cabeçalhos .h com **<** e **>**, ao invés de entre aspas. Além disso, você pode centralizar a configuração do openMSX no seu diretório home, compartilhando o arquivo em vários projetos.

# Usando o Diretório Tools

Algumas ferramentas que acompanham a Fusion-C rodam no Linux, como o DskTool, enquanto outras podem rodar com o Wine, o emulador de binários Windows para Linux.