

Uso do CMAKE

Esta página faz uma orientação geral para compilação do código fonte do SW7 na AWS.

Porque?

A demanda por uma nova maneira de compilação se deu pela migração do SW75 para a cloud da Amazon. Esta migração provocou uma troca nos ambientes do produto, onde deixamos de utilizar Unix AIX com compilador XLC para utilizar Linux Red Hat com compilador GCC.

Para organizar as diretivas de compilação GCC do projeto decidi-se por adotar o Makefile. Com o Makefile e a ferramenta 'make' automatizamos a construção dos objetos e binários, bem como a linkagem com as bibliotecas.

Ao começarmos a construção dos Makefiles deparamos com uma dificuldade de replicá-los entre os módulos e submódulos da aplicação. Esta é uma característica já conhecida dos Makefiles, em que a sua criação manual é indicada para projetos de pequeno e médio porte.

A alternativa para contornar esta dificuldade foi adotar um gerador automático de Makefiles: CMake.

O que é CMake?

CMake é um sistema que gerencia o processo de compilação de um software. O CMake **não substitui** o Makefile. O uso do CMake é um passo anterior, onde através dele, que tem uma linguagem mais amigável, criamos automaticamente os Makefiles responsáveis por organizar e direcionar a compilação do código fonte.

Outra vantagem do CMake é ser independente de compilador. Acima falamos sobre utilizá-lo para criar Makefiles, mas podem, também, criar arquivos para compilação via Ninja, QT Creator, Microsoft Visual Studio e outros.

Como Instalar CMake?

O CMake por padrão estará disponível nas instâncias de desenvolvimento dos projetos 'online'. Caso não encontre, pode instalá-lo da seguinte maneira:

```
wget https://github.com/Kitware/CMake/releases/download/v3.18.3/cmake-3.18.3-Linux-x86_64.tar.gz
tar xvf cmake-3.18.3-Linux-x86_64.tar.gz -C <dir destino>
```

Pronto, o binário estará disponível para utilização em <dir destino>/bin/cmake.

Organização

Os scripts CMakeListstxt suportam hierarquia por diretório, podendo invocar uns aos outros.

Nos repositórios do SW7-online criamos o primeiro CMakeLists.txt no diretório de "src" do projeto. A partir daí este script chama os demais CMakeLists.txt contido nos módulos ou, ainda, nos diretórios de include, libsrc, etc.

Como Funciona?

O CMake faz uso apenas de scripts chamados CMakeLists.txt. Estes scripts contém O QUE nós gostaríamos de compilar e suas dependências. Note que não será dito em COMO compilar, esta informação é abstraída pelo CMake e estará contida apenas no Makefile gerado.

Para facilitar o processo de compilação do SW7 online foi criado um script, chamado de **swmake**, para abstrair as particularidades do CMake e make, tornando a compilação mais direta e amigável.

CMake + make

A chamada do cmake segue o formato:

```
cmake <local onde esta o CMakeLists.txt>
```

O CMake, durante o processo de criação de Makefiles, cria diversos outros arquivos de controle. Para não poluirmos o repositório, é uma boa prática direcionar o CMake para utilizar um diretório para armazenar todo arquivo gerado por ele. Este direcionamento ocorre através da flag "-B":

```
cmake <local onde esta o CMakeLists.txt> -B <diretório>
```

Após execução do cmake, tem-se um <diretório build> com os Makefiles contidos. Neste momento já podemos invocar o **make** para interpretação dos Makefiles e consequente compilação do projeto.

```
cd <diretório build>
make
make install DESTDIR=<diretorio install>
```

ou faz uso do parâmetro -C para identificar a localidade do diretório com Makefiles

```
make -C <diretorio build>
make install -C <diretório build> DESTDIR=<diretorio install>
```

Abaixo um exemplo de uso:

```
cmake . -B build
make -C build
make install -C build DESTDIR=~/.deploy
```

swmake

O **swmake** [repo] é um script criado para abstrair as chamadas do CMake e dos Makefiles subsequentes.

Ele foi criado para evitar que toda vez formos compilar, não tenhamos que: *ir até o repositório... executar o cmake... executar o make... mover os binários gerados..., etc.*

Para informações detalhadas, leia o *usage* – exibido ao executar **swmake** sem parâmetros

A chamada do swmake segue o seguinte formato:

```
swmake <make-target> <project> [-mD]
```

Onde:

- **make-target** [opcional – caso não seja enviado é utilizado target **all**] é qual tipo de make desejamos fazer, tendo como possibilidades: all, clean, build, install. A ideia é ser semelhante ao próprio comando make,
- **project** indica qual o projeto no qual será feito esse make. **Ex:** pdv, pos, gen-lib, gen-deliverable, etc.
- **[-mD]** *argumentos opcionais*
 - m indica em qual módulo específico deste projeto queremos realizar o make. **Ex:** fsw, pdv_common, dbaccess, etc.
 - d indica em qual diretório os entregáveis gerados serão instalados pelo *make install*. Este argumento na maior parte das vezes só é útil para a pipeline CI/CD
 - D indica para o CMake que deve utilizar flags de DEBUG na compilação (essas flags são definidas por nós, nos CMakeFiles.txt