

6 Referências Bibliográficas

ABNT/CEET-00:001.85. Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiofusão digital – Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações. Setembro 2007.

Babin, Steve, Harrison, Richard. Developing Software for Symbian OS An Introduction to Creating Smartphone Applications in C++. Jonh Wiley & Sons Ltd, 2006.

Berenger, Francis Machado, Maia, Luiz Paulo. Arquitetura de Sistemas Operacionais. 3ª edição. LTC, 2002.

Costa R.M.R., Soares L.F.G.. Modelo Temporal Hiperídia para Suporte a Apresentações em Ambientes Interativos. Rio de Janeiro: PUC-Rio - Departamento de Informática. 2007.

Cruz, Vitor Medina, Moreno, Marcio Ferreira, Luiz Fernando Gomes, Soares. TV Digital Para Dispositivos Portáteis – Middlewares. Relatório Técnico; Rio de Janeiro: PUC-Rio – Departamento de Informática, jan. 2008.

Devsphere. SAX versus DOM. Benchmark performed with Xerces and Crimson. Disponível em: <http://www.devsphere.com/xml/benchmark/method.html>. 2007. Último acesso em Fevereiro de 2008.

directfb.org | Main. Disponível em: <http://www.directfb.org/>. mar. de 2008. Último acesso em Março de 2008.

Digital TV Facts. DTV Guide Transmission. Disponível em: <http://dtvfacts.com/latest/250/as-digital-tv-reception-controversy-dims-e-vsbs-gets-another-look/>. maio 2006. Último acesso em janeiro de 2008.

Farwick, Matthias, Hafner ,Michael. XML.com: XML Parser Benchmarks: Part 1. Disponível em: <http://www.xml.com/pub/a/2007/05/09/xml-parser-benchmarks-part-1.html>. Maio de 2007. Último acesso em Fevereiro de 2008.

Forum Nokia. Open C for S60:Increasing Developer Productivity, Version 1.2. mar 2007.

Franklin, Steve. XML Parsers: DOM and SAX Put to the Test. Disponível em: <http://www.devx.com/xml/Article/16922/1954?pf=true>. 2007. Último acesso em Fevereiro de 2008.

Harrison, Richard et al. Symbian OS C++ for Mobile Phones. Volume 1. John Wiley & Sons, 2003.

Harrison, Richard et al. Symbian OS C++ for Mobile Phones. Volume 2. John Wiley & Sons, 2004.

Java Technology. Disponível em: <http://java.sun.com/>. Último acesso em Setembro de 2007.

Jez, Marco. Disponível em: <http://marcoplusplus.blogspot.com/2007/05/stlport-for-symbian-os-released.html>. maio 2007. Último acesso em Fevereiro de 2008.

Moreno, Marcio Ferreira. Um Middleware Declarativo para Sistemas de TV Digital Interativa. Dissertação de mestrado; Rio de Janeiro: PUC-Rio – Departamento de Informática, abril de 2006.

Morley, Jason. Leaves and Exceptions Version: 1.3.1. Published by the Symbian Developer Network, set 2007.

Nokia. Device Details -- Nokia 5700 XpressMusic. Disponível em: http://www.forum.nokia.com/devices/5700_XpressMusic. 2008. Último acesso em Julho de 2008.

Nokia. Device Details -- Nokia N81. Disponível em: <http://www.forum.nokia.com/devices/N81>. 2008. Último acesso em Julho de 2008.

Nokia. S60 Platform SDKs for Symbian OS, for C++. Disponível em: <http://www.forum.nokia.com/info/sw.nokia.com/id/4a7149a5-95a5-4726-913a-3c6f21eb65a5/S60-SDK-0616-3.0-mr.html>. 2008. Último acesso em Fevereiro de 2008.

Nokia. Open C SDK Plug-In for S60 3rd Edition SDKs, for Symbian OS, for C++. Disponível em: http://www.forum.nokia.com/info/sw.nokia.com/id/91d89929-fb8c-4d66-bea0-227e42df9053/Open_C_SDK_Plug-In.html. 2008. Último acesso em Fevereiro de 2008.

Nokia. Carbide.c++ v1.2, the development tool for C++ for Symbian OS and Open C developers. Disponível em: <http://www.forum.nokia.com/info/sw.nokia.com/id/dbb8841d-832c-43a6-be13-f78119a2b4cb.html>. 2008. Último Acesso em Fevereiro de 2008.

Oren, Yuval. SAX Parser Benchmarks. Disponível em: <http://piccolo.sourceforge.net/bench.html>. 2002. Último acesso em Fevereiro de 2008.

SDN. P.I.P.S. - Symbian Development Public Wiki – SDN Wiki. Disponível em: <http://developer.symbian.com/wiki/display/pub/P.I.P.S>. Fevereiro de 2008. Último acesso em Fevereiro de 2008.

Silva, Heron V. O., Rodrigues, Rogério Ferreira, Soares, Luiz Fernando Gomes. Frameworks para Processamento de Documentos XML. Rio de Janeiro: Departamento de Informática, PUC-Rio. 2005.

Smil 2.1 – Timing and Synchronization. Disponível em: <http://www.w3.org/TR/2005/REC-SMIL2-20051213/smil-timing.html>. dec. 2005. Último acesso em fevereiro de 2007.

Soares, L.F.G., Rodrigues, R.F., Moreno, M.F. Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System. Rio de Janeiro: PUC-Rio - Departamento de Informática. abril 2007.

SOARES, L.F.G., RODRIGUES, R.F., MUCHALUAT-SAADE, D.C. Modelo de Contextos Aninhados – versão 3.0, Relatório Técnico, Laboratório TeleMídia, Departamento de Informática, PUC-Rio, 2003.

Stichbury, Jo. Symbian OS Explained. Effective C++ Programming for Smartphones. John Wiley & Sons Ltd, 2004.

STLPort. Disponível em: <http://www.stlport.org/>. 2001. Último acesso em Fevereiro de 2008.

Symbian Ltd. P.I.P.S. Home - P.I.P.S. - Symbian Wiki. Disponível em: <http://developer.symbian.com/wiki/display/oe/P.I.P.S.+Home>. fev de 2008. Último acesso em Fevereiro de 2008.

Symbian Ltd. Using Symbian OS GETTING STARTED, 2007 2nd edition. 2-6 Boundary Row, London SE1 8HP, UK, jan. de 2007.

Symbian Ltd. Using Symbian OS P.I.P.S. fev de 2007.

Symbian Ltd, Nokia. S60 3rd Edition SDK for Symbian OS, Supporting Feature Pack 1, for C++. SDK Help. out. de 2006.

Violleau, Thierry. Java Technology and XML-Part Two. Disponível em: http://java.sun.com/developer/technicalArticles/xml/JavaTechandXML_part2/. Março 2002. Último Acesso em Fevereiro de 2008.

Apêndice A – Instalação e Dispositivos Testados

Este apêndice tem como objetivo servir como um manual para a criação do ambiente necessário à compilação dos pacotes Ginga-NCL, bem como para a execução da aplicação resultante em um emulador de dispositivo Symbian ou para a instalação dessa aplicação em um dispositivo Symbian real. Ao final, também serão apresentados os dispositivos onde o Ginga-NCL foi instalado e os resultados obtidos.

Todas as ferramentas Symbian são disponibilizadas, oficialmente, apenas nas plataformas Windows XP e Windows 2000. Sendo assim, todas as instalações especificadas por esta seção devem ser feitas em um desses dois sistemas.

O primeiro elemento que precisa ser instalado para a criação do ambiente necessário a compilação e execução do Ginga-NCL é o S60 3rd Edition SDK F1, que é um SDK específico para a versão 9.2 do sistema operacional Symbian. Esse SDK pode ser encontrado em (Nokia, 2008c). O procedimento de instalação é simples e envolve a leitura e aceitação da licença, além de algumas escolhas típicas de instalação. Em seguida, precisam ser instaladas as bibliotecas P.I.P.S., Open C e STLPort para Symbian, que podem ser encontradas, respectivamente, em (SDN, 2008), (Nokia, 2008d) e (Jez, 2007). A primeira foi desenvolvida pela Symbian e é usada na realização do porte de códigos redigidos em C. A segunda é uma extensão (superset) da P.I.P.S., ou seja, oferece uma capacidade de porte ainda maior, mas funciona somente para a plataforma do Symbian S60. A terceira é um porte, feito por um membro da comunidade Symbian, de uma implementação da biblioteca STL que foi originalmente desenvolvida para dispositivos embedded. A necessidade dessas API's foi explicada na seção 3.2. O procedimento de instalação dessas bibliotecas pode ser resumido nas seguintes etapas:

- Descompactar os arquivos zip do PIPS do Open C e da STLPort em três pastas separadas;
 - Copiar o conteúdo da pasta epoc32 presente em cada uma das três pastas e colá-lo dentro da pasta <Diretório_de_Instalação_do_SDK_3rd_Edition_FP1>\Eoc32\.
- Quando solicitado, é preciso selecionar a opção de substituir todos os arquivos repetidos pelos os que estão sendo copiados.

Depois que essas três bibliotecas estiverem instaladas, o próximo passo consiste em instalar um ambiente de desenvolvimento apropriado. Como o Carbide C++ Free 1.2 foi utilizado ao longo de todo o processo de desenvolvimento do Ginga-NCL, e como os projetos da implementação Ginga-NCL para dispositivos portáteis estão no formato dessa ferramenta, recomenda-se o uso do Carbide C++ Free 1.2 ou, ainda, uma versão sua paga. O Carbide C++ Free 1.2 pode ser encontrado em (Nokia, 2008e) e a sua instalação também é trivial. Ressalta-se, entretanto, que, para o seu correto funcionamento, o ActivePerl-5.6.1.635 deve ser instalado. Se uma versão mais nova ou mais antiga desse elemento for instalada, o Carbide não funcionará corretamente. Para fazer uso das bibliotecas P.I.P.S. Open C e STLPort nessa ferramenta, os seguintes procedimentos devem ser realizados:

1. Dada a existência de um projeto Carbide 1.2, abrir o arquivo <Nome_do_Projeto>.mmp que se encontra dentro da pasta "group" do projeto. Nele são definidos todos os includes e as bibliotecas usadas pelo projeto;
2. Adicionar "/epoc32/include/stdapis" em System Include para fazer uso do PIPS e do OpenC;
3. Adicionar "/epoc32/include/stlport" em System Include para fazer uso da STLPort;
4. Adicionar a "libc" e a "stlport_s" na guia de bibliotecas. A primeira deve ser ligada dinamicamente enquanto que a segunda deve ser ligada estaticamente.

A implementação do Ginga-NCL é dividida em seis pacotes: `telemidia-util_Symbian`, `ncl30_Symbian`, `ncl30-converter_Symbian`, `gingaccio_Symbian`, `gingacc-player-Symbian` e `gingancl_Symbian`. O primeiro implementa algumas funções auxiliares que são muito usadas nos outros pacotes. O segundo, a estrutura NCM da qual o documento NCL deverá ser convertido, e o terceiro pacote é o que fica responsável por realizar a conversão propriamente dita, sendo nele, portanto, onde é encontrado o parser SAX. O quarto pacote implementa o controle de tela e dos gráficos, enquanto que o quinto, os exibidores. O último pacote corresponde a máquina de apresentação, e é, portanto, aquele que gera a aplicação Ginga-NCL.

Para importar um projeto Symbian no Carbide C++, basta selecionar o menu File->Import dessa ferramenta para que uma janela de importação apareça com uma série de opções, incluído duas para projetos Symbian: a "Symbian OS Bdl.inf file" e a "Symbian OS Executable". A primeira opção deve

ser selecionada. Em seguida, será necessário selecionar um arquivo `bdl.inf` para ser importado, que normalmente encontra-se dentro da pasta “group” de um projeto Symbian. Esse tipo de arquivo guarda informações sobre a organização do projeto. Depois disso, será preciso escolher as configurações de “Build”. Cada SDK oferece duas opções desse tipo, uma que gera binários para execução no emulador e outra para execução nos dispositivos portáteis. Ambas devem ser selecionadas. Em seguida, deverão ser escolhidos os arquivos de compilação que devem ser importados. Todos os disponíveis devem ser selecionados. Por fim, será solicitado o nome a ser dado para o projeto importado no Carbide C++ e em que pasta ele será copiado. Recomenda-se aceitar os valores padrões. Esse procedimento deve ser feito para todos os seis pacotes do Ginga-NCL para que a sua compilação possa ser efetuada.

A compilação pode ser feita de forma a gerar binários tanto para o emulador quanto para dispositivos Symbian S60. Para escolher uma dessas opções, é necessário clicar com o botão direito do mouse no projeto e ir em Active Build Configuration. Serão dadas as duas opções de Build: “Emulator Debug” e “Phone Release”, sendo que a primeira gera um binário do programa para o emulador, e a segunda, o binário para um dispositivo Symbian S60. Seja qual for a opção escolhida, a ordem correta de compilação dos pacotes é `telemidia-util_Symbian`, `ncl30_Symbian`, `ncl30-converter_Symbian`, `gingacc-io_Symbian`, `gingacc-player-Symbian` e `gingancl_Symbian`, respectivamente.

Para executar o Ginga-NCL no emulador provido pelo SDK, basta clicar com o botão direito do mouse no projeto `gingancl_Symbian` e selecionar a opção RunAs-> Run Symbian OS Application. O emulador será iniciado e a aplicação Ginga-NCL poderá ser encontrada dentro do sub-menu “Installed” do menu do celular emulado.

Para instalar a aplicação Ginga-NCL em um dispositivo S60 que possua o Symbian 9.2, é necessário usar o arquivo de instalação `.sisx` que é gerado dentro da pasta `sis` do pacote `gingancl_Symbian` após o término da compilação. Para realizar a instalação, um software de sincronismo, como o Nokia PC Suit, deve ser usado com o objetivo de se estabelecer uma conexão entre um dispositivo portátil e um computador que possua o arquivo de instalação `sisx`. Uma vez estabelecida essa conexão, basta clicar duas vezes sobre o arquivo `.sisx` de instalação do Ginga-NCL que um wizzard de instalação vai aparecer na tela do dispositivo conectado. Por fim, basta seguir as instruções do wizzard e a aplicação será instalada. As bibliotecas P.I.P.S. e Open C também precisam ser

instaladas no dispositivo. Os arquivos de instalação dessas bibliotecas podem ser encontrados na pasta S60Opensis do arquivo .zip baixado de (Nokia, 2008d).

Antes de executar o Ginga-NCL, tanto no dispositivo como no emulador, é preciso criar uma pasta de nome “Ginga” na raiz do sistema. No dispositivo portátil, a raiz é a pasta “Data”. No emulador, a raiz do dispositivo é emulada na pasta <Diretório_de_Instalação_do_SDK_3rd_Edition>\Epoc32\winscw\c\Data. A pasta “config”, que se encontra dentro da pasta “data” do pacote `gingancl_Symbian`, deve ser colocada dentro da pasta “Ginga” recém criada. A “config” possui uma série de arquivos de configuração do Ginga-NCL que são necessários para a sua execução. Dentro da pasta Ginga também devem ser colocadas todas as aplicações interativas e um arquivo de nome `ginga.conf`. Esse arquivo deve possuir, em seu conteúdo, referências às ncls das aplicações interativas que se encontram dentro da pasta Ginga. Cada linha desse arquivo deve possuir apenas uma referência para uma ncl. Por exemplo, imagine que existam duas aplicações interativas dentro da pasta “Ginga”: `matrix` e `formula1`. Dentro da pasta “matrix” existe uma ncl chamada “matrix.ncl”, e dentro da pasta “formula1”, uma outra ncl chamada “formula1.ncl”. Nesse contexto, o arquivo `ginga.conf` deverá ser criado da seguinte forma:

```
matrix/matrix.ncl
formula1/formula1.ncl
```

Dessa forma, quando o Ginga-NCL for executado, uma lista com essas duas aplicações será apresentada. Aquela que for selecionada será tocada pelo middleware.

A implementação do Ginga-NCL para dispositivos portáteis foi instalada e testada em dois dispositivos: o Nokia 5700 Express Music e o Nokia N81 1G.

O Nokia 5700 Express Music possui 369 Mhz e 64 Mb de memória RAM com aproximadamente 18 Mb livres para o uso em aplicações. Já o Nokia N81 1G possui 369 Mhz e 96 Mb de memória RAM com aproximadamente 42 Mb livres para o uso em aplicações. Ambos os aparelhos suportam os seguintes formatos de vídeo e áudio: 3GPPH.263, H.264/AVC, MPEG-4 AAC, eAAC+, MP3, MP4, M4A, WMA, AMR-NB, AMR-WB, Mobile XMF, SP-MIDI, MIDI Tones (poly 64) e WAV (Nokia, 2008a) e (Nokia, 2008b). Somente o Nokia N81 1G possui suporte a WLAN, que é oferecido através de interfaces 802.11 b/g, WPA ou WPA2 (AES/TKIP) (Nokia, 2008b).

Em ambos dispositivos testados, o Ginga-NCL funcionou corretamente sem qualquer problema aparente de desempenho e da mesma forma que no emulador.