

MeeGo 嵌入式操作系统的研究及应用开发*

韩德强¹, 冯云贺¹, 刘增辉²

(1. 北京工业大学 计算机学院, 北京 100124;

2. 北京电子科技职业学院, 北京 100029)

摘要: 论述了在 Linux 平台下搭建 MeeGo 嵌入式操作系统应用程序开发环境的方法和步骤, 给出了使用 Qt Creator 工具开发 MeeGo 应用程序的流程。通过在 Atom D510 嵌入式实验平台和 QEMU 虚拟机下安装 MeeGo 操作系统, 设计实现了具有 3D 操作界面且集成办公、娱乐、微博聊天、地图查询四大模块为一体的应用程序。该应用程序具有友好的人机交互界面和良好的可移植性, 体现了“一次编写, 到处运行”的理念。

关键词: 嵌入式系统; MeeGo; Qt; 可移植性

中图分类号: TP316.5

文献标识码: A

文章编号: 0258-7998(2012)10-0030-04

The research and application development of MeeGo embedded operating system

Han Deqiang¹, Feng Yunhe¹, Liu Zenghui²

(1. College of Computer, Beijing University of Technology, Beijing 100124, China;

2. Beijing Vocational College of Electronic Science, Beijing 100029, China)

Abstract: This paper discusses the methods and steps of establishing MeeGo application development environment on Linux platform and the procedures of MeeGo application development using Qt Creator tools. By installing MeeGo operating system on Atom D510 embedded test platform and on QEMU virtual machine, a 3D user interface application which integrates office module, entertainment module, microblog and chat module, map search module has been designed and implemented. The application has a friendly interactive interface and good portability indicating the concept of "write once, run everywhere".

Key words: embedded systems; MeeGo; Qt; program portability

MeeGo 嵌入式操作系统由英特尔和诺基亚倡导, 旨在为手机、上网本、平板电脑、互联网电视、车载信息娱乐系统和多媒体电话等智能设备提供操作系统解决方案, 并且有助于这些设备实现无缝集成。MeeGo 融合了英特尔 Moblin 与诺基亚 Maemo 系统的精华, 继承了分别掌握底层架构和终端设备优势的两大巨头的优秀基因。该嵌入式操作系统基于 Linux 架构, 不仅支持所有 x86 处理器平台, 还跨平台支持非 x86 架构的设备。面向跨平台的多元化设备, MeeGo 为开发者提供了一致性的 API, 选择 Qt 框架作为应用程序的开发工具^[1]。MeeGo SDK1.x 中集成了基于 Qt 4.7.x 的 Qt Creator 2.0, 并引进了功能强大的 Qt Quick, 使得应用程序更加快速流畅地在 MeeGo 平台上运行。

1 MeeGo 嵌入式操作系统架构

1.1 MeeGo 的分层模型

MeeGo 嵌入式操作系统的分层结构如图 1 所示, 共包括三层: 用户体验层、应用开发层、操作系统核心层。

用户体验层为不同的终端设备提供不同的应用框架, 比如上网本采用了 Clutter 和 MX 界面开发库绘制界面; 手持设备采用了 MeeGo 特有的触控框架, 包括触摸、输入法、手势, 为用户提供触控体验。MeeGo 用户体验层集成了个人社交网络服务和底层基础库, 全面支持网络服务、多媒体服务、最新电话技术等标准。强大的工具和动画效果简化了用户界面的定制。

应用开发层涵盖了 MeeGo 的应用开发接口, 目前 MeeGo release 版本包含了 Qt4.7、Qt mobility 1.0、Open

* 基金项目: 2010 度“教育部—英特尔精品课程”建设项目(教高司函[2010]217 号)



图1 MeeGo 系统分层模型^[2]

GL ES 1.1 和 Open GL ES 2.0 相关的类和接口。

操作系统核心层包含了所有中间件和操作系统的服务域以及硬件适配层。内核方面, MeeGo 1.0 采用了 Linux Kernel 2.6.33, 支持多任务、本地应用开发框架、动画效果设计的开发框架、3D 加速、最新 Linux 文件系统(btrfs)等。MeeGo 1.1 支持 Linux Kernel 2.6.35, 新增支持 Intel Atom Z6xx 处理器和 GCC4.5 工具链。MeeGo 1.2 支持 Linux Kernel 2.6.37, 新增 Policy 开发框架和电源管理功能。

1.2 MeeGo 的域视图

MeeGo 嵌入式操作系统的域视图及各域的具体内容如图 2 所示^[3], 共包括 12 个域: 安全域、系统域、个人信息管理域、软件管理域、数据管理域、多媒体域、定位服务域、通信域、内核域、核心组件域、Qt 域和图形域。



图2 MeeGo 系统域视图

内核域是应用程序和硬件层之间的桥梁, 支持 MeeGo 的硬件平台需要提供三类相关组件(内核驱动、核心架构组件、内核配置文件)。内核驱动是一套需要与硬件进行交互的驱动程序, 例如 MeeGo notebook 1.1 中集成了摄像头、USB、声卡、显卡、蓝牙及触摸屏等硬件驱动程序; 核心架构组件是一个额外的软件包, 将它附加到内核中以支持某些硬件的相关功能(如固件); 内核配置文件为特定设备提供了详细的配置信息, 包括设备架构、处理器类型、设备驱动配置菜单、网络支持以及内核调制选项等。

Qt 域提供了跨平台开发工具(Qt、Qt Mobility、Qt Webkit、Qt WebRuntime), 其中 Qt 提供了应用程序和用户界面的

开发工具包, Qt Mobility 为 MeeGo 提供了开发移动应用程序的接口, Qt WebKit 能够快速创建包含实时网络内容和服务的应用程序。

2 搭建开发环境

2.1 Linux 下安装配置 Qt Creator

在 Linux 和 Windows 环境下均可以开发 MeeGo 应用程序。鉴于 MeeGo 操作系统是基于 Linux 内核并且是完全开源的, 故选择了在 Ubuntu 10.04 LTS 系统上搭建 MeeGo 应用程序的开发环境。

为搭建 MeeGo 应用程序开发环境, 在 Ubuntu 10.04 LTS 上安装配置 QtCreator 需要 4 个步骤, 如图 3 所示。



图3 Ubuntu 10.04 LTS 安装配置 Qt Creator 流程图

(1) 配置软件库安装源

① 添加软件源:

```
deb http://repo.meego.com/MeeGo/sdk/host/repos/ubuntu/10.04//
```

② 为软件仓库增加公钥:

```
$ gpg --keyserver pgpkeys.mit.edu --recv 0BC7BEC479FC1F8A
$ gpg --export --armor 0BC7BEC479FC1F8A | sudo apt-key add -
```

③ 更新软件源:

```
$ sudo apt-get update
```

④ 测试软件源是否正确安装:

```
$ apt-cache policy madde
```

(2) 安装 MeeGo SDK

```
$ sudo apt-get install meego-sdk-ia32
```

```
$ sudo apt-get install meego-sdk-armv7l
```

如果开发基于 x86 架构设备的应用程序, 相应的 SDK 需要安装 meego-sdk-ia32; 如果开发基于 ARM 架构设备的应用程序, 则相应的 SDK 需要安装 meego-sdk-armv7l。

(3) 安装 MeeGo 目标机交叉编译环境

① 下载并安装目标机交叉编译环境的压缩文件包:

```
$ sudo mad-admin create -f meego-netbook-ia32-1.1.2
```

② 检查目标机交叉编译环境是否正确安装:

```
$ mad -t meego-netbook-ia32-1.1.2 pscreate -t qt-simple qthello
```

```
$ cd qthello
```

```
$ mad -t meego-netbook-ia32-1.1.2 qmake
```

```
$ mad -t meego-netbook-ia32-1.1.2 make
```

③ 检查目标机交叉编译环境是否正确运行:

```
$ file build/qthello
```

(4) 配置 Qt Creator

添加 MeeGo 交叉编译器, 打开 Qt Creator, 选择 Tools > Options > Qt4 > Qt Versions。点击“添加”按钮, 路径选

择为 `usr/lib/madde/linux -i686/targets/meego -netbook -ia32-1.1.2/bin/qmake`, 点击 Rebuild 按钮。如果不进行交叉编译, 则可以添加 MeeGo SDK 的 Qt 版本, 步骤同上, 但路径选择为 `/opt/meego/meego-sdk-qt/bin/qmake`。

2.2 搭建基于 Atom D510 嵌入式实验平台的开发环境

Atom D510 处理器针对 MeeGo 操作系统进行了优化, 在 Atom D510 嵌入式实验平台下搭建 MeeGo 应用开发环境需要 4 个步骤, 如图 4 所示。

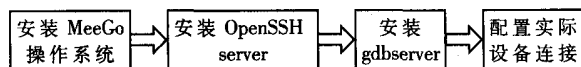


图 4 搭建开发环境流程图

(1) 在 Atom D510 平台上安装 MeeGo 上网本操作系统

由于 Atom D510 嵌入式实验平台提供了 UBS 接口, 最简单快捷的安装方式是使用 U 盘安装。首先, 下载 U 盘镜像制作工具 Win32DiskImager.exe 和 MeeGo 上网本镜像文件; 其次, 制作 MeeGo 上网本系统启动 U 盘; 最后, 用该 U 盘重启实验平台, 进入 MeeGo 上网本系统安装界面进行安装。

(2) 安装 OpenSSH sever

① 安装 OpenSSH sever。从主机 Qt Creator 向远程实体设备部署程序时, 需要进行文件拷贝, 因此需要在实体设备上安装 OpenSSH 服务。

```
$ sudo zypper install openssh-server
```

② 启动 SSH 服务:

```
$ sudo /etc/init.d/sshd start
```

③ 将 OpenSSH 加入初始化序列中, 开机自动启动:

```
$ sudo chkconfig --add sshd
```

(3) 安装 gdbserver

如果进行远程调试应用程序, 需要安装 gdbserver。

```
$ sudo zypper install gdb-gdbserver
```

(4) 配置实际设备连接

打开 Qt Creator, 选择 Tools → Options → Projects → MeeGo Device Configurations。点击“添加”按钮, 添加一个程序运行的实际设备。设置 Device type 为 Real Device、Authentication type 为 Password、Username 为 root、Password 为 meego。SSH 使用默认端口号 22, Gdb Server 使用默认端口号 10000(如果该端口被占用可使用其他端口)。

除此之外, 如果开发主机与实际设备使用交叉网线进行通信, 则两者的 IP 地址必须配置在同一网段内。如果使用局域网进行通信, 则开发主机与实际设备接入同一局域网即可。

2.3 QEMU 虚拟机下搭建开发环境

在 QEMU 虚拟机下搭建开发环境, 需要安装 MeeGo 运行时(即运行在 QEMU 中的 MeeGo 操作系统镜像)。安装配置 QEMU 运行时需要 4 个步骤, 如图 5 所示。

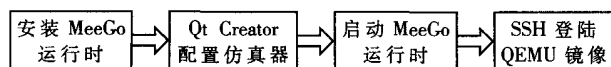


图 5 安装配置 QEMU 运行时流程图

(1) 安装 MeeGo 运行时

```
$ sudo mad-admin create -f -e meego-netbook-ia32-qemu-1.1.2-runtime
```

如果运行时安装成功, 则执行下述指令:

```
$ sudo mad-admin list runtimes
```

屏幕将显示 meego-netbook-ia32-qemu-1.1.2-runtime (installed) 安装 MeeGo SDK。

(2) 在 Qt Creator 中配置 MeeGo 仿真器

打开 Qt Creator, 选择 Tools → Options → Projects → MeeGo Device Configurations。点击“添加”按钮, 为仿真器添加一个 runtime(运行时)。设置 Device type 为 MeeGo Emulator、Authentication type 为 Password、Username 为 root、Password 为 meego。SSH 使用默认端口号为 6666。Gdb Server 使用默认端口号为 13619(如果该端口被占用可使用其他端口)。

(3) 启动 MeeGo 运行时

```
$ sudo mad remote -r meego-netbook-ia32-qemu-1.1.2-runtime poweron
```

如果运行时不能正常启动, 可能有两个原因: ① 开发主机 BIOS 禁用了 Virtualization Technology, 进入主机 BIOS 开启 Virtualization Technology 即可解决问题; ② 开发主机没有加载 KVM 模块。

对于 Intel 处理器, 执行 `$ sudo modprobe kvm_intel` 指令可加载 KVM 模块

对于 AMD 处理器, 执行 `$ sudo modprobe kvm_amd` 指令可加载 KVM 模块

(4) 用 SSH 登录运行在 QEMU 中的 MeeGo 镜像

```
$ ssh meego@127.0.0.1 -p 6666
```

MADDE 启动 QEMU 时, 完成了从主机 6666 端口到 MeeGo 镜像 SSH 端口的重定向。目前 MeeGo 镜像中有如下两个帐号: ① 用户名: meego 密码: meego; ② 用户名: root 密码: meego。

3 应用程序开发流程

(1) 新建一个 Qt 工程。在 Qt Creator 开始界面, 点击右下角 Creat Project 按钮, 创建 Qt 工程。

(2) 根据目标设备选择合适的应用程序模板。开发 MeeGo 上网本应用程序, 需要选择 Qt C++ Program 工程中的 Mobile Qt Application 模板。

(3) 使用工程向导, 逐步设置工程的名称、路径、Qt 版本信息、程序类信息、工程管理信息。程序类信息包括: 类的名称、基类、头文件、源文件和 Form 文件。工程管理信息列出了工程的路径和工程中所有的文件。

(4) 编写相应的应用程序代码, 设置应用程序运行环境。如果目标设备为 MeeGo 虚拟机, 则将 Device configuration 设置成 meego emulator; 否则将其设置为实体设备。设置完成后, 依次点击编译和运行按钮, 应用程序即可部署到目标设备上并运行。

4 应用程序集的设计与开发

4.1 总体设计方案

应用程序集总体结构如图 6 所示, 由主操作界面、办公模块、通信模块、娱乐模块、地图模块组成。通过主操作界面可启动不同子模块, 子模块之间不能直接互相切换, 子模块必须返回主操作界面才可启动其他子模块。通过子模块界面可启动相应的应用程序。

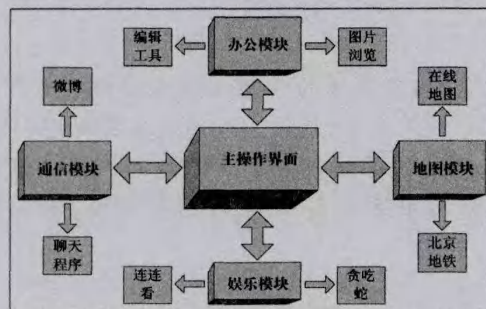


图 6 应用程序集总体结构框图

4.2 设计

(1) 主操作界面的设计与实现

主操作界面使用 OpenGL 绘制了具有 3D 特效、可旋转缩放六面体的六面体。六面体上下两个面的纹理设置为 MeeGo 的 LOGO 图片, 前后左右 4 个面分别代表了办公模块、通信模块、娱乐模块和地图模块。通过鼠标双击, 即可进入相应的子模块。主操作界面的立体和展开示意图如图 7 所示。

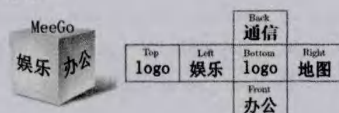


图 7 主操作界面的立体和展开示意图

进入子模块后, 主操作界面切换为子模块的操作界面。两个模块的主操作界面的结构完全相同, 只是六面体每个面的功能不同。以办公模块为例, 其上下两个面的功能为返回主操作界面, 前后左右 4 个面分别代表了文本编辑程序、表格处理程序、图片浏览程序和保留的扩展程序接口。

六面体的每个面均使用 OpenGL 逆时针正向绘制。六面体通过加载相应的纹理代表不同的模块或者应用程序。首先加载用来生成纹理的图片, 然后生成纹理, 最后加载纹理^[4]。在绘制面之前调用 `glLoadName()` 函数, 使用名字栈为六面体的面设置名字。通过调用 `faceAtPosition()` 私有函数确定光标下的面。程序使用面的名字和 `shift_flag` 变量来实现主操作界面和子模块操作界面的相互切换。

(2) 子模块的设计与实现

办公模块设计实现了图片浏览程序、文本编辑程序、表格处理程序; 通信模块设计实现了新浪微博客户端和局域网通信程序; 娱乐模块设计实现了连连看游

戏、贪吃蛇游戏和拼图游戏; 地图模块设计实现了北京地铁线路查询程序和 Google 在线地图程序。

在此只介绍地图模块中的北京地铁线路查询程序。该程序涵盖了北京地铁七条干线, 并实现了任意两个站点之间的最短路径查询和动态显示。该程序使用图存储地铁线路信息。地铁线路图中共有 115 个站点, 每个站点分配了唯一的 ID 标识为图中的一个顶点。如果两个站点直接相连, 则在这两个站点之间弧的权值设置为 1, 否则设置为 10 000, 表示这两个站点没有直接相连。程序使用著名的弗洛伊德最短路径算法, 通过图的权值矩阵求任意两个站点之间的最短路径。通过定时器和变换画刷的颜色来实现查询结果的动态显示, 每 0.7 s 更新一次地图, 如果用户正在进行线路查询, 则变换画刷的颜色闪动线路上站点。

4.3 应用程序设计举例

以 Google 在线地图为例, 本应用程序使用 MeeGo Qt 域的 Qt WebKit 进行开发, 总体的开发流程与本文第 3 节“应用程序开发流程”相同。在此, 只叙述程序代码的编写。

(1) 为了使用 Qt WebKit, 需要在工程的 .pro 文件中添加 `QT += webkit`, 并在相应头文件中添加 `#include <QWebView>`。

(2) 新建一个 QWebView 类的控件对象:

```
QWebView *mapview = new QWebView();
```

(3) 调用该控件对象的 `setUrl` 函数, 设置其显示内容为 Google 在线地图:

```
mapview->setUrl(QUrl("http://ditu.google.cn/"));
```

(4) 设置控件的窗口名字并将其显示:

```
mapview->setWindowTitle("Google 在线地图");
```

```
mapview->show();
```

本文论述了 MeeGo 嵌入式操作系统的运行机制和应用程序的开发流程, 以及在 Linux 平台上搭建 MeeGo 应用程序开发环境和使用 Qt Creator 开发应用程序集的方法和步骤。

参考文献

- [1] 英特尔开源技术中心. MeeGo™ 开源项目白皮书[Z]. 2010.
- [2] The Linux foundation. MeeGo architecture layer view [EB/OL]. [2010-10-20]. <https://www.meeGo.com/developers/meego-architecture/meego-architecture-layer-view.htm>.
- [3] The Linux foundation. MeeGo architecture domain view [EB/OL]. [2010-10-20]. <https://www.meeGo.com/developers/meego-architecture/meego-architecture-domain-view.htm>.
- [4] WRIGHT R, LIPCHAK B. OpenGL 超级宝典 (第四版) [M]. 张琪, 付飞, 译. 北京: 人民邮电出版社, 2010.

(收稿日期: 2012-04-29)

作者简介:

韩德强, 男, 1966 年生, 高级工程师, 硕士生导师, 主要研究方向: 嵌入式系统。