



3.1 Linux 和嵌入式 Linux 简介

嵌入式操作系统是嵌入式系统的重要组成部分,具备通用操作系统的基本特点,它是嵌入式系统实现多样化应用的基础。嵌入式操作系统由常见的通用操作系统衍生而来,它们具有密不可分的联系,嵌入式操作系统具有通用操作系统的一般特性以及类似的体系结构。目前应用最广泛的通用操作系统是采用商业授权的 Windows 操作系统和开源的 Linux 操作系统,它们均有相应的嵌入式版本: Windows CE 和嵌入式 Linux。

嵌入式 Linux 是对 Linux 内核进行裁剪、定制、重新编译和封装之后得到的适用于嵌入式系统的操作系统。由于 Linux 开源、免费等特性,世界各地的开发人员开发出了数量众多、特点各异的基于 Linux 的嵌入式操作系统。嵌入式 Linux 继承了 Linux 的诸多优良特性,并已经成为目前应用最为广泛的嵌入式操作系统之一。

3.1.1 Linux 的起源与发展

Linux 是一套不断完善、可以免费使用和自由传播的类 UNIX 开源操作系统,它符合 POSIX(Portable Operating System Interface of UNIX,可移植操作系统接口)标准,并且与 UNIX 兼容,具有 UNIX 的许多优良特性。Linux 作为一个典型的现代网络型操作系统,其中所涉及的技术实现涵盖了操作系统技术的最新成果。Linux 是一个多用户多任务操作系统,支持分时处理和软实时处理,并带有微内核特征(如模块加载/卸载机制),具有很好的定制特性。由于它是开放源代码的,全世界很多科学技术人员在对它进行不断完善的同时,还增加了越来越多的新功能,比如说支持硬实时任务处理等。

Linux 的起源和发展是与 UNIX、GNU 和 Internet 三个要素紧密联系在一起^[1]。UNIX 操作系统是美国 AT&T 公司于 1969 年开始进行开

发的操作系统,具有多用户、多任务的特点,采用商业授权。1987 年 Andrew S. Tanenbaum 开发了基于微内核架构的类 UNIX 计算机操作系统 MINIX,用于大学的教学和科研工作并开放全部源代码。1991 年 4 月,芬兰赫尔辛基大学的学生 Linus Benedict Torvalds 对教学使用的 MINIX 操作系统不甚满意,出于个人爱好,他开发了操作系统核心 Linux 0.01,但没有使用 UNIX 和 MINIX 的任何源代码。Linus Torvalds 宣布这是一个免费的操作系统,任何人都可以从 FTP 上免费得到它的源代码,并希望大家共同完善 Linux。于是,当今世界最著名的开源操作系统就诞生了^[2]。

GNU 是 GNU's Not Unix 的缩写,GNU 计划开始于 1983 年,其目标是开发一个免费、自由、完整的类 UNIX 操作系统,包括操作系统内核、软件开发工具和各种应用软件。到 1991 年,软件开发工具和各种必备应用软件的开发已经基本完成,但是操作系统内核的开发却进展缓慢。Linux 的诞生为解决 GNU 计划的困局提供了可能。经过 Linus Torvalds 和其他开发人员的努力,GNU 计划的应用软件逐渐被移植到 Linux 上。到了 1992 年,Linux 内核与 GNU 的其他软件完成了兼容,免费、自由、完整的操作系统正式诞生了。

之后伴随着 Internet 在全球的兴起,借助 Internet 的便利条件,世界各地的开发人员都可以方便地参与到 GNU 计划的开发和 Linux 内核的改进工作中来,于是 GNU 计划得到了广泛的支持和蓬勃的发展,Linux 操作系统也迅速发展壮大起来。截至目前,Linux 已经成为世界上最著名的开源操作系统,并且是唯一能与微软公司的 Windows 抗衡的操作系统。

Linux 作为一个现代操作系统的典型实现,可以说是一个计算机业与时俱进的产物,它不断更新,不断完善,其新功能的加入和完善速度超过了现今世界任何一种操作系统。功能的不断增加和完善,灵活多样的实现,可定制的特性、开放源码等,使得它的应用日益广泛,大到服务器和计算机集群,小到智能手机和控制器,可以说是无处不在。

Linux 在嵌入式系统应用方面尤其显示出其优越性。Linux 强大的网络功能将赋予嵌入式系统对网络的天然亲和力,从而为嵌入式系统的网络互连和功能扩展提供了非常广阔的发展空间。

3.1.2 Linux 内核

计算机首先是一个信息处理工具,其核心硬件就是处理器和内存;其次,计算机也是一个信息存储工具,其核心硬件是各类永久存储设备如硬盘等;同时,计算机也是一个信息交换工具,其核心硬件是各类通信 I/O 设备,如串口,网络适配器等。由此可见,一个比较完整的计算机硬件系统包含了信息处理、信息存储和信息交换三大基本功能,而所有这些功能实现的基础就是程序和数据。操作系统需要研究程序运行和数据与数据的存储和传输。

在冯·诺依曼计算机体系中,程序必须装入到内存中之后才能运行。这就涉及一个程序在内存的存放和管理问题,从而需要引入内存管理机制;多个程序在运行中常常需要同步和协调以保证计算逻辑的正确性,使得进程管理机制成为必需;如何调度各个程序从而保证资源的最大限度利用是调度机制研究的重点;计算机发展至今,I/O 设备种类繁多,如何才能有效管理这些设备是 I/O 设备管理研究的重要内容。内存管理,进程管理,调度和 I/O 设备管理等机制就构成了操作系统的核心,即所谓的操作系统内核。

严格地说,Linux 这个单词只代表 Linux 内核,它是指由 Linus Torvalds 维护的操作系统内核,主要通过 <http://www.kernel.org> 和镜像网站发布。厂商、个人或者其他组织可以对 Linux 内核和常用应用软件进行封装,这样就形成了数量众多的发行版。发行版可以是免费的,也可以是付费的,目前常见的发行版有 Ubuntu、Debian、RedHat、SUSE、红旗等^[3]。

3.1.3 Linux 2.6 内核新特性

Linux 内核的升级主要通过版本号来体现。在 2.6 内核之前,内核的版本号采用通过“.”分隔开的三个数字来标示,中间数字为奇数的内核表示开发中的内核,偶数则表示稳定的内核^[3, 4],如 Linux 2.4.20,其中 2.4 是版本号,20 是发行号。从 2.6 内核开始,命名规则发生了重大变化,例如 2.6.10 和 2.6.11 之间的区别可能会很大,同时版本号采用通过“.”分隔开的四个数字来标示,当内核变化不大时利用新增加的第四个数字进行区分。截至 2011 年 3 月,最新的稳定内核版本号为 2.6.38.2。

从 2003 年 12 月 Linux 2.6 内核发布至今,2.6 内核仍然处于开发之中。相比于 2.4 内核,2.6 内核的改进主要有^[1]:

(1) 采用了新的进程管理算法,在高负荷的情况下算法执行效果非常出色,并且在多处理器的情况下能够很好地进行扩展。

(2) 在 2.6 内核中,内存可以被抢占,这样做的好处是显著提高了操作系统的实时性,降低了用户交互、多媒体应用等程序的延迟。这项改进对嵌入式系统来说是非常有利的。

(3) 融合了反向映射技术(Reverse Mapping),在一定程度的负荷下能够显著改善虚拟内存的性能。

(4) 改进了内核编译系统,加快了编译速度,增加了改进的图形化工具 `make xconfig`(需要 Qt 库)和 `make gconfig`(需要 GTK 库)。

3.1.4 Linux Shell

目前,Linux 桌面发行版已经能够提供比较完善的图形用户界面,但是在嵌入式开发中,掌握命令行式 Shell 的使用方法和技巧仍然是很有必要的。Shell 是用户与

Linux 内核交互的界面和接口,它是一个命令解释器,接收用户输入的命令并把它送到内核中去执行。命令行式 Shell 在 Linux 操作系统中的地位非常重要,它可以提供比图形用户界面更全面、更强大的功能,Linux 操作系统中的所有操作都可以作为 Shell 命令来执行^[5]。

Linux 有多种命令行式 Shell,其中默认使用的是 BASH(Bourne Again SHell),在绝大多数情况下 Linux 的命令行式 Shell 指的就是 BASH。BASH 提供了完善的命令控制机制,功能强大并且使用方便^[5],它也是目前使用最广泛的 Linux Shell。嵌入式系统开发经常用的 Linux Shell 命令如表 3.1 所示。

表 3.1 常用的 Linux Shell 命令

命 令	功 能	常 用 参 数
用 户 管 理		
login	向系统注册,也可进行用户间切换	
su	类似 login,可进行用户间切换	
logout	注销	
exit	非超级用户需要先用 exit 命令退出操作环境,再用 root 用户登录,才能执行关闭操作	
who	查看当前已登录的用户	
useradd/adduser	使用 root 用户登录时可以用于添加用户	
passwd	修改密码	
进 程 管 理		
ps	查看系统进程	-a: 显示终端上的所有进程
kill	向进程发送信息,可用于终止进程	-9 t_pid: 终止进程(t_pid 表示进程的 pid,可以通过 ps 命令查看)
文 件 系 统		
ls	显示指定工作目录下的文件	-l: 列出目录下各文件的详细信息
mount	挂载一个文件系统	
umount	卸载一个文件系统	
pwd	查看当前目录的绝对路径	
cd	改变当前目录	
mkdir	创建目录	
rm	删除文件及目录	
cat	把命令行给出的各个文件逐个打印到标准输出	
cp	将一个文件复制到另一个文件或目录	
mv	将一个文件移动到另一个文件或目录	
file	查看文件的属性	
chmod	修改文件存取权限	

续表

命 令	功 能	常 用 参 数
网 络 系 统		
ping	查看网络上主机是否在工作	
lftp	标准的文件传输协议的用户接口	lftp 主机名/IP
磁 盘 管 理		
df	查看文件系统磁盘空间占用情况	-h
dh	显示文件及目录占用磁盘空间情况	-l: 对当前目录下每个目录及文件名只给出占用的数据块总数 -a: 递归显示指定目录及子目录中各文件所占用的数据块数 -h: 以容易辨认的方式显示
其 他 命 令		
echo	把命令行给出的字符串输出到标准输出	
man	查看联机帮助	
uname	查看当前系统信息	-a
vim	启动 vim 编辑器,编辑文件	

3.2 Linux 文件系统

3.2.1 传统文件系统

ext2 是专门为 Linux 设计的一种文件系统,具有较高的效率,其优点是存取文件的性能很好,拥有极快的速度和极小的 CPU 占用率。ext2 既可以用于标准的块设备,也被应用在软盘等移动存储设备上。

ext3 向下兼容 ext2,是一种日志式文件系统(Journal File System,JFS),其基本特点是会将整个磁盘的写入动作完整地记录在磁盘某个区域上,以便有需要时进行回溯。当某个过程中断时,系统可以根据记录的日志直接回溯并重整被中断的部分,重整速度相当快。目前 ext3 已经成为 Linux 中使用最广泛的文件系统^[6]。

ext4 是 ext3 的升级,从多个方面提高了性能,例如,ext3 支持最大 16TB 的文件系统和最大 2TB 的单个文件,而 ext4 增加了 48 位块地址,支持最大 1EB 的文件系统和最大 16TB 的单个文件;ext3 每个目录最多可以包含 32 000 个子目录,而 ext4 每个目录最多可以包含 64 000 个子目录,在特定情况下数量可以更多^[7, 8]。

btrfs 被认为是下一代的 Linux 文件系统,其基本特点是采用了 COW 技术(Copy on Write),这样 btrfs 在普通操作中不会直接覆盖数据,而是将元数据(metadata)和数据的新值写到别的地方,然后在文件系统中指向新的位置。COW 的优点在于避免了对同一个物理单元的反复写操作^[9],同时保证了一致性和完整性。

btrfs 还提供了快照、加密等功能^[8, 10, 11]。

此外, Linux 还支持 FAT32、NTFS 等 Windows 操作系统常用的文件系统。

3.2.2 Flash 文件系统

Flash 存储器具有容量大、掉电后数据不丢失以及可多次擦写等优点, 目前已成为嵌入式系统中广泛使用的存储设备。常见的 Flash 存储器有 NOR Flash 和 NAND Flash 两种, 两者具有相同的工作原理和存储单元。NOR Flash 的读性能好, 但写入和擦除的速度很慢, 并且容量较小; NAND Flash 同时具备较好的写擦性能和大容量的优点, 但输入输出接口和控制电路更为复杂^[3]。Flash 存储器内部分为多个存储单元块(block), 存储单元块是可擦除的最小单位; 每个存储单元块又由多个页(page)组成, 页是写入数据的最小单位。除了常规需求外, Flash 存储器的特性对文件系统提出了新的要求, 主要有以下几点^[12]:

(1) 由于 Flash 存储器在使用前和使用过程中都会出现坏块, 因此 Flash 文件系统需要能够对坏块进行妥善的管理。

(2) Flash 存储器的使用寿命是一定的, 一般可擦除的次数为 10 万~100 万次, 因此一方面需要平衡各块地址的操作频率, 避免频繁地对同一地址进行操作, 以免造成局部单元提前损坏; 另一方面由于擦除操作耗时较多, 应减少擦除操作, 尽量保持擦写次数均衡。

常见的 Flash 文件系统及其特性如表 3.2 所示。

表 3.2 常见的 Flash 文件系统

文 件 系 统	特 性
romfs	只读文件系统, 使用顺序存储方式。具有占用内存少、体积小、可靠性好、读取速度快等优点, 嵌入式 Linux 系统中常用作根文件系统
jffs/jffs2	jffs 是针对 Flash 存储器的特性为嵌入式设备开发的日志式文件, 系统提供了负载平衡、坏块管理等功能。jffs2 是由 RedHat 公司开发的, 改进了 jffs 的一些缺点, 提高了 Flash 利用率, 降低了 Flash 损耗, 并支持数据压缩等
yaffs/yaffs2	专门为 NAND Flash 设计的日志式文件系统, 可读写, 完全开源, 针对启动时间和 RAM 的使用进行了优化, 适用于大容量的存储设备
cramfs	只读压缩文件系统, 采用 zlib 压缩, 其容量上限为 256M。采用实时解压缩方式, 但解压缩的时候有延迟。嵌入式 Linux 系统中常用作根文件系统

3.2.3 RAM 文件系统

在 Linux 内核中, 虚拟内存(VM)资源由物理内存(RAM)和交换分区.swap)共

同组成,内核中的虚拟内存子系统负责分配和管理 VM 资源。

tmpfs 是一个虚拟内存文件系统,可以只使用物理内存,也可以同时使用物理内存和交换分区。tmpfs 不同于采用块设备形式实现的 ramdisk,也不同于针对物理内存设计的 ramfs,其优点主要有^[13]:

(1) tmpfs 文件系统驱动程序可以动态地对文件系统大小进行调整,在适当的时候自动减小文件系统并释放虚拟内存资源,这样操作系统的其他组件就可以使用这部分虚拟内存资源了。

(2) tmpfs 文件系统可以完全驻留在物理内存 RAM 中,因此读写速度极快,几乎是瞬时完成的。在实际应用中即便使用了一些交换分区,在经过调整后读写性能仍然可以很高。

3.2.4 网络文件系统

在嵌入式系统的开发和调试阶段还经常会用到网络文件系统。经过不断的发展,网络文件系统已经成为分布式文件系统的基础。

NFS 是由 Sun 公司开发的允许通过网络在不同机器、不同操作系统之间共享文件的技术,它是第一个现代网络文件系统,截至 2011 年年初的最新版本是 NFS V4.1。NFS 基于 TCP/IP,一般由一台服务器和数台客户机组成。客户机上的程序通过网络可以像操作本地文件一样方便地操作远程服务器上的文件。NFS 可以使嵌入式系统的开发和调试变得更加方便。利用 NFS 技术在主机上建立基于 NFS 的根文件系统,再挂载到嵌入式设备,这样就可以方便地直接修改嵌入式设备上的根文件系统内容,免去了来回拷贝文件的烦恼。

Ceph 是正在研发和测试的最新的分布式文件系统之一,Linux 也提供了对 Ceph 的支持。Ceph 是 PB 级的容错分布式文件系统,符合 POSIX 标准,它可以扩展到数 PB 的容量,并具有很高的可靠性。

3.2.5 虚拟文件系统

Linux 的一大特色是支持的文件系统种类很多,包括前面提到的 ext3、ext4、btrfs、romfs、jffs2、tmpfs 等,并且不断地有新的文件系统得到支持。每一种文件系统都有自己独特的组织结构和操作过程,这样就给统一管理带来了麻烦。为了实现所有的文件系统采用统一的文件管理框架,使得用户可以不关心各种文件系统的具体操作过程,Linux 引入了虚拟文件系统(Virtual File System, VFS)对这些文件系统进行统一管理。

VFS 是一个内核软件层,用来管理挂载各种文件系统。VFS 只存在于内存中,它在系统启动时创建,在系统关闭时注销。VFS 的作用就是对各类文件系统作进一步抽象,最终实现各类文件系统展现在用户面前的是一个统一的操作界面,并且能够提供一

个统一的应用编程接口。所有的文件系统被设计成可加载模块,在系统需要时进行加载,这样用户只需要面对 VFS 的操作接口就可以完成对各种不同文件系统的操作。

devfs(设备文件系统)是由 Linux 2.4 内核引入的,它使得设备驱动程序能够自主地管理设备文件。devfs 运行在内核空间,挂载在/dev 目录下,提供了一种类似于文件管理的方法来管理位于/dev 目录下的所有设备。设备初始化时可以在/dev 目录下创建设备文件,卸载设备时则将其删除。/dev 目录下的每一个文件都对应一个设备。devfs 文件系统的缺点主要有:某个设备映射的设备文件是不确定的,主/辅设备号有数量限制,设备文件命名方式不灵活等^[1]。

为了克服 devfs 的上述缺点, Linux 2.6 内核引入了 sysfs 虚拟文件系统,它挂载在/sys 目录下。sysfs 可以产生一个包括所有硬件设备的层级视图,与提供进程和状态信息的 proc 文件系统非常类似(proc 文件系统只存在于内存中,以文件系统的方式为访问操作系统内核数据的操作提供接口)。sysfs 把连接到系统上的设备和总线组织成一个分级的文件,用户空间的程序可以利用这些信息实现和系统内核的交互。Linux 2.6 内核同时提供了 udev 来取代 devfs。udev 运行在用户空间,利用 sysfs 提供的信息来实现 devfs 的所有功能,不存在 devfs 一些先天的缺陷^[1]。

3.3 Linux 图形界面

从用户角度来看,操作系统更重要的是用户界面。随着计算机硬件技术的飞速发展,计算机的计算能力在应付普通应用方面逐渐显得绰绰有余,在这种情况下,人们更关心的是人机交互的便利性、友好性等方面,并且使得用户界面越来越成为任何一种操作系统不可或缺的一部分。

图形用户界面(Graphical User Interface, GUI)是迄今为止计算机系统中最为成熟的人机交互技术。一个好的图形用户界面的设计不仅要考虑到具体硬件环境的限制,而且还要考虑用户的喜好习惯等。图形用户界面的引入主要是从用户角度出发的,因此用户自身的主观感受对图形用户界面的评价占了很大比重,比如说,易用性、直观性、友好性等。另外,从纯技术的角度看,仍然也会有一些标准需要考虑,比如说,跨平台性、对硬件的要求等。在嵌入式系统开发和应用中,我们所考虑的问题主要还是集中在图形用户界面对硬件的要求以及对硬件类型的敏感性方面,提供给用户的最终界面还是以满足简单实用的要求为主要目的。

虽然不同的 GUI 系统因为其使用场合或服务目的的不同,具体实现互有差异,但是总结起来,一般在逻辑上可以分为五个模块,如图 3.1 所示:底层 I/O 设备驱动(显示设备驱动、鼠标驱动、键盘驱动等)、基本图形引擎(画点、画线、区域填充)、消息驱动机制(消息处理流程)、高层图形引擎(画窗口、画按钮)以及 GUI 应用程序接口(API)。另外,为了实现 GUI 系统,一般需要用到操作系统内核提供的功能,如线程机制、进程管理。当然,不可避免地需要用到内存管理、I/O 管理,甚至还可能有文件管理。

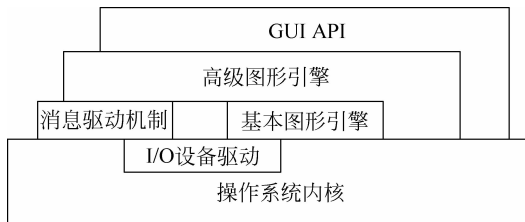


图 3.1 GUI 的一般架构

Linux 使用 X Window System 作为 GUI 协议, Linux 桌面发行版上最为流行的两种图形工作环境: GNOME 和 KDE 就是基于 X Window System 构建的。接下来我们将简单介绍 X Window System 的基本原理, 然后介绍嵌入式 Linux 系统常用的 The Nano-X Window System、MiniGUI、Qt 等图形用户界面系统。

3.3.1 X Window System

X Window System, 也称为 X11 或 X, 是 UNIX 及类 UNIX 操作系统适用的一种标准化软件工具包及显示架构的运作协议, 它通过软件工具及架构协议来建立操作系统所用的图形用户界面。

X 是一个客户/服务器型的视窗系统(如图 3.2 所示)。和其他视窗系统一样, 你可以在它的视窗(Display, 一套键盘、鼠标及显示设备)上同时运行多个应用程序, 而每个应用程序都拥有自己的一个主窗口。应用程序并不是直接和视窗打交道, X 的视窗是由 X 服务器(X Server)控制, 应用程序要显示什么内容只需通过某个指定的通信通道向 X 服务器发出申请就行了。除了处理应用程序的显示请求之外, X 服务器还负责接收鼠标、键盘事件, 并将它们送到相应的应用程序。这里, 应用程序被称为 X 客户(X Client)。

注意这里的通信通道可以是网络也可以是进程间通信常常提及的共享内存。因此, X 系统的客户和服务端可以在同一台机器上, 也可以不在同一台机器上。一个应用程序不管是在本地运行, 还是在远程某台机器上运行, 都可以实现在本地显示。X 系统的这种特性即所谓的网络透明。

需要注意的是 X 服务器主要提供视窗服务, 即接收键盘鼠标等输入设备事件并将它们传送给对应的应用程序, 同时接收并处理应用程序发出的显示请求, 而对诸如按钮、窗口这样的用户界面并没有作定义或者有相应的实现, 这些工作都留给了 X 客户程序自行完成。

在 X Window 系统中, 用户界面进一步被分成两个部分, 一部分是窗口管理器(Window Manager), 它负责管理你的桌面环境, 包括窗口的移动、缩放、创建和注销等; 另外一部分是应用程序界面, 这部分的功用包括决定标题栏的显示方式等, 它最终决定你的应用程序的观感。通常来讲, 对于一个 X Window 系统而言, 总是有一个

独立的程序来负责实现窗口管理器的功能(见图 3.2)。由于应用程序界面是内置于应用程序的,因此在同一个显示屏上会有不同风格的界面类型出现。

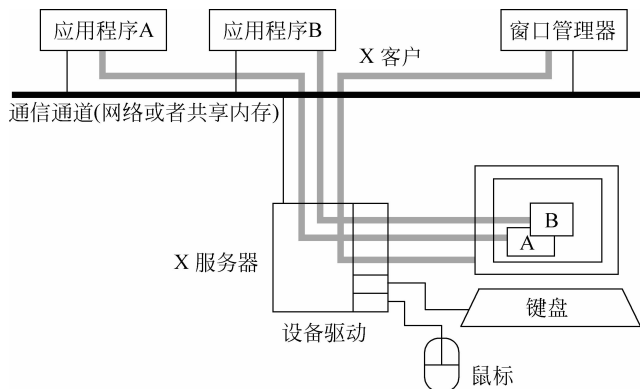


图 3.2 X Window System 一般架构

3.3.2 The Nano-X Window System

The Nano-X Window System 原名 Microwindows,由于和微软公司的注册商标 Windows 冲突,因而在 2005 年 1 月改为目前的名字。Nano-X 是一个著名的开源项目,目的是将现有图形视窗系统的特性引入磁盘和内存容量较小的袖珍设备和平台中。Nano-X 不需要任何操作系统或者其他图形系统的支持,可以直接对显示设备进行写入操作。

Nano-X 设计的非常小巧,可以运行在多种硬件和软件环境上,它可以运行在 32 位有内核帧缓冲支持的 Linux 系统上。Nano-X 的架构分为:驱动层(driver)、引擎层(engine)和应用层(API),这样的分层结构有利于开发者根据实际情况对代码进行层次化的修改。

截至 2010 年 12 月,Nano-X 能够下载的最新版本号为 0.92,更多详细的信息可以浏览其官方主页:<http://www.microwindows.org/>。

3.3.3 MiniGUI

MiniGUI 是一款面向嵌入式系统的图形界面支持系统。1998 年底,魏永明先生开始开发 MiniGUI。2002 年,魏永明先生成立了北京飞漫软件技术有限公司,继续提供开源版本的同时开始为 MiniGUI 提供商业技术支持。最后一个采用 GPL 授权的 MiniGUI 版本是 1.6.10,从 2.0.4 开始,MiniGUI 的代码被重写并使用商业授权。

MiniGUI 对中文的支持很好,并且非常小巧。目前 MiniGUI 已经广泛应用于通信、医疗、工控、电力、机顶盒、多媒体终端等领域。MiniGUI 在 TD-SCDMA 手机中