ARMOS

ArmOS is a full-featured, prioritized, multitasking, message based, real-time operating system which provides a comprehensive set of system services to the tasks which operate under it. These include process management services, memory management services, interprocess messaging services, timer services, interrupt management services, console services, Networking services, Mutex services, Device services, Storage services, System control services and more. The OS also provides a facility for installable services, so that extensions can be provided which integrate seemlessly with the OS API.
An integrated debugger is incorporated into the OS which provides an isolated context to troubleshoot processor exceptions, memory protection violations, software panics and general softare debugging. The debugger includes standard debug commands (memory examine/modify, CPU/register manipulation, disassembly, breakpoint, single-step etc.) as well as well as functions specific to the operation system, including message/pool auditing, stack callback tracing, and commands to dump OS state, timer, console message, mutex & network queues, as well as internal operating system state variables.
There are two build modes: A debug build incorporates extensive error diagnostics, and traps directly to the debugger in the event of an hardware or software exception. A release build disables the more processor intensive error diagnostics, and automatically recovers from exceptions (It can be set to trap to the debugger so that problems can be diagnosed in the field if needed).
ArmOS employs a device independant driver model allowing it to be deployed on vastly different hardware platforms. It is currently used in line of ARM7 and ARM9 based telephone/voip switching products.

FILE SYSTEM

The ArmOS file system is a general purpose file system with a few unique additional capabilities. Although it appears at first glance to be an integral part of ArmOS, it is in fact an independant installed extension/service.
The main features of the ArmOS file system are:

Very efficient real-time profile.

Very flexible device geometry support, allowing it to support any NAND flash device type available cuffrently or in the forseeable future.

Wear leveling to insure erase/write activity is spread over the entire device.

Bad block detection and management.

Standard API functions include: open, close, read, write, seek, delete, mkdir, rmdir, find_first, find_next, modify_dir_entry etc.

Provides both an OS API and a message based interface, including the ability to be efficiently accessed from within an interrupt handler.

Additional/unusual features provided in support of telephony application:

Ability to select error correction on a file by file basis.

Ability to handle an unlimited number of open directories (used for voice mailboxes) with near-zero overhead in access via those directories.

Ability to easily obtain the directory entry corresponding to an open file.

Ability to easily obtain the full path to a given directory index.

Ability to drop unwanted data from the beginning of a file.

A file update notification service to advise an application when a file has been modified.

Presentation of the NOR code flash device as a virtual device allowing firmware updates to be shared and installed over the network file system.

DSOS

Data Switch Operating System is the proprietary multitasking kernel embedded in a line of small office Data PBX switching systems. It's main features are:

Very efficient real-time performance.

Support for up to 32 serial devices running at 19200bps, with fast routing of data between devices.

Ability to hand off processing to "smart" linecards for differing interfaces, and route the data through the same channels as directly attached devices.

Console interface for configuration, as well as status reporting and real time monitoring.

CUBIX

Cubix is a single user operating system that I developed in the 1980s for my 6809 Portable Computer. It's main features are:

ROM based (8K including I/O drivers) for instant access.

Portable, easily implemented on any 6809 system.

Support for virtually any disk drive geometry.

Directory structured file system with protections.

Integral command line interpreter can be involced from with application programs.

Built in scripting language allows "programs" to be written using Cubix commands and utilities as statements.

Over 100 system calls provide a built in library of common I/O, file access and utility functions.

Any device driver (including disks) can be installed or replaced via system call at any time.

Ability to support for background task(s)

Includes many utility programs for the manipulation and diagnostics of files, directories and disks.

Many powerful applications, including:

Powerful screen (window) text editor

6809 Assembler

6809 Debugger

Micro APL interpreter

Micro BASIC intereter

Micro C compiler

Micro FORTH compiler

Intel 8080 simulator with integrated debugger

The "Dunfield 6809" system and it's software is described in more detail on my classic computing website.

DMF

Device Management Facility is a small OS I wrote in the late 1970s for my 8080 based Altair 8800. As the name suggests it was mainly a means of managing the system devices which included the console and several other serial ports, a NorthStar MDS-A disk subsystem and a 9-track tape drive with a handmade controller board.

The main features of DMF are:

Loads into the "orphan" block of memory above the NorthStar disk controller (F000-FFFF) leaving the entire contiguous block below the disk controller (0000-E8FF) available for application use.

User definable I/O drivers with a common interface specification.

Simple (one level directory) file system.

A command interpreter/shell, with several built in commands, and utilities for performing system maintenance and other functions.

Several larger application program including:

Line and visual mode text editors.

8080 Assembler, disassembler and debugger.

BASIC interpreter.

Tape backup utility for transferring files to/from the 9-track drive.

NorthStar DOS emulator - this loads into RAM at the address of NorthStar DOS and translated the NorthStar DOS alls into DMF service requests, allowing NorthStar DOS applications to be run under DMF.

Source : http://mediatoget.blogspot.in/2011/05/operating-systems.html