

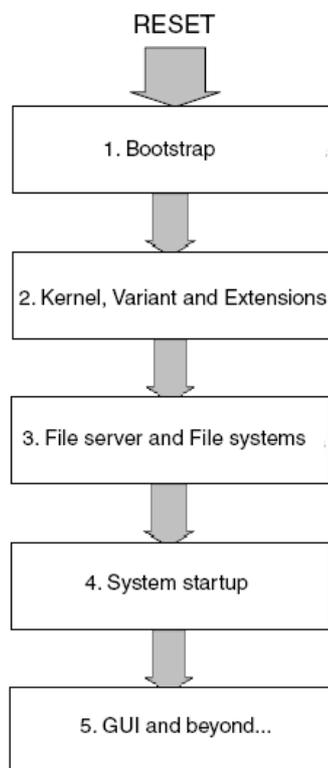
Symbian OS startup

This article describes the Symbian startup sequence, from bootstrapping through to loading the GUI libraries.

Introduction

Bootstrapping process of an operating system is a carefully managed operation. The different stages of initialization must be correctly sequenced to avoid services being initialized before the services they depend on are ready. To successfully initialize the hardware and OS, it is important to know what state the hardware will be in immediately after it has been switched on or reset. For the most part, the OS has to assume that hardware is in an "unknown" state because the boot process may arise from several causes. Only memory essential for reset is visible and RAM will contain garbage. The initialization carried out during the boot process must be robust enough to handle any reason for reset.

High level process



As the OS starts up, the boot process runs through mainly five stages.

- Switching the phone on triggers a hardware reset. The first software that is executed after this is known as the **bootstrap**. On entry to the **bootstrap**, the execution environment is very primitive, and the **bootstrap** is tasked with providing a basic execution environment for the **kernel** process.
- On entry to the **kernel**, the **CPU** is now running at full speed and an execution stack allows typical C++ code to be run. However, there is still only a primitive memory environment and only one execution path. Static data is now initialized and interrupts are masked. Once **kernel** and base support package initialization is complete, there is full management of **CPU**, memory, power and peripherals and the second OS process is started.
- At this stage in the boot there is a fully functional **micro-kernel** that supports multiple multi-threaded, protected user-mode processes. However, the OS has not yet provided the means to instantiate new processes, to extract file-based data from the Flash or to persist data in read/write **Flash Memory**.
- All of the **kernel**, user library and file server services are now fully initialized ready for the rest of the OS to begin its boot process - a job that is given to the system starter process. The system starter manages the initialization of the rest of the OS system services in an ordered manner, and can also provide monitoring and restart for those which terminate unexpectedly. The precise order in which the persistence, communications, multimedia, security and other services are started is controlled by a script and is phone-specific. * Once enough of the system services are running, the primary **GUI** service, the window server, can be started. This now allows initialization of the other UI services and the applications that make up the aspect of the OS that is most evident to the user.

This is a standard sequence for booting the phone. According to this process, the OS services for displaying images or animations, or playing audio become available quite late in the boot. This does not reconcile with the typical experience of using a Symbian phone - in practice phones often use lower level graphics services to display images during boot without demanding that the full screen-sharing capabilities of the window server are available.

