

## COMPONENTIZED OPERATING SYSTEM

## TECHNICAL FIELD

This invention relates to operating systems, and more particularly to a componentized operating system.

## BACKGROUND OF THE INVENTION

Conventional computers generally execute an operating system to manage various aspects of the computer as it is running. Typically, the operating system is responsible for managing access to storage devices as well as input and/or output devices, and controlling the execution of one or more additional applications. Before the operating system can be executed by the computer, it typically must be installed on the computer, a process which usually involves copying multiple files from a distribution medium (e.g., a CD ROM) onto a storage device (e.g., a hard disk) of the computer.

A typical operating system includes a large number of files (currently numbering into the thousands), which can include instructions and/or data. These instructions, when executed by the computer, provide the operating system functionality. With the large number of files also comes a large number of dependencies among files. For example, in order for their intended functionality to be carried out, many files require the installation of one or more other files. Although such dependencies are often known by the manufacturer of the operating system at the time of installation, it can be difficult for a user, administrator, or other software developer to learn about these dependencies. Not knowing about such dependencies can prove troublesome, as software developers may not know what data or instructions in other files affect the particular file(s) they are concerned with, an administrator or user troubleshooting a malfunctioning computer may not know what files are applicable to the problem, etc.

These problems are only exacerbated by post-installation modifications to the operating system. An operating system can be modified in any of a wide variety of manners, such as by adding or replacing one or more particular files, by any of a wide variety of people (e.g., a user, administrator, software developer other than the operating system developer, etc.). When such modifications occur, it increases the difficulty of identifying dependencies among files existing on the computer. Furthermore, it becomes even more difficult to troubleshoot a malfunctioning computer or update the operating system because it is difficult for the user or administrator to know exactly what functionality is (or should be) installed on the computer.

In some systems, the operating system installation and/or updating process(es) display to the user an interface that illustrates the operating system in terms of its functionality (e.g., accessories, games, dial-up networking, etc.). Such interfaces, however, are merely that—user interfaces. Although they provide the user with the ability to select groups of files of the operating system to be installed based on functionality, the operating system itself is still a large collection of files that is not itself componentized and which does not typically identify dependencies among any groups of files.

The componentized operating system described below addresses these and other disadvantages.

## SUMMARY OF THE INVENTION

A componentized operating system is described herein. In certain embodiments, the operating system is separated into multiple different components, each including one or more files, and dependencies among these components are maintained.

According to one aspect, a bill of materials (BOM) identifies at least one component that is installed (or is to be installed) for an operating system image. Each component also has a corresponding manifest that identifies which other components (if any) the component is dependent on. Thus, during an installation or upgrading process, the manifest for a component can be accessed to determine which additional components (if any) are to be installed even if they are not explicitly identified in the BOM.

According to another aspect, an operating system is upgraded based on the current BOM for the operating system and an update BOM indicating what the updated operating system should be. Any differences (e.g., additional components that are needed or components that are no longer needed) are determined by comparing the current BOM to the update BOM. The updating process uses these differences to modify the operating system (deleting and/or adding components), and then uses the update BOM as the new current BOM for the operating system.

According to another aspect, the components (and thus functionality) of an operating system, as well as the current versions of components, can be identified by accessing the BOM corresponding to the operating system. The BOM maintains the record of at least some of the components included in the operating system, so the information is readily available. Additionally, a check of manifests corresponding to the components in the BOM can also be made to identify additional components that are part of the operating system image but not listed in the BOM.

## BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings. The same numbers are used throughout the figures to reference like components and/or features.

FIG. 1 is a block diagram illustrating an exemplary computer system in accordance with certain embodiments of the invention.

FIG. 2 illustrates an exemplary component and corresponding manifest in accordance with certain embodiments of the invention.

FIG. 3 illustrates an exemplary bill of materials in accordance with certain embodiments of the invention.

FIG. 4 is a block diagram illustrating another exemplary componentized operating system in accordance with certain embodiments of the invention.

FIG. 5 is a block diagram exemplifying how a componentized operating system can be updated in accordance with certain embodiments of the invention.

FIG. 6 illustrates an exemplary operating system installation process in accordance with certain embodiments of the invention.

FIG. 7 illustrates an exemplary process for installing an operating system in accordance with certain embodiments of the invention.

FIG. 8 illustrates an exemplary process for updating an operating system in accordance with certain embodiments of the invention.