

COMMONWEALTH of AUSTRALIA
Patents Act 1952

APPLICATION FOR A STANDARD PATENT

623708

I/We

Sun Microsystems, Inc.

of

2550 Garcia Avenue, Mountain View, California, 94043, United States of America

hereby apply for the grant of a Standard Patent for an invention entitled:

Method for loading an operating system through a network

which is described in the accompanying complete specification.

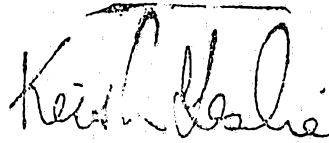
Details of basic application(s):-

<u>Number</u>	<u>Convention Country</u>	<u>Date</u>
343843	United States of America	26 April 1989

The address for service is care of DAVIES & COLLISON, Patent Attorneys, of 1 Little Collins Street, Melbourne, in the State of Victoria, Commonwealth of Australia.

DATED this FOURTH day of DECEMBER 1989

To: THE COMMISSIONER OF PATENTS



.....
a member of the firm of
DAVIES & COLLISON for
and on behalf of the
applicant(s)

Davies & Collison, Melbourne

M 014340 041289

(12) PATENT ABRIDGMENT (11) Document No. AU-B-45870/89
(19) AUSTRALIAN PATENT OFFICE (10) Acceptance No. 623708

- (54) Title
METHOD FOR LOADING AN OPERATING SYSTEM THROUGH A NETWORK
- International Patent Classification(s)
(51)⁴ **G06F 009/24 G06F 015/16**
- (21) Application No. : **45870/89** (22) Application Date : **04.12.89**
- (30) Priority Data
- (31) Number (32) Date (33) Country
343843 26.04.89 US UNITED STATES OF AMERICA
- (43) Publication Date : **01.11.90**
- (44) Publication Date of Accepted Application : **21.05.92**
- (71) Applicant(s)
SUN MICROSYSTEMS, INC.
- (72) Inventor(s)
TADD VAUGHN OTTMAN
- (74) Attorney or Agent
DAVIES COLLISON CAVE , 1 Little Collins Street, MELBOURNE VIC 3000
- (56) Prior Art Documents
EP 222370
EP 196661
EP 106213
- (57) Claim

1. In a computer network comprising a plurality of computers, each computer comprising a central processing unit (CPU), memory, offline storage such as disk drives, input/output devices and an operating system located on a disk drive, a method for loading an operating system onto a first computer from the operating system stored on a second computer comprising:

setting the root directory to be resident in memory;

loading the subset of the operating system into the memory of the first computer, said subset of the operating system comprising commands that provide for the copying of files, the creation of file directories, the formatting of disks and the establishment of a network connection and operation on the network, said subset being small in size to be resident in the memory;

initializing and starting the first computer using the subset of the operating system loaded into memory;

establishing communications between the first computer and the network such that the first computer can communicate with the second computer;

copying the files representative of the operating system from the second computer and transferring the operating system files from the second computer over the network to the disk drive of the first computer;

(11) AU-B-45870/89

-2-

(10) 623708

whereby the disk of the first computer is rewritten with the operating system transferred from the second computer without corrupting the operating system currently running in the memory of the first computer system.

623708

COMMONWEALTH OF AUSTRALIA
PATENTS ACT 1952
COMPLETE SPECIFICATION

NAME & ADDRESS
OF APPLICANT:

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View California 94043
United States of America

NAME(S) OF INVENTOR(S):

Tadd Vaughn OTTMAN

ADDRESS FOR SERVICE:

DAVES & COLLISON
Patent Attorneys
1 Little Collins Street, Melbourne, 3000.

COMPLETE SPECIFICATION FOR THE INVENTION ENTITLED:

Method for loading an operating system through a network

The following statement is a full description of this invention, including the best method of performing it known to me/us:-

1. FIELD OF THE INVENTION:

5 This invention relates to the loading and upgrading of operating systems on computer systems connected in a network environment.

2. ART BACKGROUND:

10 An operating system consists of program modules within a computer system that govern the control of equipment resources such as processors, main storage, secondary storage and I/O devices. These modules resolve conflicts, attempt to optimize performance and simplify the effective use of the computer. They act as an interface between the user programs and the physical computer hardware. In order to load an operating system, an I/O
15 device such as a tape drive executes a program resident in the tape drive and inputs the new operating system data into a predetermined area of memory in the computer. During the course of loading an operating system the disk space where the operating system is loaded is erased and re-written with the new
20 operating system data. The amount of information to be loaded on the system is significant and takes one and one-half to two hours to read and download the information from the tapes.

25 Before the advent of the distributed systems of today, a single computer system would service a multiplicity of users. Thus when it came time to upgrade the operating system, the system manager would only have to load the new operating system onto a single computer system. However, in today's distributed processing environment, it is quite common that each user has his own computer with its own operating system whereby the computer is
30 networked together with other computers. Thus, when an operating system upgrade is performed, the process of loading the operating system from an I/O

device has to be performed with respect to each individual computer system in the network. The upgrade of multiple computer systems can be quite time consuming, and it can be even more burdensome in an operating system development environment where operating system upgrades may be performed
5 as frequently as once a week. Thus, it would be preferable in a distributed environment to load the operating system onto one machine from a tape and thereafter transmit copies of the operating system over the network to each computer system connected to the network.

10 If the operating system is transferred from one system to another through a network, the amount of time required to load the operating system is decreased. However, to load the system onto a computer in a network environment, the computer system must be booted and running an operating system accessible by the computer hardware and capable of network
15 communication. It should be remembered that when the new operating system is loaded onto the computer the disk space is re-written thereby over-writing and corrupting the existing operating system concurrently being used by the computer hardware. A way to avoid the problem of over-writing the presently used operating system is to boot the computer system to be loaded or upgraded
20 from an operating system resident on another computer in the network. That is, when the computer hardware is powered on a signal is sent out over the network to which the computer hardware with the operating system to be accessed responds. The remote network computer then provides disk partitions on its own local disk for the computer to use. The computer system to be
25 upgraded is then booted using the operating system located on the network computer. Once the computer is booted and active on the network, the new version of the operating system can be down-loaded from the network computer system.

However, in order to utilize such an approach, the network must be configured and initialized to know that when the particular computer is powered on, that computer is to be booted using the operating system accessible on a specific network computer. Thus if the network configuration 5 changes or if it is desirable to boot up from a different network computer than the one initially specified, the network parameters then have to be changed.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a method for loading operating systems from one computer to another computer in a network environment.

5

In the method of the present invention a subset of the operating system supporting network communications is loaded into a root partition created in the transferee computer's memory. Once the transferee computer system is thus booted from and running off the memory resident operating system and connected to the network, it is ready to receive from the transferor computer system the new version of the operating system to be loaded on its disk. Because, the currently utilized operating system on the transferee computer is running off of memory as opposed to off the disk, the problems in the prior art with over-writing the currently utilized operating system when downloading the new version of the operating system are eliminated. In addition, the time required to download the operating system is decreased because the data transfer rate between two computer systems located on the network is much greater than that between an external I/O device, such as a tape drive and a computer. Also, updates to the operating system on the transferor computer do not entail the creation of tapes for the installation of the updated version of the operating system on the transferee computer.

10

15

20

BRIEF DESCRIPTION OF THE INVENTION

The objects, features and advantages of the method of the present invention will be apparent from the following detailed description of the preferred embodiment in which:

FIG. 1 is a flow chart outlining the steps performed in the method of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

An entire operating system can be transferred to a computer (transferee computer) from another computer (transferor computer) located on the same network. This capability is available by loading a subset of an operating system into a root directory established in RAM in the transferee computer thereby freeing the disk to receive the operating system files. The subset of the operating system to be loaded into RAM includes those commands necessary for the computer to connect and operate on the network thereby permitting the communication between the transferor computer and the transferee computer. By establishing the root directory in RAM and storing the operating system in RAM, the integrity of the system being installed on the disk is insured because all data relative to the running operating system is located within memory.

15

Throughout the discussion below, the method of the present invention is described with respect to the UNIX® (UNIX is a Registered Trademark of AT&T) operating system and in particular, the Sun operating system, SunOS™ (SunOS is a trademark of Sun Microsystems, Inc.) manufactured by Sun Microsystems, Inc., Mountain View, California. However, as it will be clear from the discussion, the method of the present invention may be utilized in conjunction with most any operating system which provides for a network environment.

25

The process for transferring an operating system across a network is described in the flow chart of Fig. 1. At block 20, the root directory is set to be in the RAM and a subset of the operating system is loaded into memory. This is typically achieved by powering up the system and interrupting the default loading of the operating system so that the root device can be reset to be the RAM and the operating system can be reinitialized to be located in RAM.

30

Typically the subset of the operating system is transferred from a device, such as a tape, to the RAM. However, in a UNIX system, a subset of the operating system may be transferred from another computer connected through a network, using the Trivial File Transfer Protocol (TFTP), a low level, device specific, address specific protocol which permits a file transfer from one device to another over a network

The subset of the operating system comprises only those commands required to boot the system and perform basic functions such as copy files, rename files, link files, create directories. In addition the subset should include commands to format the disk, mount file systems and establish and maintain network communications. Thus in the present embodiment, a subset of the UNIX operating system may be loaded in RAM. It should be noted, however, that the operating system loaded in RAM need not be a subset of the operating system to be transferred to the disk of the transferee computer. In SunOS, it is preferable that a superset of MUNIX is loaded into RAM. MUNIX is a very small subset of the UNIX operating system which is used to create disk partitions on new computer systems. The version of MUNIX used in the system of the present invention however, includes additional commands which permit the system to be connected to and active on the network and therefore can receive and transmit data. These extra commands give the computer the capability referred to, in the SunOS environment, as "client-side NFS."

Once the subset of the operating system is loaded into RAM, at block 30, the system is booted using the memory resident operating system, and the local disk where the new operating system is to be loaded is cleared. In addition, it is preferred that two disk partitions are created where the operating system will be stored. The first partition is linked to or mounted on the subdirectory of the root directory, and the second partition is mounted on the first partition. Thus when the operating system is loaded to the subdirectories of

the root directory resident in RAM, the data will be transferred appropriately to the first or second disk partition. It is also preferred that another disk partition be created for the swapping of data during the execution of commands during the process. This disk partition provides the extra scratch memory that may be
5 needed during the execution of certain commands executed during the transfer of the operating system. Typically, it is the same swap partition used by the computer during normal operation.

At block 40, the network addresses of the transferor computer and
10 the transferee computer as well as the addresses of gateways that may be located between the two computers are input to the system, and at block 50 the network routine to connect the two computers is executed. For example, in SunOS a shell script routine is used to initialize the network environment which is NFS. NFS is a utility which provides a capability which permits one computer
15 system to mount the remote disk space of another computer system in the network and use it as if it is locally attached disk space (For more information on NFS see R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, B. Lyon, Design and Implementation of the Sun Network File System, USENIX, Summer 1985, p.
20 119-130)

At block 60, once the transferor computer system is connected to the transferee computer system through the network, a copy of the new operating system is transferred from the transferor computer system to the transferee computer system. In the SunOS-NFS environment, the mount
25 command is used to mount the file system of the transferor computer onto the root directory of the transferee computer to simplify the copying process and the UNIX TAR command is executed to transfer the files representative of the operating system. The TAR command archives and extracts multiple files into a single file archive called a TAR file. A TAR file is usually located on a magnetic
30 tape but it can be any file. In the present case the file is virtually a stream of

information between two TAR processes: one TAR process reading the operating system files on the transferor system, the other TAR process extracting files from the stream and writing them onto the disk of the transferee computer recreating the directories and the position and protection of files as they exist on the transferor computer system.

5

While the invention has been described in conjunction with the preferred embodiment, the UNIX operating system, and in particular SunOS running NFS, it is evident that numerous alternatives, modifications, variations and uses will be apparent to those skilled in the art . In particular it is evident, in light of the foregoing description, that a variety of computer types, operating systems and networks can be applied to the method of the present invention.

10

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:-

1. In a computer network comprising a plurality of computers, each computer comprising a central processing unit (CPU), memory, offline storage
5 such as disk drives, input/output devices and an operating system located on a disk drive, a method for loading an operating system onto a first computer from the operating system stored on a second computer comprising:
- setting the root directory to be resident in memory;
 - loading the subset of the operating system into the memory of the first
10 computer, said subset of the operating system comprising commands that provide for the copying of files, the creation of file directories, the formatting of disks and the establishment of a network connection and operation on the network, said subset being small in size to be resident in the memory;
 - initializing and starting the first computer using the subset of the
15 operating system loaded into memory;
 - establishing communications between the first computer and the network such that the first computer can communicate with the second computer;
 - copying the files representative of the operating system from the second computer and transferring the operating system files from the second computer
20 over the network to the disk drive of the first computer;
- whereby the disk of the first computer is rewritten with the operating system transferred from the second computer without corrupting the operating system currently running in the memory of the first computer system.
- 25 2. The method of claim 1 wherein the operating system runs a network file system which permits file systems to be mounted from remote devices connected to the network.
3. The method of claim 2 wherein the operating system is a UNIX
30 operating system.

4. The method of claim 2 further comprising the steps of:
cleaning the portions of the disk of the first computer where the operating system is to be loaded;

5 establishing two partitions on the disk where the operating system is to be loaded;

mounting the first partition onto a subdirectory of the root directory located in memory;

creating a directory in the first partition; and

10 mounting the second partition onto the directory created in the first partition;

whereby the operating system files to be transferred and loaded are loaded into directories connected to the root directory resident in memory.

15 5. The method of claim 4 further comprising:

mounting a file system located on the second computer, containing the operating system files to be copied, on to a subdirectory of the root directory of the first computer system to simplify the transfer of files from the second computer system to the first computer system.

20

6. The method of claim 1 wherein the operating system is a UNIX operating system and the commands to copy and transfer operating system files from the second computer system to the first computer system is the UNIX TAR command.

25

7. The method of claim 1 further comprising creating a disk partition on the disk of the first computer for the swapping of data during execution of the commands in the subset of the operating system.

8. A method for loading an operating system substantially as hereinbefore described with reference to the drawings.

5

10

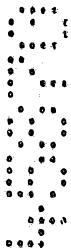
DATED this 28th day of February, 1992

SUN MICROSYSTEMS, INC.

By its Patent Attorneys

DAVIES COLLISON CAVE

15



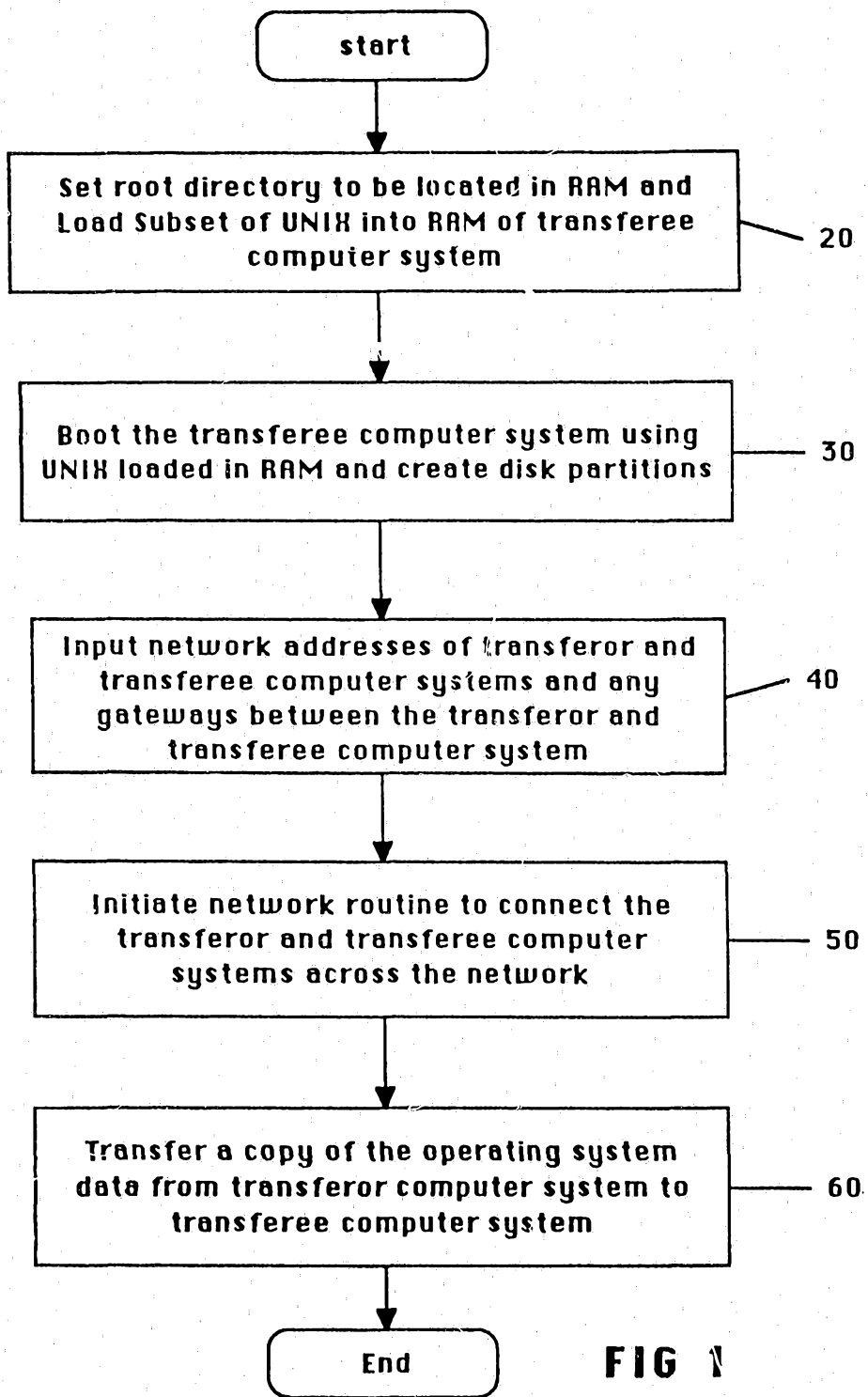


FIG 1