Claims

1. A method for upgrading software, comprising:

reporting, by a client, a upgrading request of software to a server, and the upgrading request carrying file information of to-be-upgraded software in local;

determining, by the server, difference between the to-be-upgraded software and a latest version software according to the file information of the to-be-upgraded software in the upgrading request, and generating, by the server, upgrading instruction information according to the difference and sending the upgrading instruction information to the client;

downloading and updating, by the client, corresponding files and executing, by the client, a corresponding local upgrading operation according to an instruction in the upgrading instruction information.

2. The method according to claim 1, wherein, the file information of the to-be-upgraded software carried in the upgrading request comprises: file name and version information of each of files of the to-be-upgraded software;

determining, by the server, difference between the to-be-upgraded software and the latest version software according to the file information of the to-be-upgraded software in the upgrading request comprises: comparing, by the server, the to-be-upgraded software with the latest version software according to file name and version information of each of files of the to-be-upgraded software, and determining, by the server, file difference and version difference;

generating, by the server, upgrading instruction information according to the difference comprises:

finding, by the server, a newly-added file and a modified file in the latest version software according to the file difference, and organizing, by the server, list information of the newly-added file and modified file into a to-be-updated file list;

finding, by the server, operation instruction sets of all software each of which the version is higher than the version of the to-be-upgraded software in the upgrading request according to the version difference, and combining, by the server, found operation instruction sets;

taking, by the server, the to-be-updated file list and combined operation instruction

set as the upgrading instruction information.

3. The method according to claim 2, wherein,

generating, by the server, upgrading instruction information further comprises: finding, by the server, function description information of the latest version software, and adding, by the server, the function description information to the upgrading instruction information;

after receiving, by the client, the upgrading instruction information and before downloading and updating, by the client, a corresponding file and executing a corresponding local upgrading operation, further comprising: displaying, by the client, the function description information comprised in the upgrading instruction information.

4. The method according to claim 2, wherein, the to-be-updated file list comprises: a download path and a relative path of each to-be-updated file;

downloading and updating, by the client, a corresponding file according to an instruction in the upgrading instruction information comprises: downloading, by the client, the corresponding file according to the download path in the to-be-updated file list, and updating downloaded file to the relative path of an installation directory of the to-be-upgraded software.

5. The method according to claim 4, wherein, file information in the upgrading request further comprises a HASH value of each file;

after finding, by the server, a modified file in the latest version software, further comprising: finding, by the server, a HASH value in the upgrading request according to a file name of the modified file, and finding, by the server, an original file which has the same HASH value and is stored in the server, and performing, by the server, a binary comparison on the original file and a latest version file with the same name as the original file, generating a binary difference file, replacing a download path of the file which has the same name as the binary difference file and is carried in the to-be-updated file list with a download path of the binary difference file;

downloading, by the client, a corresponding binary difference file according to the download path in the to-be-updated file list, and combining the binary difference file with a file which has the same name as the binary difference file and is located in the relative path of the installation directory of the to-be-upgraded software.

6. The method according to claim 2, wherein, each operation instruction in the operation instruction set comprises: an operation instruction name identifiable for the

client, a relative path and a file name of a target file corresponding to an operation of the operation instruction;

executing, by the client, a corresponding local upgrading operation according to an instruction in the upgrading instruction information comprises: finding a target file with the file name in an operation instruction from the relative path of the installation directory of the local to-be-upgraded software, and performing the operation instruction on the target file.

7. The method according to claim 1, wherein, operations on the client side in the method being executed by an upgrading module corresponding to the local to-be-upgraded software, and the method further comprises: after the upgrading module is started and before reporting the upgrading request, interacting, by the upgrading module, with the server first to determine whether the upgrading module needs to be upgraded, and downloading, by the upgrading module, files of a new version upgrading module when the upgrading module needs to be upgraded, and replacing files of original upgrading module with the files of the new version upgrading module.

8. The method according to claim 1, wherein,

after downloading, by the client, the corresponding file according to the instruction in the upgrading instruction information, and before updating the corresponding file, further comprising: backing up local to-be-updated original files before updating the files;

when a file is not updated successfully, determining whether the file must be updated successfully, when the file must be updated successfully, using backup files to restore upgraded files; otherwise, registering the latest download file corresponding to the file in a restart replacement service of an operating system, and starting the restart replacement service to update the file when the operating system is restarted.

9. A system for upgrading software, comprising:

an upgrading module set in a client, configured to report a software upgrading request carrying file information of a to-be-upgraded software to the server, download and update a corresponding file and executing a corresponding local upgrading operation according to an instruction in upgrading instruction information returned by a server; and

an upgrading service module set in the server, configured to receive the upgrading request sent by the upgrading module in the client, and determine difference between the to-be-upgraded software and a latest version software according to file information of the to-be-upgraded software in the upgrading request, generate upgrading instruction

information according to the difference and issues the upgrading instruction information to the upgrading module in the client.

10. The system according to claim 9, wherein, the upgrading service module comprises:

a version difference determining module, configured to compare the to-be-upgraded software with the latest version software according to file name and version information of each file comprised in the to-be-upgraded software, and determine file difference and version difference;

a to-be-updated file determining module, configured to find a newly-added file and a modified file in the latest version software according to the file difference, and organize list information of the newly-added file and modified file to form a to-be-updated file list;

an operation instruction determining module, configured to find operation instruction sets of all software each of which the version is higher than version of the to-be-upgraded software in the upgrading request, and combine found operation instruction sets; and

an upgrading instruction combining and sending module, configured to take the to-be-updated file list and combined operation instruction set as upgrading instruction information and send the upgrading instruction information to the upgrading module of the client.

11. The system according to claim 10, wherein, the upgrading service module further comprises:

a description information determining module, configured to find function description information of the latest version software, and send the function description information to the upgrading instruction combining and sending module;

the upgrading instruction combining and sending module further adds the function description information to the upgrading instruction information and send the upgrading instruction information to the upgrading module of the client.

12. The system according to claim 10, wherein,

the to-be-updated file list comprises: a download path and a relative path of each to-be-updated file;

each operation instruction in the operation instruction set comprises: an operation instruction name identifiable for local client, a relative path and a file name of a target file corresponding to an operation of the operation instruction;

the upgrading module in the client comprises:

a file downloading module, configured to download a corresponding file according to a download path in the to-be-updated file list in the upgrading instruction information;

a difference file combining module, configured to, according to the to-be-updated file list, update the downloaded file to the relative path of the installation directory of the local to-be-upgraded software.

an operation performing module, configured to find a target file with the file name in an operation instruction from the relative path of the installation directory of the to-be-upgraded software, and perform the operation instruction on the target file.

13. The system according to claim 12, wherein, the to-be-updated file determining module comprises a binary difference file generating module, configured to, after the to-be-updated file determining module finds a modified file in the latest version software, find a corresponding HASH value in the upgrading request according to a file name of the modified file, and find an original file which is stored at the server and has the same HASH value according to the HASH value, perform a binary comparison on the original file and a latest version file which has the same file name as the original file, generate a binary difference file, and replace a download path of a file, which has the same name as the binary difference file and is carried in the to-be-updated file list with a download path of the binary difference file;

the difference file combining module in the client further comprises a binary difference combining module, configured to, after downloading the binary difference file, combine the binary difference file and a file which has the same name as the binary difference file and is located at the relative path of the installation directory of the local to-be-upgraded software.

14. The system according to claim 12 or 13, wherein, the difference file combining module further comprises a backup operating module, configured to, before the corresponding file is updated, back up local to-be-updated original files, when a file is not updated successfully, determine whether the file must be updated successfully, when the file must be updated successfully, use the backup files to restore all upgraded files; otherwise, register the latest download file corresponding to the file in a restart replacement service of an operating system, and start the restart replacement service to update the file when the operating system is restarted.

15. The system according to claim 9, wherein, the upgrading module in the client

comprises a self-upgrading module, configured to, before a upgrading request is reported, interact with the server to determine whether the upgrading module needs to be upgraded, when the upgrading module needs to be upgraded, download files of a new version upgrading module, and replace files of the original upgrading module with the files of the new version upgrading module; otherwise, the upgrading request is reported.