



(51) International Patent Classification:
G06F 9/30 (2006.01)

(21) International Application Number:
PCT/US2015/018555

(22) International Filing Date:
3 March 2015 (03.03.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventors: **CULTER, Bradley G**; 11445 Compaq Center Dr W, Houston, Texas 77070 (US). **BANDA, Kishore Kumar**; Sy.No.192, Whitefield Road, Mahadevapura Post, Bangalore 560048 (IN). **CHRISTIAN, Charles E**; 13 Queenberry Way, Basking Ridge, New Jersey 07920 (US).

(74) Agents: **ADEKUNLE, Olaolu O** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

- with international search report (Art. 21(3))

(54) Title: COMMAND PARSING IN COMMAND-LINE-INTERFACE

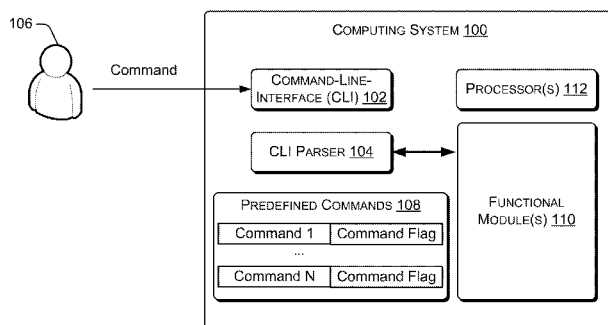


Fig. 1

(57) Abstract: The present subject matter relates to parsing of commands in a command-line-interface (CLI). An example computing system comprises a CLI parser to parse a command entered by a user, and identify a command, from predefined commands stored in the computing system, that matches the command entered by the user. Each predefined command is associated with a command flag. The flag for each predefined command is set to a status to allow execution of the respective predefined command in at least one of predefined user modes. When a user mode in which the computing system is being operated is same as the at least one of the predefined user modes in which the execution of the identified command is allowed based on the associated flag, the CLI parser forwards a request to a functional module to perform an operation. The functional module and the operation are indicated by the identified command.



COMMAND PARSING IN COMMAND-LINE-INTERFACE

BACKGROUND

[0001] A computing system, such as a desktop computer, a laptop, a server, and an enclosure, generally, has a command line interface (CLI) to which a user can enter commands for execution on the computing system. The commands may include CLI commands for tasks, such as setting or resetting of system parameters and components, viewing of system parameters, switching ON or OFF of system components, restoring factory defaults, and upgrading the firmware. The system parameters may include serial number, MAC address, IP address, administrator password, timeouts, system time, and such. The system components may include CPU, switches, microcontrollers, and such.

BRIEF DESCRIPTION OF DRAWINGS

[0002] The following detailed description references the drawings, wherein:

[0003] Fig. 1 illustrates a computing system with a command-line-interface and a command-line-interface parser, according to an example implementation of the present subject matter;

[0004] Fig. 2 illustrates predefined commands arranged in a hierarchical tree structure, according to an example implementation of the present subject matter;

[0005] Fig. 3(a) illustrates predefined commands in hierarchical nodes of the hierarchical tree structure with command flag settings, according to an example implementation of the present subject matter;

[0006] Fig. 3(b) illustrates predefined commands in hierarchical nodes of the hierarchical tree structure with command flag settings, according to another example implementation of the present subject matter;

[0007] Fig. 4 illustrates a method of parsing commands in a command-line-interface of a computing system, according to an example implementation of the present subject matter;

[0008] Fig. 5 illustrates a method of parsing commands in a command-line-interface of a computing system, according to an example implementation of the present subject matter; and

[0009] Fig. 6 illustrates an example system environment for parsing commands in a command-line-interface of a computing system, according to an example implementation of the present subject matter.

DETAILED DESCRIPTION

[0010] CLI is an interface through which a user can interact with a computing system for executing various commands related to system settings and configuration. The commands may be executed to set, reset, or view serial number, MAC address, IP address, administrator password, timeouts, system time, etc. The commands may also be executed to upgrade the firmware, restore factory defaults, and switch ON or OFF CPU, switches, microcontrollers, etc. Such commands are predefined and are tabulated in a table stored in the computing system.

[0011] The user may enter to the CLI a command which the user wishes to execute on the computing system. For execution of the command entered by the user, the command is parsed by a CLI parser of the computing system. Based on the parsing, a command, from among the tabulated predefined CLI commands, that matches the command entered by the user is identified. A request may then be sent to a functional module of the computing system to perform an operation based on the identified command.

[0012] Generally, a set of commands, for example, commands for setting or resetting serial numbers, MAC address, and factory-user password, are allowed for execution only when the computing system is operated in a factory-user mode. Other commands, for example, commands for viewing system time, viewing version, and resetting CPU and other components are allowed for execution irrespective of whether the computing system is operated in the factory-user mode or an end-user mode. The factory-user mode may be understood as a user mode in which the computing system is operated by a factory-user in the factory. The end-user mode may be understood as a user mode in which the computing system is operated by any end-user, for example, a customer of the system, outside the factory. The commands that are allowed for execution only in the factory-user mode may be referred to as factory-

restricted commands, and the commands that are not restricted and allowed for execution in any user mode may be referred to as all-users commands.

[0013] The computing system may be associated with a flag, where the status of the flag is indicative of the user mode in which computing system is being operated. This flag, also referred to as a system mode flag, may be stored in a flash memory of the computing system. The computing system may be operated in the factory when the computing system is assembled and powered ON for the first time in the factory. The system mode flag may have a status, for example '0', to indicate that the computing system is operating in the factory-user mode. While operating in the factory, a command, such as, "set IAP" (set initial administrator password) is executed to set an administrator password for the computing system. On execution of this command, the status of the system mode flag is changed to another status, for example '1.' The changed status of the system mode flag may indicate that the computing system can no more be operated in the factory-user mode and can only be operated in the end-user mode.

[0014] Generally, the computing system has two tables of commands stored therein and also has two CLI parsers, where one pair of CLI parser and table of commands is for parsing and execution of commands when the computing system is operating in the factory-user mode, and the other pair of CLI parser and table of commands is for parsing and execution of commands when the computing system is operating in the end-user mode.

[0015] To elaborate this, the computing system may have a first table of commands including both the factory-restricted commands and the all-users commands, and a second table of commands including only the all-users commands. The user mode in which the computing system is operating may be determined based on the status of the system mode flag. When the computing system is operating in the factory-user mode, a first CLI parser may parse and execute commands based on the first table of commands, and when the computing system is operating in the end-user mode, a second CLI parser may parse and execute commands based on the second table of commands.

[0016] The use of two separate CLI parsers and two separate tables of commands increases the maintenance cost of the computing system. Moreover, storing the same set of all-users commands individually in each of the two tables of commands results in duplicate storage of the all-users commands. The duplicate storage leads to increase in memory requirement of the computing system. The larger memory may increase the cost of the computing system. Further, in case the all-users commands are to be updated or a new all-users command is to be added, both the tables of commands may have to be updated separately. Incomplete or inconsistent update of any of the two tables may lead to an inconsistent execution of all-users commands in the factory-user mode and in the end-user mode. Further, when the user mode of the computing system is switched from the factory-user mode to the end-user mode, or vice-versa, the CLI parsers may have to be switched. This switching of the CLI parsers may increase the operational cost for execution of commands in the computing system.

[0017] The present subject matter describes parsing of commands by a single CLI parser in a computing system, irrespective of the operating mode of the computing system. The CLI parser may be table-driven parser that utilizes a table having predefined commands for executing commands entered by a user. The table may be stored in the computing system. In an implementation of the present subject matter, each of the predefined commands in the table is associated with a command flag. The command flag associated with a predefined command is set to a status to allow the execution of that predefined command in at least one user mode from among a plurality of predefined user modes. The command flag may be understood as a parameter that can be represented by one or more binary bits. The status of the command flag may be understood as a binary number corresponding to the one or more binary bits.

[0018] In an example, the plurality of predefined user modes may include a factory-user mode and an end-user mode, and the predefined commands may include factory-restricted commands and all-users commands. For said example, the status of the command flag associated with a factory-restricted command may be set to a first status to allow the execution only in the factory-

user mode, and the status of the command flag associated with an all-users command may be set to a second status to allow the execution in the end-user mode and in the factory-user mode.

[0019] In accordance with the present subject matter, as a user may enter a command to the CLI of the computing system, the CLI parser may process the CLI to parse the command entered by the user. Based on the parsing, a command, from the predefined commands in the table stored in the computing system, which matches the command entered by the user may be identified. Further, the status of the command flag associated with the identified command may be determined. In addition, a user mode in which the computing system is being operated may be determined. In an example implementation, the user mode in which the computing system is being operated may be determined based on a system mode flag associated with the computing system. When the user mode in which the computing system is being operated is same as one of the predefined user modes to which the execution of the identified CLI command is allowed based on the associated command flag, a request may be forwarded to a functional module of the computing system to perform an operation. The functional module and the operation may be indicated by the identified command.

[0020] With the methods and the systems of the present subject matter, a single, common CLI parser can be used for parsing and executing commands, for example, the all-users commands and the factory-restricted commands, irrespective of whether the systems are operated in the factory-user mode by a factory-user or in the end-user mode by an end-user. The single, common CLI parser utilizes a single table that has the predefined commands, where allowance of the execution of each of the predefined commands is distinguished by the associated command flag. With the command flags associated with the predefined commands, the use of two separate tables and two separate CLI parsers for execution of factory-restricted commands in the factory-user mode and for execution of all-users commands in any of the user modes can be eliminated. This facilitates saving of maintenance cost of the computing system. By eliminating the use of two tables, the duplicate storage of the all-users

commands can be avoided. This facilitates reduction of the memory requirement of the computing system.

[0021] Further, since a single CLI parser is used, the process of switching of the CLI parser is also eliminated. This may help in reducing the cost of execution of the commands when the user mode, in which the computing system is being operated, is switched. Further, with the use of a single table of commands, the possibility of incomplete or inconsistent update of the table of commands, which may otherwise be present, can be avoided.

[0022] Further, by associating a command flag with each of the predefined commands in the table, it is easy to control between allowance and disallowance of execution of a command in a particular user mode. The allowance and disallowance of execution can be controlled by setting appropriate status of the command flag for that command. For example, an all-user command can be switched to a factory-restricted command, or vice-versa, by changing the status of the command flag appropriately.

[0023] The following detailed description refers to the accompanying drawings. Wherever possible, the same reference numbers are used in the drawings and the following description to refer to the same or similar parts. While several examples are described in the description, modifications, adaptations, and other implementations are possible. Accordingly, the following detailed description does not limit the disclosed examples. Instead, the proper scope of the disclosed examples may be defined by the appended claims.

[0024] Fig. 1 illustrates a computing system 100, according to an example implementation of the present subject matter. The computing system 100 may be a desktop computer, a laptop, a tablet, a server, an enclosure, and such. As shown, the computing system 100 includes a CLI 102 and a CLI parser 104. The CLI 102 is a user interface provided to a user 106 for entering a command which the user 106 wishes to execute in the computing system 100. The CLI parser 104 is a table-driven parser that can parse the command entered by the user 106 for the execution of the command in the computing system 100, in accordance with the present subject matter.

[0025] The computing system 100 also includes predefined commands 108 related to system settings and configuration. As shown, the predefined commands 108 may include N commands, from command 1 to command N, where each of the N commands has a command flag associated therewith, in accordance with the present subject matter. The command flag associated with each of the N commands is set to a status to allow execution of the respective command in one or more of predefined user modes. The details of the predefined commands 108, the predefined user modes, and the command flag settings for the predefined commands 108 are described in detail later in the description.

[0026] In an implementation, the predefined commands 108 may be stored in a memory of the computing system 100. In an example, the predefined commands 108 may be stored in a non-volatile memory, such as a flash memory of the computing system 100. The predefined commands tabulated in a table. The table may be referred to by the CLI parser 104, in accordance with the present subject matter, for the purpose of execution of the commands entered by the user 106.

[0027] Further, the computing system 100 includes functional module(s) 110. The functional module(s) 110 may include modules that perform operations as indicated by the commands that are being executed in the computing system 100. In an example, the functional module(s) 110 may include, but are not restricted to, network related modules, a system timer, a logging module, power ON/OFF modules for microcontrollers and other components, set/reset modules for one or more processors, and such. Each of predefined commands 108 stored in the computing system 100 may include at least one keyword that indicate a particular functional module 110 and an operation to be performed on that functional module 110. The details of the functional module(s) 110 and the operations associated with the predefined commands 108 are provided further ahead in the description.

[0028] The computing system 100 further includes processor(s) 112. The processor(s) 112 may be implemented as microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state

machines, logic circuitries, and/or any devices that manipulate signals based on operational instructions. Among other capabilities, the processor(s) 112 may fetch and execute computer-readable instructions stored in a memory coupled to the processor(s) 112 of the computing system 100. The memory may include any non-transitory computer-readable storage medium including, for example, volatile memory (e.g., RAM), and/or non-volatile memory (e.g., EPROM, flash memory, NVRAM, memristor, etc.). In an implementation, the CLI 102, the CLI parser 104, and the functional module(s) 110 may be coupled to the processor(s) 112 to perform various functions related to receiving a command from the user 106, and parsing and executing the command provided by the user 106, in accordance with the present subject matter. The functions of the various elements shown in Fig. 1, including any functional blocks labeled as “processor(s)”, may be provided through the use of dedicated hardware as well as hardware capable of executing computer-readable instructions.

[0029] The description hereinafter describes examples of predefined commands 108, predefined user modes, and the command flag settings associated with the predefined commands 108. The description also describes examples of functional module(s) 110 and operations associated with some of the predefined commands 108. The number of different status of the command flag that may be used depends on the number of types of commands and the number of predefined user modes.

[0030] In an example implementation, the predefined commands 108 may include factory-restricted commands and all-user commands, and the predefined user modes include a factory-user mode and an end-user mode. The factory-restricted command may refer to the commands that can be executed only when the computing system 100 is being operated in the factory-user mode. The computing system 100 may said to be operating in the factory-user mode when the computing system 100 is being accessed by a factory-user in the factory. On the other hand, the all-user commands may refer to the commands that can be executed when the computing system 100 is being operated either in the end-user mode or in the factory-user mode. The computing system 100 may said to be operating in the end-user mode when the

computing system 100 is being accessed by an end-user, for example, a customer of the computing system 100. It may be noted that the computing system 100 may have a system mode flag, where the status of the system mode flag is indicative of whether the computing system 100 is being operated in the factory-user mode or the end-user mode.

[0031] For the above described example, the command flag associated with each of the factory-restricted commands may be set to a first status, for example '1', and the command flag associated with each of the all-user commands may be set to a second status, for example '0'. Thus, the status of the command flag of a command is set to the first status to allow execution of the command only in the factory-user mode, and the status of the command flag of a command is set to the second status to allow the execution of the command in both the end-user mode and the factory-user mode.

[0032] Table 1 shows the details of the example status of the command flag and the predefined user mode(s) in which the execution is allowed for the all-user commands and the factory-restricted commands. The command flag herein may have 1-bit representation. The value of the bit may refer to the status of the command flag.

Table 1

Type of predefined command	Command flag	Predefined user mode(s) in which the execution is allowed
all-users command	0	end-user mode and factory-user mode
factory-restricted command	1	only factory-user mode

[0033] The predefined commands 108 may include a variety of commands for setting or resetting various system parameters and functionalities, for viewing various system parameters, for switching ON or OFF various system components, for enabling or disabling functionalities, for upgrading the firmware, for help, for exit, and such. Table 2 illustrates some examples of predefined commands 108 in the computing system 100. The predefined commands 108 include factory-restricted commands, for example, set password, set MAC

(media access control), and set DHCP (dynamic host configuration protocol) enable. The command “set password” is to set the factory user password for the computing system 100. The command “set MAC” is to set MAC address for the computing system 100. The command “set DHCP enable” is to enable the DHCP for the computing system 100. The predefined commands 108 also include all-user command, for example, set time, set DHCP disable, show version, show net, and update. The command “set time” is to set the system time. The command “set DHCP disable” is to disable the DHCP. The command “show version” is to view the hardware and the firmware versions of the computing system 100. The command “show net” is to view the network related configuration settings in the computing system 100. The command “update” is to update the firmware of the computing system 100. It may be noted that Table 2 illustrates some examples of predefined commands, and other examples of predefined commands are also possible.

Table 2

Predefined command	Command flag
.....	...
set time	0
set password	1
set DHCP enable	1
set DHCP disable	0
set MAC	1
show version	0
show net	0
update	0
.....	...
.....	...
.....	...

[0034] Further, as shown in Table 2, each of the factory-restricted commands is associated with a command flag with status ‘1’, and each of the

all-user commands is associated with a command flag with status '0'. An all-user command can be changed to a factory-restricted command and thus be restricted only to the factory-user mode by changing the status of the command flag associated with the all-user command. Similarly, a factory-restricted command can be changed to an all-user command and thus be allowed for execution in both the factory-user mode and the end-user mode by changing the status of the command flag associated with the factory-restricted command.

[0035] Further, each of the predefined commands 108 comprises one or more keywords that indicate the operation to be performed, the functional module 110, on which the operation is to be performed, and the manner in which the operation is to be performed. For example, the command "set time" comprises two keywords "set" and "time". The keyword "set" indicates that the operation to set a system parameter is to be performed, and the keyword "time" indicates that operation is to be performed on the system timer to set the system time. Similarly, the command "set DHCP enable" comprises three keywords "set", "DHCP", and "enable". The keyword "set" indicates that the operation to set a system functionality is to be performed, and the keyword "DHCP" indicates that operation is to be performed on a network related module associated with the DHCP functionality, and the keyword "enable" indicates that the DHCP functionality is to be enabled in the computing system 100. Similarly, the command "update" comprises only one keyword, which indicates to upgrade the firmware of the computing system 100.

[0036] The description hereinafter describes, in detail, the operation of the computing system 100 for executing a command entered by the user 106 in the computing system 100.

[0037] In an example implementation, for the purpose of execution of a command, the user 106 may access the computing system 100. The CLI parser 104 may determine a user mode in which the computing system 100 is being operated. As mentioned earlier, the user mode may be determined based on the status of the system mode flag for the computing system 100.

[0038] After this, the computing system 100 may display the CLI 102 to the user 106. The CLI 102 may be displayed on a display screen (not shown) of the

computing system 100. The CLI 102 may be displayed based on receiving a command execution request from the user 106. The user 106 may enter a command to the CLI 102. The CLI parser 104 may parse the command entered by the user 106. In an example, the CLI parser 104 may parse the command from left to right. Based on the parsing, the CLI parser 104 may identify a command, from the predefined commands 108 stored in the computing system 100, that matches with the command entered by the user 106.

[0039] After identifying the command that matches the command entered by the user 106, the CLI parser 104 may determine the operation and the functional module 110 indicated by the identified command. In addition, the CLI parser 104 may determine the status of the command flag associated with the identified command. After determining the status of the command flag, the CLI parser 104 determines whether the user mode in which the computing system 100 is being operated is same as one of the predefined user modes in which the execution of the identified command is allowed. This determination is made based on the status of the command flag.

[0040] Further, when the user mode in which the computing system 100 is being operated is same as the one of predefined user modes in which the execution of the identified command is allowed, the CLI parser 104 may send a request to the identified functional module 110 to perform the identified operation. The functional module 110 may perform the identified operation, prepare a response based on the operation, and send the response to the CLI parser 104. The CLI parser 104 may then display the response on the display unit of the computing system 100.

[0041] However, when the user mode in which the computing system 100 is being operated is not same as the one of the predefined user modes in which the execution of the identified command is allowed, the CLI parser 104 may prepare an error message and display the error message on the display unit of the computing system 100. In an example, the error message may display "invalid command".

[0042] In an example implementation, the predefined commands 108 may be arranged in a hierarchical tree structure. In the hierarchical tree structure, a

predefined command 108 is arranged in hierarchical nodes forming a branch of the hierarchical tree structure. The hierarchical nodes in a branch are formed based on the keywords associated with one predefined command 108. Each keyword forms a node in the branch. Thus, the number of hierarchical nodes in the branch depends on the number of keywords associated with the predefined command 108. In an example, the hierarchical nodes in a branch representing a predefined command 108 may include a parent node, a child node, and at least one further child node depending of the number of keywords in the predefined command 108. The parent node may represent the first keyword, the child node may represent the second keyword, and the further child node may represent the third keyword, and so on.

[0043] In an example, the parent node may represent a keyword indicative of an operation associated with the predefined command 108. The child node of the parent node may represent a keyword indicative of a functional module on which the operation is to be performed. The at least one further child node may represent keywords indicative of a manner of performing the operation on the functional module.

[0044] With the above described hierarchical tree structure-based arrangement, the predefined commands 108 that include a common keyword would branch out from a common node in the hierarchical tree structure. For example, any two predefined commands 108 that have a common first keyword would have a same parent node. Similarly, any two predefined commands 108 that have common first and second keywords would have same parent and child nodes.

[0045] Fig. 2 illustrates predefined commands 108 arranged in a hierarchical tree structure 200, according to an example implementation of the present subject matter. The hierarchical tree structure 200 depicts the example predefined commands 108 of Table 2. As shown, the predefined commands “show versions” and “show net” have the same parent node 202 that represents the keyword “show”. The parent node 202 has a child node 208-1 that represents the keyword “versions”. The parent node 202 and the child node 208-1 form a branch of the hierarchical tree structure 200 which corresponds to

the predefined command "show versions". The parent node 202 also has another child node 208-2 that represents the keyword "net". The parent node 202 and the child node 208-2 form a branch of the hierarchical tree structure 200 which corresponds to the predefined command "show net".

[0046] Similarly, the predefined commands "set time", "set password", "set MAC", "set DCHP enable", and "set DHCP disable" have the same parent node 204 that represents the keyword "set". The parent node 204 has child nodes 210-1, 210-2, 210-3, and 210-4 that represents the keywords "time", "password", "MAC", and "DHCP". The parent node 204 and the child node 210-1 represent the predefined command "set time". The parent node 204 and the child node 210-2 represent the predefined command "set password". The parent node 204 and the child node 210-3 represent the predefined command "set MAC". The child node 210-4 has two further child nodes 212-1 and 212-2 that represent "enable" and "disable". The parent node 204, the child node 210-4 and the further child node 212-1 represent the predefined command "set DHCP enable". The parent node 204, the child node 210-4 and the further child node 212-2 represent the predefined command "set DHCP disable".

[0047] Further, in the hierarchical tree structure 200, the predefined command "upgrade" is in a single node 206 that represents the keyword "upgrade". The node 206 does not have any child node as the predefined command "upgrade" has only one keyword.

[0048] In an example implementation, the command flag associated with a predefined command may be set at one of the hierarchical nodes in the branch for that predefined command. Fig. 3(a) illustrates predefined commands in hierarchical nodes of the hierarchical tree structure 200 with command flag settings, according to an example implementation of the present subject matter. The hierarchical nodes in branches for the predefined commands "set time", "set password", "set MAC", "set DCHP enable", and "set DHCP disable" are shown in Fig. 3(a). The command flag 302 associated with the predefined command "set time" is set with status '0' at the child node 210-1 representing the keyword "time". Similarly, the command flag 304 associated with the predefined command "set password" is set with status '1' at the child node 210-2

representing the keyword “password”, and the command flag 306 associated with the predefined command “set MAC” is set with status ‘1’ at the child node 210-3 representing the keyword “MAC”. Further, the command flag 308 associated with the predefined command “set DHCP enable” is set with status ‘1’ at the further child node 212-1 representing the keyword “enable”, and the command flag 310 associated with the predefined command “set DHCP disable” is set with status ‘0’ at the further child node 212-2 representing the keyword “disable”.

[0049] Fig. 3(b) illustrates predefined commands in hierarchical nodes of the hierarchical tree structure 200 with command flag settings, according to another example implementation of the present subject matter. Fig. 3(b) also shows the hierarchical nodes in branches for the predefined commands “set time”, “set password”, “set MAC”, “set DHCP enable”, and “set DHCP disable”. As shown in Fig. 3(b), the command flag 312 is set with status ‘1’ at the child node 210-4 representing the keyword “DHCP”. With this, both the predefined commands “set DHCP enable” and “set DHCP disable” are made factory-restricted commands which can now be executed only in the factory-user mode. It may be noted that any two predefined commands having a common node and the command flags for such predefined commands have the same status, a single command flag can be set at the common node instead of setting command flags individually for the two predefined commands. Thus, with hierarchical tree structure-based arrangement of the predefined commands 108, the command flags for the predefined commands 108 can be set with a high level of granularity.

[0050] In an example implementation, the CLI parser may provide command completion suggestions for a partial command entered by the user 106. The partial command may be understood as a portion of a command entered by the user 106. As the user 106 enters a portion of a command, the CLI parser 104 may provide one or more possible suggestions on the complete command. For example, when the user 106 enters a portion “set”, the CLI parser 104 may provide command completion suggestions by displaying on the screen “time”, “MAC”, “password”, and so on. The command completion

suggestions may be provided based on the predefined commands 108 stored in the computing system 100.

[0051] With the setting of the command flag for each of the predefined commands 108, the command completion suggestions may include the predefined commands 108 for which the status of the associated command flag allows the execution in the user mode in which the computing system 100 is being operated. With this, if the computing system 100 is being operated in the end-user mode, the command completion suggestions may include the predefined commands 108 for which the status of the command flag is '0'. Similarly, if the computing system 100 is being operated in the factory-user mode, the command completion suggestions may include the predefined commands 108 for which the status of the command flag is '1' and for which the status of the command flag is '0'.

[0052] For providing the command completion suggestions, the CLI parser 104 may parse the hierarchical tree structure. The portion of the command entered by the user 106 may be compared with the keywords in the nodes, starting from the parent nodes in the hierarchical tree structure. As the keyword in a parent node matches with the portion of the command, the CLI parser 104 scans the child nodes and further child nodes of the parent node to determine the status of the command flags associated with the corresponding predefined commands 108. If the computing system 100 is being operated in the factory-user mode, the CLI parser 104 provides the keywords in the child nodes and in the further child nodes of the corresponding predefined commands 108 for which the status of the command flag is '1' and for which the status of the command flag is '0'. If the computing system 100 is being operated in the end-user mode, the CLI parser 104 provides the keywords in the child nodes and in the further child nodes of the corresponding predefined commands 108 for which the status of the command flag is '0'.

[0053] In an implementation, the CLI parser may provide help information when a help command is executed in the computing system 100 by the user 106. The help information may include usage information for the predefined commands that are supported by the CLI parser 104. The usage information for

a predefined command may illustrate keywords associated with the predefined command and also details of exemplary responses provided by the computing system 100 when the predefined command is executed successfully and when the predefined command is not executed successfully. The usage information associated with each of the predefined commands 108 may be stored in the computing system 100.

[0054] With the setting of the command flag for each of the predefined commands 108, the help information may include usage information of the predefined commands 108 for which the status of the command flag allows the execution in the user mode in which the computing system 100 is being operated. With this, if the computing system 100 is being operated in the end-user mode, the help information may include the usage information for the predefined commands 108 for which the status of the command flag is '0'. Similarly, if the computing system 100 is being operated in the factory-user mode, the help information may include the usage information for the predefined commands 108 for which the status of the command flag is '1' and for which the status of the command flag is '0'.

[0055] Fig. 4 illustrates a method 400 of parsing commands in a command-line-interface of a computing system, according to an example implementation of the present subject matter. The order in which the method 400 is described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the method 400. Furthermore, the method 400 can be implemented by processor(s) or computing device(s) through any suitable hardware, non-transitory machine readable instructions, or combination thereof. Further, although the method 400 is described in context of the aforementioned computing system 100, other suitable computing devices or systems may be used for execution of at least one step of method 400. It may be understood that steps of method 400 can be executed based on instructions stored in a non-transitory computer readable medium, as will be readily understood. The non-transitory computer readable medium may include, for example, digital memories, magnetic storage media,

such as a magnetic disks and magnetic tapes, hard drives, or optically readable digital data storage media.

[0056] Referring to Fig. 4, at block 402, a command entered to a CLI 102 by the user 106 is parsed by the CLI parser 104 of the computing system 100. As described earlier, the computing system 100 may provide the CLI 102 to the user 106 based on a request received from the user 106. The user 106 may accordingly enter a command that the user 106 wishes to execute in the computing system 100.

[0057] At block 404, the command entered by the user 106 is compared with predefined commands 108 stored in the computing system 100 to identify a command, from the predefined commands 108, that matches the command entered by the user 106. As described earlier, each of the predefined commands 108 is associated with a command flag, where the command flag associated with the each predefined command 108 is set to a status to allow execution of that predefined command 108 in at least one of predefined user modes. The comparison and the identification may be performed by the CLI parser 104 of the computing system 100.

[0058] As described earlier, the predefined commands 108 may include factory-restricted commands, where the status of the command flag associated with each factory-restricted command is set to a first status, for example '1', to allow the execution only in the factory-user mode. Also, the predefined commands 108 may include all-user commands, where the status of the command flag associated with each all-user command is set to a second status, for example '0', to allow the execution in the factory-user mode and in the end-user mode. In an example implementation, the predefined commands 108 may be arranged in a hierarchical tree structure, where each predefined command 108 is arranged in hierarchical nodes forming a branch of the hierarchical tree structure, as described earlier in the description. The command flag associated with a predefined command 108 is set at one of the hierarchical nodes in the branch corresponding to that predefined command 108.

[0059] At block 406, a user mode in which the computing system 100 is being operated is determined. The user mode may be determined based on the

status of the system mode flag associated with the computing system 100. The determination may be done by the CLI parser 104 of the computing system 100.

[0060] At block 408, a request is forwarded to a functional module 110 of the computing system 100 to perform an operation, where the request is forwarded when the determined user mode is same as the at least one of the predefined user modes in which the execution of the identified command is allowed based on the command flag for the identified command. The request may be forwarded by the CLI parser 104 of the computing system 100. As mentioned earlier, the functional module 110 to which the request of forwarded and the operation to be performed on the functional module 110 are indicated by the identified command.

[0061] In an example implementation, the CLI parser 104 may provide command completion suggestions for a partial command entered by the user 106. The command completion suggestions may include the predefined commands 108 for which the status of the associated command flag allows the execution in one of the predefined user modes that matches with the user mode in which the computing system 100 is being operated.

[0062] Fig. 5 illustrates a method 500 of parsing commands in the CLI 102 102 of the computing system 100, according to an example implementation of the present subject matter. The order in which the method 500 is described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the method 500. Furthermore, the method 500 can be implemented by processor(s) or computing device(s) through any suitable hardware, non-transitory machine readable instructions, or combination thereof. Further, although the method 500 is described in context of the aforementioned computing system 100, other suitable computing devices or systems may be used for execution of at least one step of method 500. It may be understood that steps of method 500 can be executed based on instructions stored in a non-transitory computer readable medium, as will be readily understood. The non-transitory computer readable medium may include, for example, digital memories, magnetic storage media,

such as a magnetic disks and magnetic tapes, hard drives, or optically readable digital data storage media.

[0063] Referring to Fig. 5, at block 502, a command entered to the CLI 102 by the user 106 is received. At block 504, the command entered by the user 106 is parsed by the CLI parser 104. Based on the parsing, it is determined whether the command is matching with a command from the predefined commands 108 in the computing system 100, at block 506. If there is a matching ('Yes' branch from block 506), the command that matches with the command entered by the user is identified, at block 508. If there is no matching ('No' branch from block 506), an error message is provided on the display screen of the computing system 100, at block 510. In an example, this error message may display "invalid command".

[0064] Further, at block 512, a user mode in which the computing system 100 is being operated is determined based on the status of the system mode flag associated with the computing system 100. Then, at block 514, it is determined whether the user mode in which the computing system 100 is being operated is same as one of the predefined user mode in which the execution of the identified command is allowed based on the command flag for the identified command. If there is a matching ('Yes' branch from block 514), then a request is forwarded to a functional module 110 to perform an operation indicated by the identified command, at block 516. If there is no matching ('No' branch from block 514), an error message is provided on the display screen of the computing system 100, at block 518. In an example, this error message may display "invalid command".

[0065] Fig. 6 illustrates an example system environment 600 for parsing commands in a CLI of a computing system, according to an example implementation of the present subject matter. In an example implementation, the system environment 600 includes a computing system 600 having a processing resource 602 communicatively coupled to a non-transitory computer readable medium 604 through a communication link 606.

[0066] The non-transitory computer readable medium 604 can be, for example, an internal memory device or an external memory device. In an

example implementation, the communication link 606 may be a direct communication link, such as any memory read/write interface. In another example implementation, the communication link 606 may be an indirect communication link, such as a network interface. In such a case, the processing resource 602 can access the non-transitory computer readable medium 604 through a network (not shown). The network may be a single network or a combination of multiple networks and may use a variety of different communication protocols.

[0067] In an example implementation, the non-transitory computer readable medium 604 includes a set of computer readable instructions for parsing commands in the CLI of the computing system 100. The set of computer readable instructions can be accessed by the processing resource 602 through the communication link 606 and subsequently executed to perform acts for parsing the commands.

[0068] Referring to Fig. 6, in an example, the non-transitory computer readable medium 604 includes instructions 608 that cause the processing resource 602 to parse a command entered to the CLI by a user. The non-transitory computer readable medium 604 includes instructions 610 that cause the processing resource 602 to identify a command, from predefined commands stored in the computing system, that matches the command entered by the user. Each of the predefined commands is associated with a command flag. The command flag associated with the each of the predefined commands is set to a status to allow execution of the respective predefined command in at least one of a factory-user mode and an end-user mode. The non-transitory computer readable medium 604 includes instructions 612 that cause the processing resource 602 to determine a user mode in which the computing system 100 is being operated. The non-transitory computer readable medium 604 includes instructions 614 that cause the processing resource 602 to forward a request to a functional module to perform an operation when the user mode in which the computing system is being operated is same as the at least one of the factory-user mode and the end-user mode to which the execution of the identified command is allowed based on the command flag associated with the identified

command. The functional module and the operation are indicated by the identified command.

[0069] In an example implementation, the non-transitory computer readable medium 604 may further include instructions that cause the processing resource 602 to provide command completion suggestions for a partial command entered by the user, where the command completion suggestions may include the predefined commands for which the status of the associated command flag allows the execution in one of the factory-user mode and the end-user mode that matches with the user mode in which the computing system is being operated.

[0070] Although implementations for parsing commands in a CLI in a computing system have been described in language specific to structural features and/or methods, it is to be understood that the present subject matter is not necessarily limited to the specific features or methods described. Rather, the specific features and methods are disclosed and explained as example implementations for parsing commands in the CLI in the computing system.

We claim:

1. A computing system comprising:

a processor; and

a command-line-interface (CLI) parser coupled to the processor to:

parse a command entered to a CLI by a user;

identify a command, from predefined commands stored in the computing system, that matches the command entered by the user, wherein each of the predefined commands is associated with a command flag, wherein the command flag associated with the each of the predefined commands is set to a status to allow execution of the respective predefined command in at least one of predefined user modes; and

when a user mode in which the computing system is being operated is same as the at least one of the predefined user modes in which the execution of the identified command is allowed based on the command flag associated with the identified command, forward a request to a functional module to perform an operation, wherein the functional module and the operation are indicated by the identified command.

2. The computing system as claimed in claim 1, wherein the predefined commands comprise factory-restricted commands, and wherein the status of the command flag associated with each of the factory-restricted commands is set to a first status to allow the execution in a factory-user mode.

3. The computing system as claimed in claim 1, wherein the predefined commands comprise all-users commands, and wherein the status of the command flag associated with each of the all-users commands is set to a second status to allow the execution in an end-user mode and a factory-user mode.

4. The computing system as claimed in claim 1, wherein the predefined commands are arranged in a hierarchical tree structure, each of the predefined commands comprising at least one keyword arranged in hierarchical nodes forming a branch of the hierarchical tree structure, and wherein the hierarchical nodes in the branch comprise:

- a parent node representing a keyword indicative of an operation associated with the respective predefined command;

- a child node representing a keyword indicative of a functional module on which the operation is to be performed; and

- at least one further child node representing keywords indicative of a manner of performing the operation on the functional module;

- wherein the command flag associated with the respective predefined command is set at one of the hierarchical nodes in the branch.

5. The computing system as claimed in claim 1, wherein the CLI parser is to provide command completion suggestions for a partial command entered by the user, and wherein the command completion suggestions comprise the predefined commands for which the status of the command flag allows execution in the user mode in which the computing system is being operated.

6. The computing system as claimed in claim 1, wherein the CLI parser is to provide help information when a help command is executed in the computing system, and wherein the help information comprises usage information of the predefined commands for which the status of the command flag allows execution in the user mode in which the computing system is being operated.

7. A method of parsing commands entered to a command-line-interface (CLI) of a computing system, the method comprising:

- parsing, by a CLI parser, a command entered to the CLI by a user;

- comparing the command entered by the user with predefined commands stored in the computing system to identify a command, from the predefined

commands, that matches the command entered by the user, wherein each of the predefined commands is associated with a command flag which is set to a status to allow execution of the respective predefined command in at least one of predefined user modes;

determining a user mode in which the computing system is being operated; and

when the user mode in which the computing system is being operated is same as the at least one of the predefined user modes in which the execution of the identified command is allowed based on the command flag associated with the identified command, forwarding, by the CLI parser, a request to a functional module to perform an operation, wherein the functional module and the operation are indicated by the identified command.

8. The method as claimed in claim 7, wherein the predefined commands comprise factory-restricted commands, and wherein the status of the command flag associated with each of the factory-restricted commands is set to a first status to allow the execution in a factory-user mode.

9. The method as claimed in claim 7, wherein the predefined commands comprise all-users commands, and wherein the status of the command flag associated with each of the all-users commands is set to a second status to allow the execution in an end-user mode and a factory-user mode.

10. The method as claimed in claim 7, wherein the predefined commands are arranged in a hierarchical tree structure, each of the predefined commands comprising at least one keyword arranged in hierarchical nodes forming a branch of the hierarchical tree structure, and wherein the hierarchical nodes in the branch comprise:

a parent node representing a keyword indicative of an operation associated with the respective predefined command;

a child node representing a keyword indicative of a functional module on which the operation is to be performed; and

at least one further child nodes representing keywords indicative of a manner of performing the operation on the functional module;

wherein the command flag associated with the respective predefined command is set at one of the hierarchical nodes in the branch.

11. The method as claimed in claim 7, comprising providing, by the CLI parser, command completion suggestions for a partial command entered by the user, wherein the command completion suggestions comprise the predefined commands for which the status of the command flag allows execution in the user mode in which the computing system is being operated.

12. A non-transitory computer-readable medium comprising computer-readable instructions for parsing commands entered to a command-line-interface (CLI), wherein the computer readable instructions are executable by a processing resource of a computing system having a CLI parser to:

parse a command entered to the CLI by a user;

identify a command, from predefined commands stored in the computing system, that matches the command entered by the user, wherein each of the predefined commands is associated with a command flag, wherein the command flag associated with the each of the predefined commands is set to a status to allow execution of the respective predefined command in at least one of a factory-user mode and an end-user mode;

determine a user mode in which the computing system is being operated; and

when the user mode in which the computing system is being operated is same as the at least one of the factory-user mode and the end-user mode in which the execution of the identified command is allowed based on the command flag associated with the identified command, forward a request to

a functional module to perform an operation, wherein the functional module and the operation are indicated by the identified command.

13. The non-transitory computer-readable medium as claimed in claim 12, wherein the predefined commands comprise factory-restricted commands, and wherein the status of the command flag associated with each of the factory-restricted commands is set to a first status to allow the execution in the factory-user mode.

14. The non-transitory computer-readable medium as claimed in claim 12, wherein the predefined commands comprise all-users commands, and wherein the status of the command flag associated with each of the all-users commands is set to a second status to allow the execution in the end-user mode and a factory-user mode.

15. The non-transitory computer-readable medium as claimed in claim 12, comprising computer-readable instructions executable by the processing resource to provide command completion suggestions for a partial command entered by the user, wherein the command completion suggestions comprise the predefined commands for which the status of the command flag allows execution in the user mode in which the computing system is being operated.

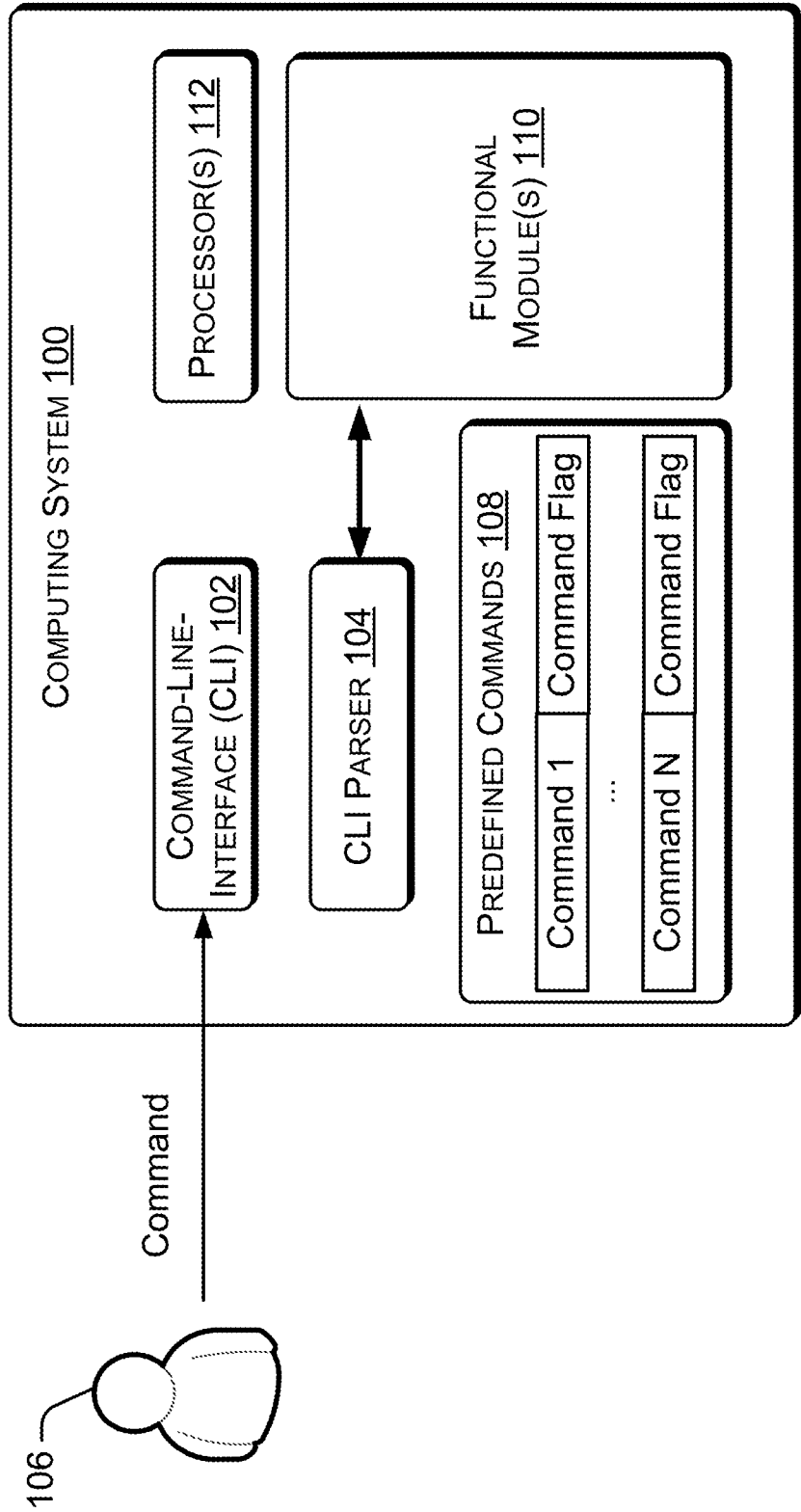


Fig. 1

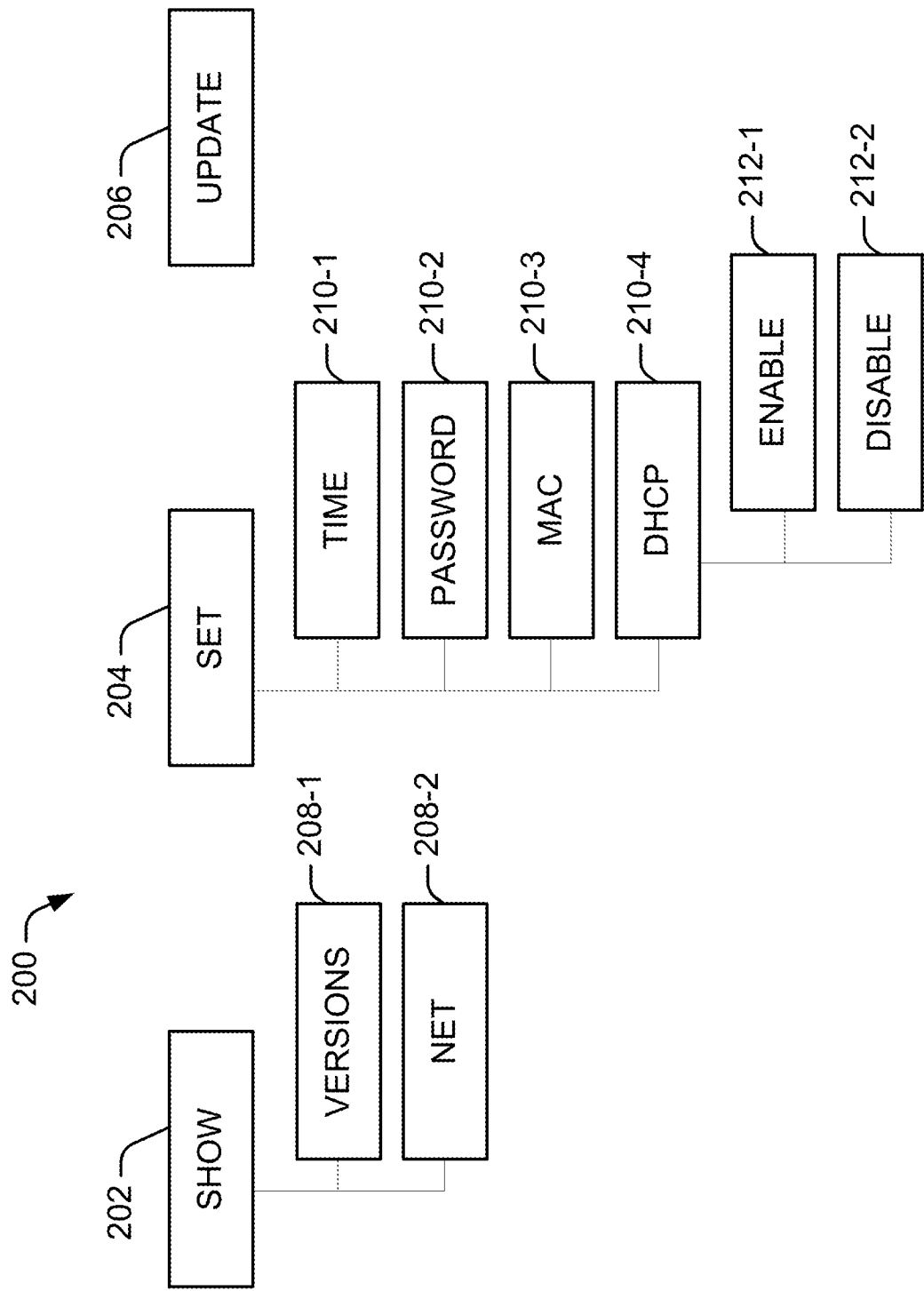


Fig. 2

3/7

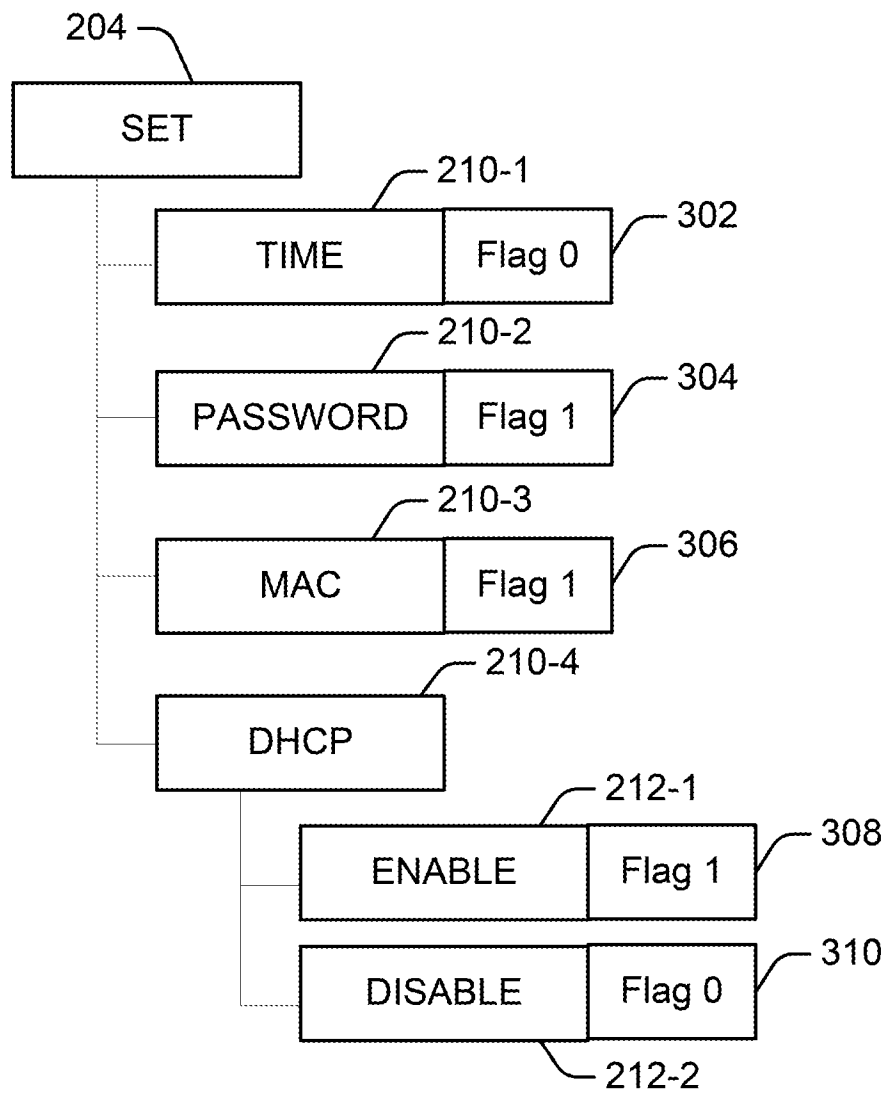
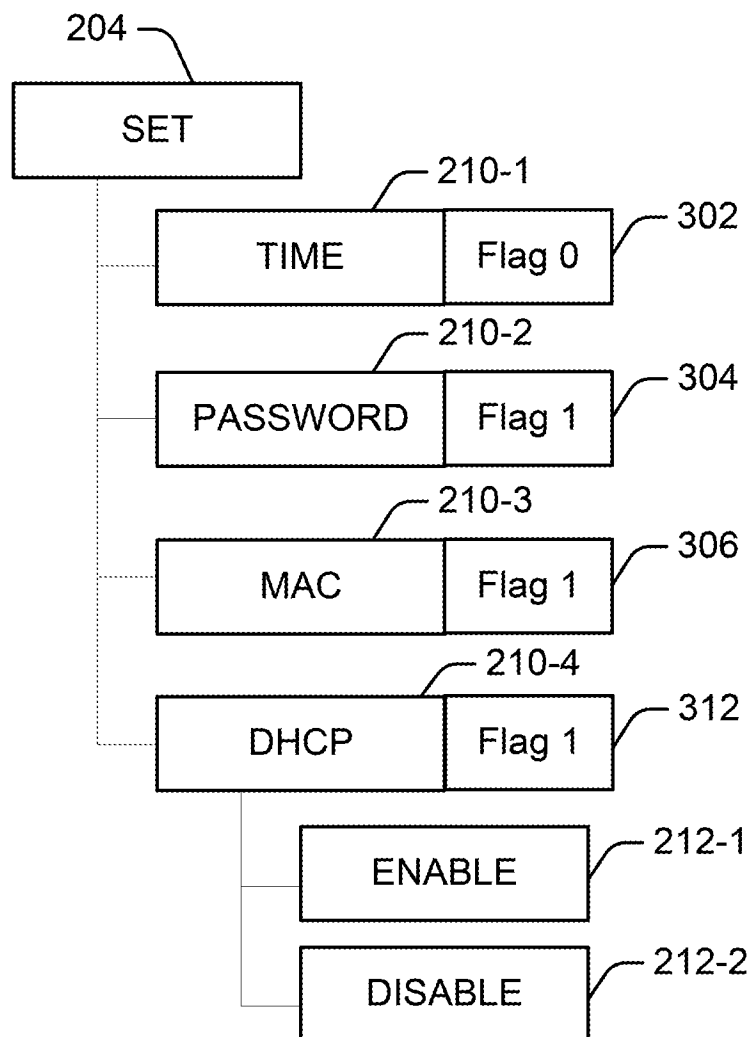


Fig. 3(a)

4/7

**Fig. 3(b)**

5/7

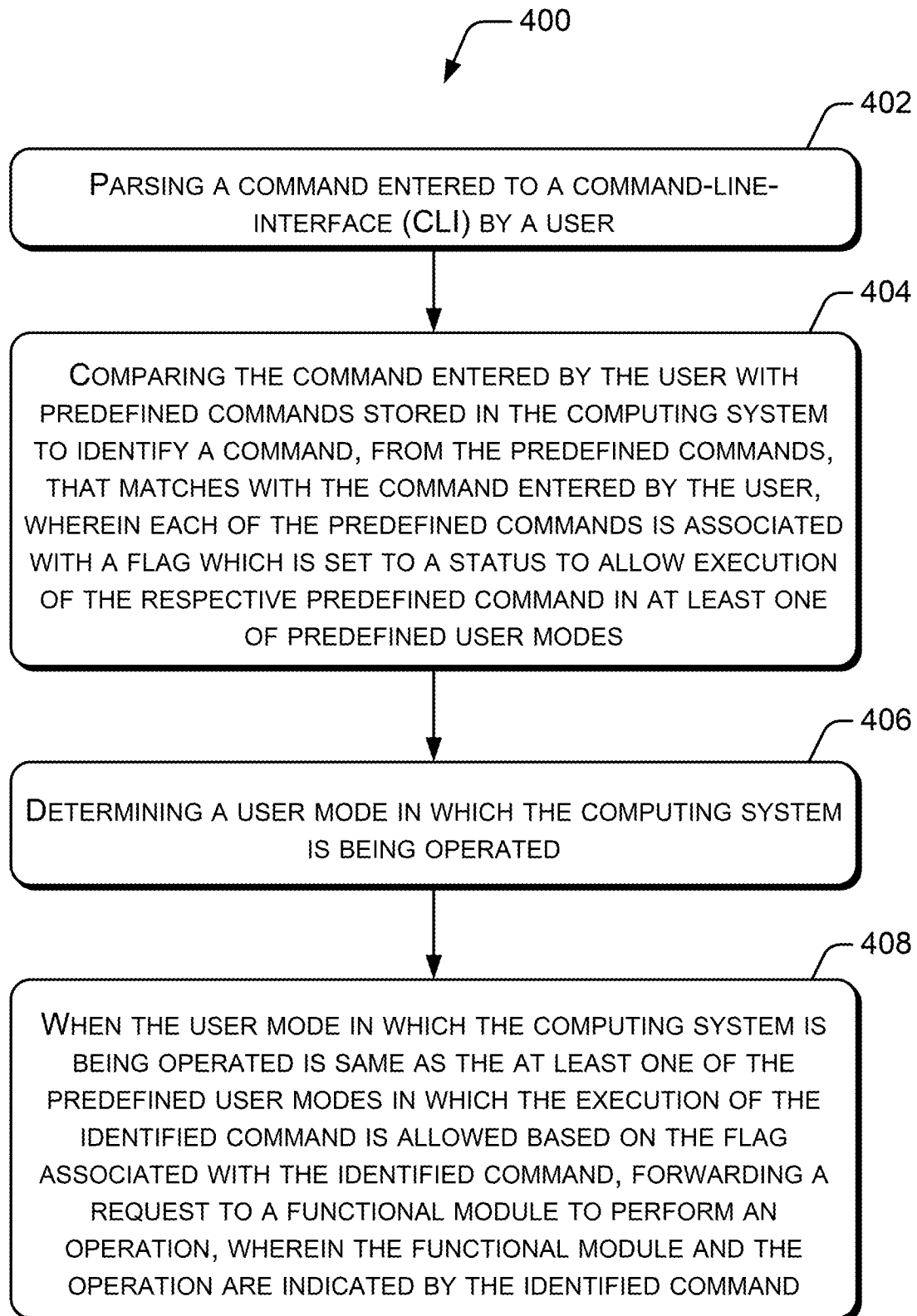


Fig. 4

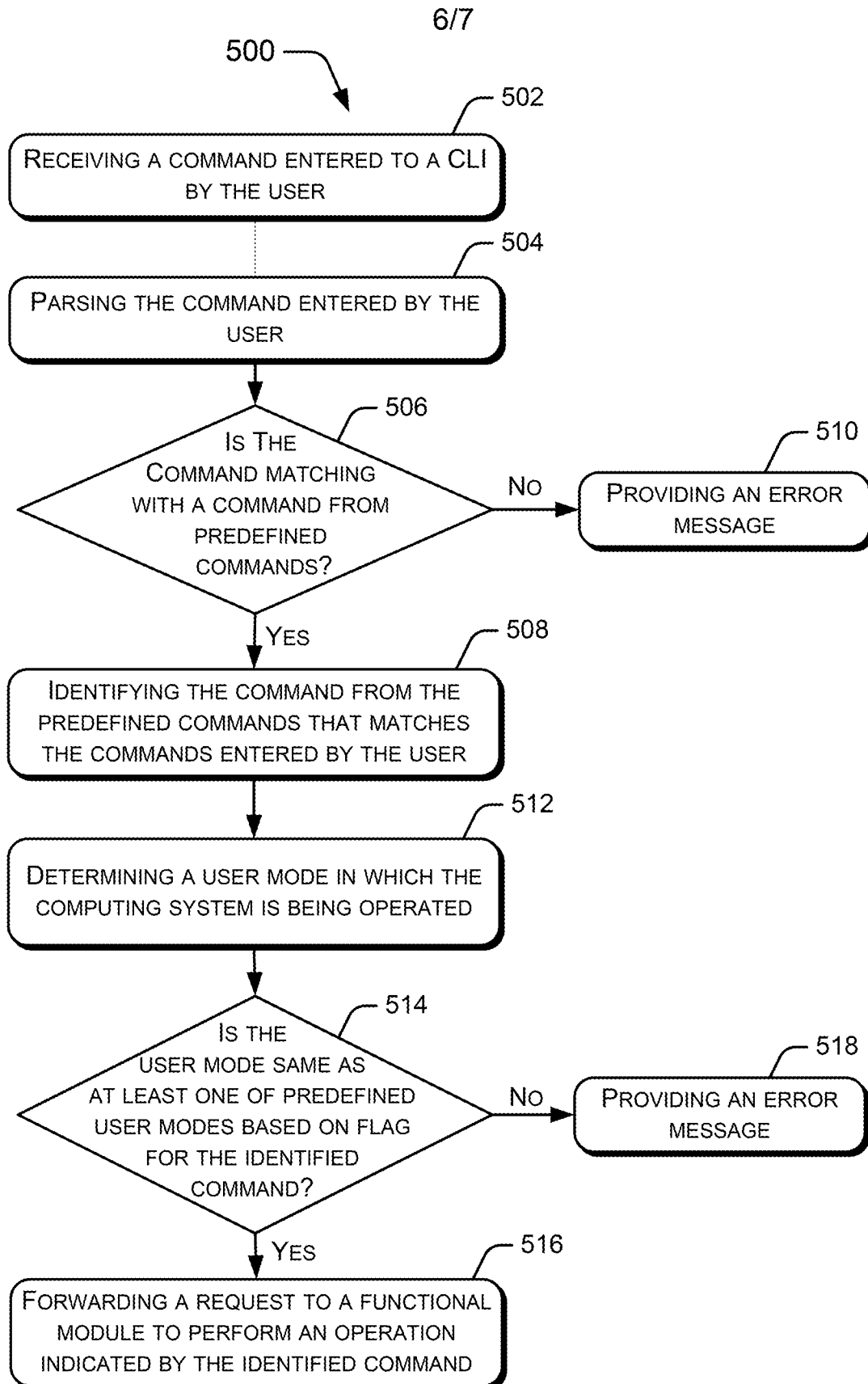


Fig. 5

7/7

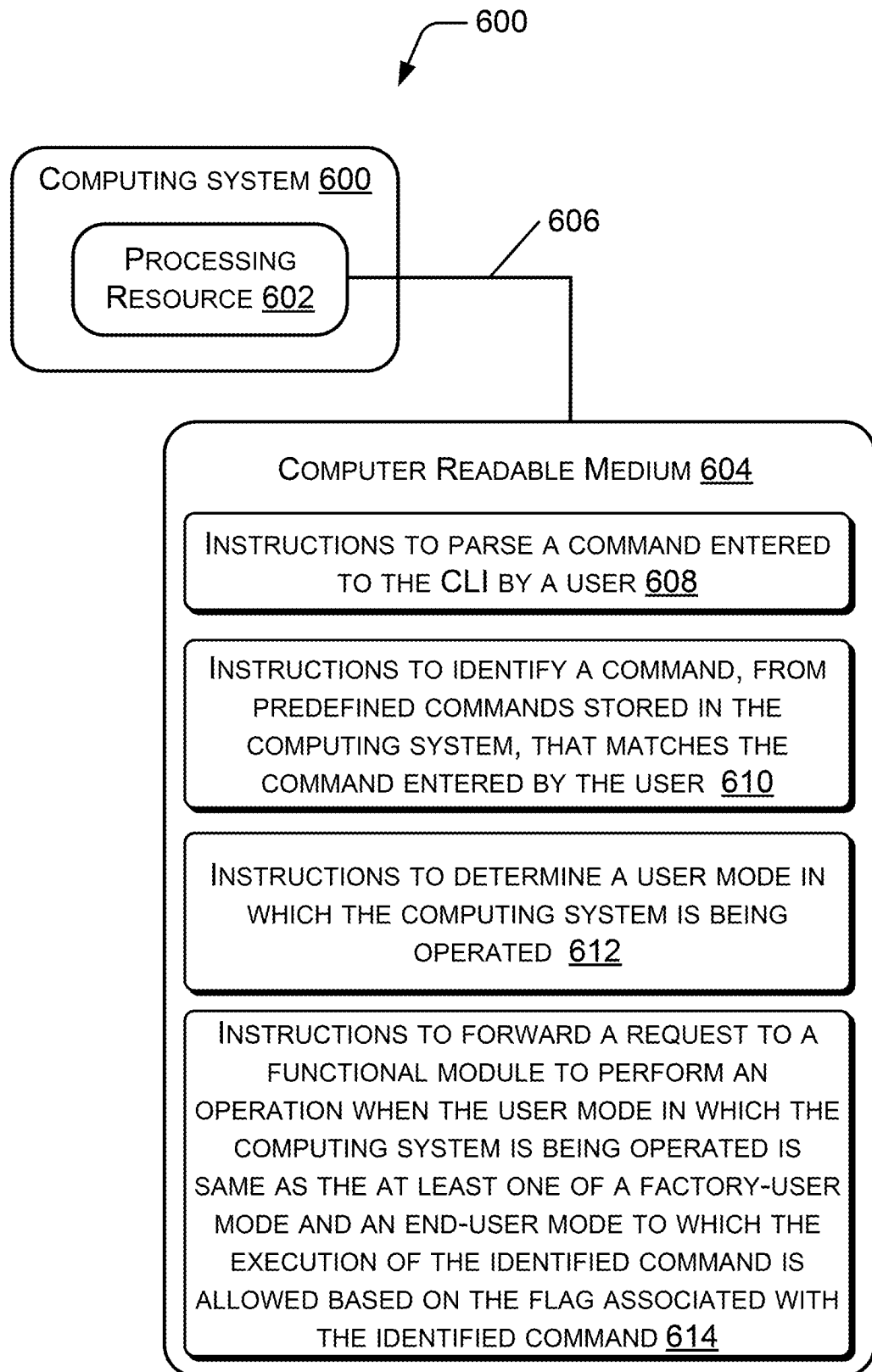


Fig. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/018555**A. CLASSIFICATION OF SUBJECT MATTER****G06F 9/30(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 9/30; G06F 15/173; H04L 9/00; G06F 9/45

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: command line interface

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2005-0132043 A1 (GONG-QIAN WANG et al.) 16 June 2005 See abstract, paragraphs [0010]-[0013], and figures 1-3.	1-15
A	US 2005-0091525 A1 (JEFFREY SNOVER et al.) 28 April 2005 See abstract, claims 1-8, and figures 2-4.	1-15
A	US 2005-0289525 A1 (MATTHEW DAVID) 29 December 2005 See abstract, paragraphs [0034]-[0075], and figures 2-4.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

19 October 2015 (19.10.2015)

Date of mailing of the international search report

20 October 2015 (20.10.2015)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

JI, Jeong Hoon

Telephone No. +82-42-481-5688



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/018555

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005-0132043 A1	16/06/2005	TW I289388B	01/11/2007
US 2005-0091525 A1	28/04/2005	AU 2004-279165 A1	23/06/2005
		AU 2004-279165 A8	19/05/2005
		AU 2004-279165 A8	02/10/2008
		AU 2004-279165 B2	11/03/2010
		BR PI0406420A	04/10/2005
		CA 2501364 A1	24/04/2005
		CA 2501364 C	03/01/2012
		CN101073057 A	14/11/2007
		CN101073057 B	13/06/2012
		EP 1588242 A2	26/10/2005
		EP 1588242 A4	21/01/2009
		JP 2007-509407 T	12/04/2007
		JP 2007-509407A	12/04/2007
		KR 10-1150059 B1	03/07/2012
		KR 2006-0114617 A	07/11/2006
		MX PA05006616A	16/08/2005
		RU2005115974 A	20/01/2006
		RU2395837 C2	27/07/2010
		US 7640540 B2	29/12/2009
		WO 2005-045565 A2	19/05/2005
		WO 2005-045565 A3	20/07/2006
US 2005-0289525 A1	29/12/2005	US 7617492 B2	10/11/2009