Item 301

git_comments:

1. Returns OK if the path exists.
2. Checks if the file is a directory. Returns an error if it doesn't exist, otherwise writes true or false into 'is_dir' appropriately.
3. Canonicalize 'path' by applying the following conversions: - Converts a relative path into an absolute one using the cwd. - Converts '.' and '..' references. - Resolves all symbolic links. All directory entries in 'path' must exist on the filesystem.
4. namespace doris
5. DEPRECATED: Use gscoped_ptr<C, doris::FreeDeleter> instead.
6. gscoped_ptr<int, doris::FreeDeleter> foo_ptr(
7. It's a normal directory.
8. Maybe a file or a symlink. Let's try to follow the symlink.
9. It's a symlink to a directory.
10. Create directory of dir_path with default Env, Create directory of dir_path, This function will create directory recursively, if dir's parent directory doesn't exist RETURNS: Status::OK() if create directory success or directory already exists
11. check path(file or directory) exist with default env
12. Delete dir or file, failed when there are files or dirs under the path
13. check path(file or directory) exist with env
14. Check the file_path is not exist with default env, or is not a dir, return false.
15. List all dirs and files in the specified directory
16. remove
17. remove paths
18. normal
19. remove_all

git_commits:

1. **summary:** Refactor and reorganize the file utils (#2089)
   message: Refactor and reorganize the file utils (#2089)

github_issues:

1. **title:** Use Env to do file operation
   body: Now, in many place, we use posix interface or boost function to do file operation. We have abstract an Env interface to do file operation. We should change it to use Env to do file operation for following reason. 1. we can unify all file operation in one place, it is easy for us to do some work later, for example audit. 2. we can use one copy of code to do the same things to avoid duplicated code. 3. If we want to our code to access remote file, we will do it without logical code modify. There are too many places to be changed, so this work will be split into some smaller job to finish.
   label: code-design
2. **title:** Use Env to do file operation
   body: Now, in many place, we use posix interface or boost function to do file operation. We have abstract an Env interface to do file operation. We should change it to use Env to do file operation for following reason. 1. we can unify all file operation in one place, it is easy for us to do some work later, for example audit. 2. we can use one copy of code to do the same things to avoid duplicated code. 3. If we want to our code to access remote file, we will do it without logical code modify. There are too many places to be changed, so this work will be split into some smaller job to finish.

github_issues_comments:

github_pulls:

1. **title:** Change refactor and reorganize the file utils
   body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
2. **title:** Change refactor and reorganize the file utils

body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
label: code-design

3. **title:** Change refactor and reorganize the file utils
   body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
   label: code-design
4. **title:** Change refactor and reorganize the file utils
   body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
5. **title:** Change refactor and reorganize the file utils
   body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
6. **title:** Change refactor and reorganize the file utils
   body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
7. **title:** Change refactor and reorganize the file utils
   body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
8. **title:** Change refactor and reorganize the file utils
   body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
   label: code-design
9. **title:** Change refactor and reorganize the file utils
   body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
10. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
11. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
12. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
13. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
14. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
15. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
    label: code-design
16. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
17. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
    label: code-design
18. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
    label: code-design
19. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)

20. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
21. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
22. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
23. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
24. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
25. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
26. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
27. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
28. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
    label: code-design
29. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
30. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
31. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)
32. **title:** Change refactor and reorganize the file utils
    body: Change remove unused file method (#2001) Change refactor and reorganize the create_file method (#2001)

github_pulls_comments:

1. @Seaven LGTM, however you should rebase the master branch to resolve the conflicts.

github_pulls_reviews:

1. **body:** This interface is to create a directory, not create a directory recursively. And create directory recursively should be implemented in file_utils, because it's the same logic for different types of Env.
   label: code-design
2. Thanks, will modify it.
3. ```suggestion std::unique_ptr<char[], kudu::FreeDeleter> r(realpath(path.c_str(), nullptr)); ``` kudu?
4. ```suggestion if (is_dot_or_dotdot(name)) { ```
5. ```suggestion auto st = env->is_directory(temp_path, &is_dir); if (st.ok()) { ```
6. **body:** better to print st's error message
   label: code-design
7. ```suggestion if (dirs != nullptr) { ```
8. ```suggestion RETURN_IF_ERROR(remove(p)); ```
9. ```suggestion RETURN_IF_ERROR(env->is_directory(path, &is_dir)); ```

10. ```suggestion if (r == nullptr) { ```
11. ```suggestion RETURN_IF_ERROR(env->is_directory(partial_path, &is_dir)); ``` better to return instantly when meeting error.
12. Yeah, the namespace is kudu in gutil/gscoped_ptr.h , maybe we should modify to doris?
13. **body:** better to change it to doris for easy use
    label: code-design
14. We can't return here because the path maybe isn't exists.
15. **body:** better to get status, and print error message.
    label: code-design
16. **body:** I think you should refactor this function to make it return error if status is no NOT_FOUND. And this may lead many changes, you can resolve this in another ISSUE
    label: code-design
17. If this path is normal file, we should return ERROR. No need to call another create_dir
18. get the status and log it if it is not OK we prefer not do operation and check in `if` clause
19. Recorded ISSUE #2166. Will modify later
20. ```suggestion Status FileUtils::list_dirs_files(const std::string& path, std::set<std::string>* dirs, ```
21. ```suggestion std::set<std::string>* files, Env* env) { ```
22. ```suggestion } else if (files != nullptr) { ```
23. ```suggestion bool FileUtils::is_dir(const std::string& file_path, Env* env) { ```
24. ```suggestion for (auto& p : paths) { ``` To avoid memory copy
25. ```suggestion Status FileUtils::remove(const std::string& path, doris::Env* env) { ```
26. **body:** I think we can call `env->delete_dir()` directly. If calling `FileUtils::remove`, it will check that it is a directory or ordinal file, which is unnecessary.
    label: code-design
27. Better to check this function's result
28. should set ret to an ERROR
29. set res to error

jira_issues:

jira_issues_comments: