#include <dirent.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

#include <stdlib.h>

#include <string.h>

#include <string>

#include <memory>

#include <vector>

#include <stdlib.h>

#include <malloc.h>

#define PATH "/home/consta/practice1/1/Test1/"

#define BUFFER\_SIZE 256

int empty\_file\_number = 1;

bool is\_pseudodirectory(struct dirent \*ent)

{

return strcmp(ent->d\_name, (".")) == 0 || strcmp(ent->d\_name, ("..")) == 0;

}

void get\_old\_name(char \*path, struct dirent \*ent, char \*old\_file\_name)

{

sprintf(old\_file\_name, "%s%s", path, ent->d\_name);

}

void get\_new\_name(char \*path, struct dirent \*ent, char \*new\_file\_name)

{

sprintf(new\_file\_name, "%s%s%d%s",

path, "zero", empty\_file\_number, ".txt");

}

// переименовывание файла

bool rename\_file(struct dirent \*ent, char \*path)

{

// старое имя

char old\_file\_name[BUFFER\_SIZE] = "";

get\_old\_name(path, ent, old\_file\_name);

// новое имя

char new\_file\_name[BUFFER\_SIZE] = "";

get\_new\_name(path, ent, new\_file\_name);

printf("Success\_rename\n");

// типа переименовывает

return rename(old\_file\_name, new\_file\_name);

}

void get\_new\_path(char \*path, struct dirent \*ent, char \*new\_path)

{

sprintf(new\_path, "%s%s%s", path, ent->d\_name, "/");

}

bool is\_directory(char \*path\_, struct stat \*st)

{

stat(path\_, st);

return S\_ISDIR(st->st\_mode);

}

void search\_file(char \*path)

{

DIR \*dir = opendir(path);

// printf("%s","0");

while (struct dirent \*ent = readdir(dir))

{

struct stat sb;

char file\_name[BUFFER\_SIZE] = "";

get\_old\_name(path, ent, file\_name);

// printf("%s","1");

if (stat(file\_name, &sb) == -1) // если работает текущий файл записывается в sb

{

continue;

}

// printf("%s","2");

if ((sb.st\_mode & S\_IFMT) != S\_IFREG) // если регулярный файл

{

continue;

}

// printf("%s","3");

if (is\_pseudodirectory(ent))

{

continue;

}

// printf("%s","4");

if (sb.st\_size != 0) // размер файла

{

continue;

}

// printf("%s","5");

rename\_file(ent, path);

empty\_file\_number++;

}

closedir(dir);

}

void rec\_dir(char \*path) // DIRECTORY\_NAME

{

printf("%s", "odin");

search\_file(path); // изначально перебираю файлы в папке -> потом рассматриваю каталоги

DIR \*dir = opendir(path);

std::vector<std::unique\_ptr<char[]>> paths;

while (struct dirent \*ent = readdir(dir))

{

struct stat sb;

printf("%s", "dva");

char dir\_path[BUFFER\_SIZE] = "";

get\_old\_name(path, ent, dir\_path);

if (!(is\_directory(dir\_path, &sb))) // если директория

{

continue;

} // директория

printf("%s", "tri");

if (is\_pseudodirectory(ent)) // не точки

{

continue;

} // не псевдодиректория

printf("%s", "chert");

paths.emplace\_back(new char[BUFFER\_SIZE]()); // мусор в вектор

get\_new\_path(path, ent, paths.back().get());

rec\_dir(paths.back().get());

}

closedir(dir); // поиск окончен

}

int main(int argc, char \*argv[])

{ // printf("%s","-1");

rec\_dir(PATH);

return 0;

}