



福建師範大學
FUJIAN NORMAL UNIVERSITY

计算机与网络空间安全学院学生实验报告

实验课程名称： 大数据导论 教师： 林鑫泓

实验名称	熟悉常用的 HDFS 操作			实验成绩	
学生姓名	叶建行	学号	116052020005	年级专业 班级	2020 级软件工程 一班
小组成员	无			实验日期	2022 年 10 月 12

1 实验目的和要求

1.1 实验目的

- 理解 HDFS 在 Hadoop 体系结构中的角色，
- 熟练使用 HDFS 操作常用的 Shell 命令，
- 熟悉 HDFS 操作常用的 Java API。

1.2 实验软硬件环境

- 操作系统：Linux（建议 Ubuntu18.04），
- JDK 版本：建议 openjdk-8，
- Hadoop 版本：建议 3.2.1。

1.3 实验要求

- 理解 HDFS 在 Hadoop 体系结构中的角色，
- 熟练使用 HDFS 操作常用的 Shell 命令，
- 熟悉 HDFS 操作常用的 Java API。

2 实验记录

（一）编程实现以下功能，并利用 Hadoop 提供的 Shell 命令完成相同任务：

（1）向 HDFS 中上传任意文本文件，如果指定的文件在 HDFS 中已经存在，则由用户来指定是追加到原有文件末尾还是覆盖原有的文件；

实验 shell 脚本代码：（put_file.sh）

```
#!/bin/sh
if $(./bin/hdfs dfs -test -e $1);then
    echo "选择覆盖原文件或在原文件基础上追加"
    echo "1.覆盖"
    echo "2.追加"
    read ans
    if [ $ans = "1" ];then
        ./bin/hdfs dfs -copyFromLocal -f $1 $1
    elif [ $ans = "2" ];then
        ./bin/hdfs dfs -appendToFile $1 $1
    else
        echo "输入错误，请输入1或2"
    fi
else
    ./bin/hdfs dfs -copyFromLocal -f $1 $1
    echo "成功上传文件"
fi
~
~
~
~
~
```

实验结果：

1. 当指定的文件在 HDFS 中不存在，上传文件

```
[root@master .hadoop]# ./bin/hdfs dfs -ls .
Found 1 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:54 local.txt
[root@master .hadoop]# ./put_file.sh text.txt
成功上传文件
[root@master .hadoop]# ./bin/hdfs dfs -ls .
Found 2 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:54 local.txt
-rw-r--r-- 1 root supergroup 3 2022-10-02 20:04 text.txt
[root@master .hadoop]# ./bin/hdfs dfs -cat text.txt
hi
```

2. 当指定文件在 HDFS 存在，用户选择执行覆盖文件或追加文件内容
（注意下面对本地的 text.txt 文件进行了内容修改）

```
[root@master .hadoop]# echo "hello">text.txt
[root@master .hadoop]# cat text.txt
hello
[root@master .hadoop]# ./put_file.sh text.txt
选择覆盖原文件或在原文件基础上追加
1.覆盖
2.追加
1
[root@master .hadoop]# ./bin/hdfs dfs -cat text.txt
hello
[root@master .hadoop]# ./put_file.sh text.txt
选择覆盖原文件或在原文件基础上追加
1.覆盖
2.追加
2
[root@master .hadoop]# ./bin/hdfs dfs -cat text.txt
hello
hello
```

实验 java 代码:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.io.*;
import java.util.Scanner;
public class HDFSApi {

    public static boolean test(Configuration conf, String path) throws
    IOException {
        FileSystem fs = FileSystem.get(conf);
        return fs.exists(new Path(path));
    }
    /**
     * 复制文件到指定路径
     * 若路径已存在, 则进行覆盖
     */
    public static void copyFromLocalFile(Configuration conf, String
    localFilePath, String remoteFilePath) throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path localPath = new Path(localFilePath);
        Path remotePath = new Path(remoteFilePath);
        /* fs.copyFromLocalFile 第一个参数表示是否删除源文件, 第二个参数表示是否
    覆
        盖 */
        fs.copyFromLocalFile(false, true, localPath, remotePath);
        fs.close();
    }
    /**
     * 追加文件内容
     */
    public static void appendToFile(Configuration conf, String
    localFilePath, String remoteFilePath) throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path remotePath = new Path(remoteFilePath);
        /* 创建一个文件读入流 */
        FileInputStream in = new FileInputStream(localFilePath);
        /* 创建一个文件输出流, 输出的内容将追加到文件末尾 */
        FSDataOutputStream out = fs.append(remotePath);
        /* 读写文件内容 */
        byte[] data = new byte[1024];
        int read = -1;
        while ( (read = in.read(data)) > 0 ) {
            out.write(data, 0, read);
        }
    }
}
```

```
        out.close();
        in.close();
        fs.close();
    }

    /**
    //
    * 主函数
    */
    public static void main(String[] args) {
        Configuration conf = new Configuration();
        conf.set("fs.default.name", "hdfs://localhost:9000");
        Scanner input = new Scanner(System.in);
        System.out.println("请输入要上传文件的本地路径");
        String localFilePath = input.next();
        // 本地路径
        System.out.println("请输入要上传文件的 HDFS 路径");
        String remoteFilePath = input.next();
        // HDFS 路径
        String choice = "0";
        try {
            /* 判断文件是否存在 */
            Boolean fileExists = false;
            if (HDFSApi.test(conf, remoteFilePath)) {
                fileExists = true;
                System.out.println(remoteFilePath + " 已存在.");
                System.out.println("用户输入覆盖或追加: ");
                System.out.println("1.覆盖");
                System.out.println("2.追加");
                choice = input.next();
            } else {
                System.out.println(remoteFilePath + " 不存在.");
            }
            /* 进行处理 */
            if ( !fileExists ) { // 文件不存在,则上传
                HDFSApi.copyFromLocalFile(conf, localFilePath, remoteFilePath);
                System.out.println(localFilePath + " 已上传至 " +
remoteFilePath);
            } else if ( choice.equals("1") ) {
                // 选择覆盖
                HDFSApi.copyFromLocalFile(conf, localFilePath, remoteFilePath);
                System.out.println(localFilePath + " 已覆盖 " + remoteFilePath);
            } else if ( choice.equals("2") ) { // 选择追加
                HDFSApi.appendToFile(conf, localFilePath, remoteFilePath);
```

```

        System.out.println(localFilePath + " 已追加至 " +
remoteFilePath);
    }
} catch (Exception e) {
    e.printStackTrace();
}
}
}
}

```

实验结果:

1. 当指定的文件在 HDFS 中不存在, 上传文件

```

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls .
Found 1 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:52 local.txt

(root@yek)~/Hadoop_work
# java HDFSApi
2022-10-04 16:41:45,188 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
请输入要上传文件的本地路径
/root/.hadoop/text.txt
请输入要上传文件的HDFS路径
/user/root/text.txt
2022-10-04 16:42:37,212 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
/user/root/text.txt 不存在.
/root/.hadoop/text.txt 已上传至 /user/root/text.txt

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls .
Found 2 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:52 local.txt
-rw-r--r-- 1 root supergroup 6 2022-10-04 16:42 text.txt

```

2. 当指定文件在 HDFS 存在, 用户选择执行覆盖文件或追加文件内容追加:

```

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -cat text.txt
hello

(root@yek)~/Hadoop_work
# java HDFSApi
2022-10-04 16:43:28,252 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
请输入要上传文件的本地路径
/root/.hadoop/text.txt
请输入要上传文件的HDFS路径
/user/root/text.txt
2022-10-04 16:44:00,125 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
/user/root/text.txt 已存在.
用户输入覆盖或追加:
1.覆盖
2.追加
2
/root/.hadoop/text.txt 已追加至 /user/root/text.txt

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -cat text.txt
hello
hello

```

覆盖:

```
(root@yek)~[~/Hadoop_work]
# java HDFSApi
2022-10-04 16:44:17,562 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultF
S
请输入要上传文件的本地路径
/root/.hadoop/text.txt
请输入要上传文件的HDFS路径
/user/root/text.txt
2022-10-04 16:44:41,567 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
/user/root/text.txt 已存在.
用户输入覆盖或追加:
1.覆盖
2.追加
1
/root/.hadoop/text.txt 已覆盖 /user/root/text.txt

(root@yek)~[~/Hadoop_work]
# ~/.hadoop/bin/hdfs dfs -cat text.txt
hello
```

(2) 从 HDFS 中下载指定文件, 如果本地文件与要下载的文件名称相同, 则自动对下载的文件重命名;

实验 shell 脚本代码:

```
#!/bin/sh

if [ -f $1 ]
then
    tmp=$1"1"
    for j in `ls ./`
    do
        for i in `ls ./`
        do
            if [ $tmp = $i ];then
                tmp=$tmp"1"
                break
            fi
        done
    done
    ./bin/hdfs dfs -copyToLocal $1 $tmp
    echo "本地产生了新文件"$tmp
    export tmp
else
    ./bin/hdfs dfs -copyToLocal $1 $1
    echo "本地产生了新文件"$1
fi
~
"dload_from_HDFS.sh" 22L, 365C written
```

实验结果:

1. 当本地不存在对应文件时, 下载对应文件

```
[root@master .hadoop]# ls
ans          include      licenses-binary  NOTICE-binary  sbin
bin          lib          LICENSE.txt      NOTICE.txt     share
dload_from_HDFS.sh libexec      local.txt        put_file.sh     tmp
etc          LICENSE-binary logs         README.txt
[root@master .hadoop]# ./dload_from_HDFS.sh text.txt
本地产生了新文件text.txt
[root@master .hadoop]# ls
ans          include      licenses-binary  NOTICE-binary  sbin
bin          lib          LICENSE.txt      NOTICE.txt     share
dload_from_HDFS.sh libexec      local.txt        put_file.sh     text.txt
etc          LICENSE-binary logs         README.txt     tmp
```

2. 当本地存在对应文件时，通过双重循环获取本地文件夹中唯一的文件名，并对下载文件按此命名

```
[root@master .hadoop]# ./dload_from_HDFS.sh text.txt
本地产生了新文件text.txt1
[root@master .hadoop]# ls
ans          lib          local.txt    README.txt   tmp
bin          libexec      logs         sbin
dload_from_HDFS.sh LICENSE-binary NOTICE-binary share
etc          licenses-binary NOTICE.txt   text.txt
include      LICENSE.txt  put_file.sh  text.txt1
[root@master .hadoop]# ./dload_from_HDFS.sh text.txt
本地产生了新文件text.txt11
[root@master .hadoop]# ./dload_from_HDFS.sh text.txt
本地产生了新文件text.txt111
```

实验 Java 代码:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.io.*;
import java.util.Scanner;

public class HDFSApi2 {
    /**
     * 下载文件到本地
     * 判断本地路径是否已存在, 若已存在, 则自动进行重命名
     */
    public static void copyToLocal(Configuration conf, String
remoteFilePath, String localFilePath) throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path remotePath = new Path(remoteFilePath);
        File f = new File(localFilePath);
        /* 如果文件名存在, 自动重命名(在文件名后面加上 _0, _1 ...) */
        if (f.exists()) {
            System.out.println(localFilePath + " 已存在.");
            Integer i = 0;
            while (true) {
                i = i + 1;
                f = new File(localFilePath + "_" + i.toString());
                if (!f.exists()) {
                    localFilePath = localFilePath + "_" + i.toString();
                    break;
                }
            }
            System.out.println("将重新命名为: " + localFilePath);
        }
        // 下载文件到本地
```

```

    Path localPath = new Path(localFilePath);
    fs.copyToLocalFile(remotePath, localPath);
    fs.close();
}

/**
 * 主函数
 */
public static void main(String[] args) {
    Configuration conf = new Configuration();
    conf.set("fs.default.name", "hdfs://localhost:9000");
    Scanner input = new Scanner(System.in);
    System.out.println("请输入要下载文件的 HDFS 路径");
    String remoteFilePath = input.next();
    // HDFS 路径
    System.out.println("请输入要下载文件的本地路径");
    String localFilePath = input.next();
    // 本地路径
    try {
        HDFSApi2.copyToLocal(conf, remoteFilePath, localFilePath);
        System.out.println("下载完成");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

实验结果:

1. 当本地不存在对应文件时, 下载对应文件

```

(root@yek) ~/Hadoop_work
# java HDFSApi2
2022-10-06 16:09:17,249 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
请输入要下载文件的 HDFS 路径
/user/root/hello.txt
请输入要下载文件的本地路径
/root/.hadoop/hello.txt
2022-10-06 16:09:56,313 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
下载完成

```

2. 当本地存在对应文件时, 通过循环判断得到唯一的文件名


```

[root@yek]~[~/Hadoop_work]
# java HDFSApi2
2022-10-06 16:10:02,827 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
请输入要下载文件的HDFS路径
/user/root/hello.txt
请输入要下载文件的本地路径
/root/.hadoop/hello.txt
2022-10-06 16:10:30,395 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
/root/.hadoop/hello.txt 已存在。
将重新命名为: /root/.hadoop/hello.txt_1
下载完成

[root@yek]~[~/Hadoop_work]
# java HDFSApi2
2022-10-06 16:10:37,052 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
请输入要下载文件的HDFS路径
/user/root/hello.txt
请输入要下载文件的本地路径
/root/.hadoop/hello.txt
2022-10-06 16:10:56,737 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
/root/.hadoop/hello.txt 已存在。
将重新命名为: /root/.hadoop/hello.txt_2
下载完成

```

(3) 将 HDFS 中指定文件的内容输出到终端中;

```

[root@master .hadoop]# ./bin/hdfs dfs -cat text.txt
hello
hello

```

实验 Java 代码:

```

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.io.*;
import java.util.Scanner;

public class HDFSApi3 {
    /**
     * 读取文件内容
     */
    public static void cat(Configuration conf, String remoteFilePath)
        throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path remotePath = new Path(remoteFilePath);
        FSDataInputStream in = fs.open(remotePath);
        BufferedReader d = new BufferedReader(new InputStreamReader(in));
        String line = null;
        while ( (line = d.readLine()) != null ) {
            System.out.println(line);
        }
        d.close();
        in.close();
        fs.close();
    }
    /**
     * 主函数
     */
}

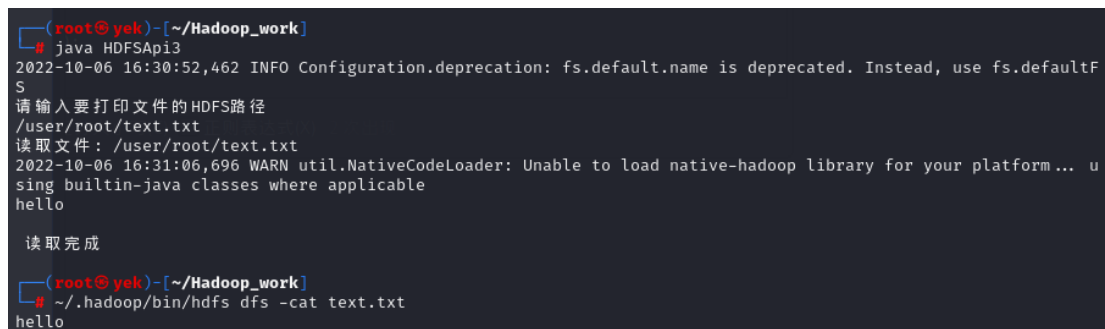
```

```

public static void main(String[] args) {
    Configuration conf = new Configuration();
    conf.set("fs.default.name", "hdfs://localhost:9000");
    Scanner input = new Scanner(System.in);
    System.out.println("请输入要打印文件的 HDFS 路径");
    String remoteFilePath = input.next();
    // HDFS 路径
    try {
        System.out.println("读取文件: " + remoteFilePath);
        HDFSApi3.cat(conf, remoteFilePath);
        System.out.println("\n 读取完成");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

实验结果:



```

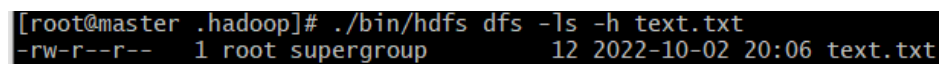
(root@yek)~/Hadoop_work
# java HDFSApi3
2022-10-06 16:30:52,462 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultF
S
请输入要打印文件的HDFS路径
/user/root/text.txt
读取文件: /user/root/text.txt
2022-10-06 16:31:06,696 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
hello

读取完成

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -cat text.txt
hello

```

(4) 显示 HDFS 中指定的文件的读写权限、大小、创建时间、路径等信息;



```

[root@master ~]# ./bin/hdfs dfs -ls -h text.txt
-rw-r--r--  1 root supergroup      12 2022-10-02 20:06 text.txt

```

实验 Java 代码:

```

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.Scanner;
public class HDFSApi4 {
    /**
     * 显示指定文件的信息
     */
    public static void ls(Configuration conf, String remoteFilePath) throws
    IOException {
        FileSystem fs = FileSystem.get(conf);
    }
}

```

```

    Path remotePath = new Path(remoteFilePath);
    FileStatus[] fileStatuses = fs.listStatus(remotePath);
    for (FileStatus s : fileStatuses) {
        System.out.println("路径: " + s.getPath().toString());
        System.out.println("权限: " + s.getPermission().toString());
        System.out.println("大小: " + s.getLength());
        /* 返回的是时间戳, 转化为时间日期格式 */
        Long timeStamp = s.getModificationTime();
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
        String date = format.format(timeStamp);
        System.out.println("时间: " + date);
    }
    fs.close();
}

public static void main(String[] args) {
    Configuration conf = new Configuration();
    conf.set("fs.default.name", "hdfs://localhost:9000");
    Scanner input = new Scanner(System.in);
    System.out.println("请输入要显示文件的 HDFS 路径");
    String remoteFilePath = input.next();
    // HDFS 路径
    try {
        System.out.println("读取文件信息: " + remoteFilePath);
        HDFSApi4.ls(conf, remoteFilePath);
        System.out.println("\n 读取完成");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

实验结果:

```

(root@yek) ~/Hadoop_work
# java HDFSApi4
2022-10-06 16:39:26,156 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultF
S
请输入要显示文件的HDFS路径
/user/root/text.txt
读取文件信息: /user/root/text.txt
2022-10-06 16:39:36,504 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
路径: hdfs://localhost:9000/user/root/text.txt
权限: rw-r--r--
大小: 6
时间: 2022-10-04 16:44:43

读取完成

(root@yek) ~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls text.txt
-rw-r--r--    1 root supergroup      6 2022-10-04 16:44 text.txt

```

(5) 给定 HDFS 中某一个目录，输出该目录下的所有文件的读写权限、大小、创建时间、路径等信息，如果该文件是目录，则递归输出该目录下所有文件相关信息；

```
[root@master .hadoop]# ./bin/hdfs dfs -ls /user/root
Found 2 items
-rw-r--r--  1 root supergroup      0 2022-10-02 15:54 /user/root/local.txt
-rw-r--r--  1 root supergroup    12 2022-10-02 20:06 /user/root/text.txt
```

实验 Java 代码：

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.Scanner;
public class HDFSApi5 {
    /**
     * 显示指定文件夹下所有文件的信息(递归)
     */
    public static void lsDir(Configuration conf, String remoteDir) throws
    IOException {
        FileSystem fs = FileSystem.get(conf);
        Path dirPath = new Path(remoteDir);
        /* 递归获取目录下的所有文件 */
        RemoteIterator<LocatedFileStatus> remoteIterator =
fs.listFiles(dirPath, true);
        /* 输出每个文件的信息 */
        while (remoteIterator.hasNext()) {
            FileStatus s = remoteIterator.next();
            System.out.println("路径: " + s.getPath().toString());
            System.out.println("权限: " + s.getPermission().toString());
            System.out.println("大小: " + s.getLen());
            /* 返回的是时间戳, 转化为时间日期格式 */
            Long timeStamp = s.getModificationTime();
            SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
            String date = format.format(timeStamp);
            System.out.println("时间: " + date);
            System.out.println();
        }
        fs.close();
    }
    /**
     * 主函数
     */
    public static void main(String[] args) {
        Configuration conf = new Configuration();
```

```

conf.set("fs.default.name", "hdfs://localhost:9000");
Scanner input = new Scanner(System.in);
System.out.println("请输入要显示目录的 HDFS 路径");
String remoteDir = input.next();
// HDFS 路径
try {
    System.out.println("(递归)读取目录下所有文件的信息: " + remoteDir);
    HDFSApi5.lsDir(conf, remoteDir);
    System.out.println("读取完成");
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

实验结果:

```

(root@yek)~/Hadoop_work
# java HDFSApi5
2022-10-06 16:53:32,628 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultF
S
请输入要显示目录的HDFS路径
/user/root
(递归)读取目录下所有文件的信息: /user/root
2022-10-06 16:53:41,099 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
路径: hdfs://localhost:9000/user/root/hello.txt
权限: rw-r--r--
大小: 0
时间: 2022-10-06 16:03:34

路径: hdfs://localhost:9000/user/root/local.txt
权限: rw-r--r--
大小: 0
时间: 2022-10-02 15:52:41

路径: hdfs://localhost:9000/user/root/text.txt
权限: rw-r--r--
大小: 6
时间: 2022-10-04 16:44:43

读取完成

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls /user/root
Found 3 items
-rw-r--r-- 1 root supergroup 0 2022-10-06 16:03 /user/root/hello.txt
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:52 /user/root/local.txt
-rw-r--r-- 1 root supergroup 6 2022-10-04 16:44 /user/root/text.txt

```

(6) 提供一个 HDFS 内的文件的路径, 对该文件进行创建和删除操作。
如果文件所在目录不存在, 则自动创建目录;

创建文件:

```

[root@master .hadoop]# if $(./bin/hdfs dfs -test -d /user/root)
> then
> ./bin/hdfs dfs -touchz /user/root/hello.txt
> else
> ./bin/hdfs dfs -mkdir -p /user/root
> ./bin/hdfs dfs -touchz /user/root/hello.txt
> fi
[root@master .hadoop]# ./bin/hdfs dfs -ls /user/root
Found 3 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 23:09 /user/root/hello.txt
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:54 /user/root/local.txt
-rw-r--r-- 1 root supergroup 12 2022-10-02 20:06 /user/root/text.txt

```

删除文件:

```
[root@master .hadoop]# ./bin/hdfs dfs -rm /user/root/hello.txt
Deleted /user/root/hello.txt
[root@master .hadoop]# ./bin/hdfs dfs -ls /user/root
Found 2 items
-rw-r--r--  1 root supergroup          0 2022-10-02 15:54 /user/root/local.txt
-rw-r--r--  1 root supergroup        12 2022-10-02 20:06 /user/root/text.txt
```

实验 Java 代码:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.io.*;

public class HDFSApi6 {
    /**
     * 判断路径是否存在
     */
    public static boolean test(Configuration conf, String path) throws
        IOException {
        FileSystem fs = FileSystem.get(conf);
        return fs.exists(new Path(path));
    }
    /**
     * 创建目录
     */
    public static boolean mkdir(Configuration conf, String remoteDir)
        throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path dirPath = new Path(remoteDir);
        boolean result = fs.mkdirs(dirPath);
        fs.close();
        return result;
    }
    /**
     * 创建文件
     */
    public static void touchz(Configuration conf, String remoteFilePath)
        throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path remotePath = new Path(remoteFilePath);
        FSDataOutputStream outputStream = fs.create(remotePath);
        outputStream.close();
        fs.close();
    }
}
```

```

* 删除文件
*/
public static boolean rm(Configuration conf, String remoteFilePath)
throws IOException {
    FileSystem fs = FileSystem.get(conf);
    Path remotePath = new Path(remoteFilePath);
    boolean result = fs.delete(remotePath, false);
    fs.close();
    return result;
}

public static void main(String[] args) {
    Configuration conf = new Configuration();
    conf.set("fs.default.name", "hdfs://localhost:9000");
    String remoteFilePath = "/user/hadoop/input/text.txt";
    // HDFS 路径
    String remoteDir = "/user/hadoop/input";
    // HDFS 路径对应的目录
    try {
        /* 判断路径是否存在, 存在则删除, 否则进行创建 */
        if ( HDFSApi6.test(conf, remoteFilePath) ) {
            HDFSApi6.rm(conf, remoteFilePath); // 删除
            System.out.println("删除路径: " + remoteFilePath);
        } else {
            if ( !HDFSApi6.test(conf, remoteDir) ) { // 若目录不存在, 则进行创建
                HDFSApi6.mkdir(conf, remoteDir);
                System.out.println("创建文件夹: " + remoteDir);
            }
            HDFSApi6.touchz(conf, remoteFilePath);
            System.out.println("创建路径: " + remoteFilePath);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

实验结果:

```

[root@yek] ~/Hadoop_work
# java HDFSApi6
2022-10-06 17:09:10,457 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
2022-10-06 17:09:10,569 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
创建文件夹: /user/hadoop/input
创建路径: /user/hadoop/input/text.txt

[root@yek] ~/Hadoop_work
# java HDFSApi6
2022-10-06 17:13:11,621 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
2022-10-06 17:13:11,728 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
删除路径: /user/hadoop/input/text.txt

```

(7) 提供一个 **HDFS** 的目录的路径, 对该目录进行创建和删除操作。创建目录时, 如果目录文件所在目录不存在, 则自动创建相应目录; 删除目录时, 由用户指定当该目录不为空时是否还删除该目录;

创建目录:

1. 目录不存在时，自动创建目录

```
[root@master .hadoop]# ./bin/hdfs dfs -mkdir -p /user/hello
[root@master .hadoop]# ./bin/hdfs dfs -ls /user
Found 2 items
drwxr-xr-x - root supergroup          0 2022-10-03 10:10 /user/hello
drwxr-xr-x - root supergroup          0 2022-10-02 23:11 /user/root
```

2. 目录存在时，不进行任何操作

```
[root@master .hadoop]# ./bin/hdfs dfs -mkdir -p /user/root
[root@master .hadoop]# ./bin/hdfs dfs -ls /user
Found 2 items
drwxr-xr-x - root supergroup          0 2022-10-03 10:10 /user/hello
drwxr-xr-x - root supergroup          0 2022-10-02 23:11 /user/root
```

删除目录:

实验代码:

[illegible]

实验结果:

1. 当目录为空时，直接删除目录

```
[root@master .hadoop]# ./bin/hdfs dfs -ls /user
Found 2 items
drwxr-xr-x - root supergroup          0 2022-10-03 10:40 /user/hello
drwxr-xr-x - root supergroup          0 2022-10-02 23:11 /user/root
[root@master .hadoop]# ./del.sh /user/hello
[root@master .hadoop]# ./bin/hdfs dfs -ls /user
Found 1 items
drwxr-xr-x - root supergroup          0 2022-10-02 23:11 /user/root
```


2. 目录不为空时，用户选定是否除目录

```
[root@master .hadoop]# ./bin/hdfs dfs -mkdir /user/hello
[root@master .hadoop]# ./bin/hdfs dfs -touchz /user/hello/text.txt
[root@master .hadoop]# ./del.sh /user/hello
目录不为空，是否要强制删除该目录(yes/no)
no
[root@master .hadoop]# ./bin/hdfs dfs -ls /user/hello
Found 1 items
-rw-r--r--  1 root supergroup      0 2022-10-03 10:53 /user/hello/text.txt
[root@master .hadoop]# ./del.sh /user/hello
目录不为空，是否要强制删除该目录(yes/no)
yes
Deleted /user/hello
[root@master .hadoop]# ./bin/hdfs dfs -ls /user
Found 1 items
drwxr-xr-x  - root supergroup      0 2022-10-02 23:11 /user/root
```

实验 Java 代码:

```
import java.io.*;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.util.Scanner;

public class HDFSApi7 {
    /**
     * 判断路径是否存在
     */
    public static boolean test(Configuration conf, String path) throws
IOException {
        FileSystem fs = FileSystem.get(conf);
        return fs.exists(new Path(path));
    }

    /**
     * 判断目录是否为空
     * true: 空, false: 非空
     */
    public static boolean isDirEmpty(Configuration conf, String
remoteDir) throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path dirPath = new Path(remoteDir);
        RemoteIterator<LocatedFileStatus> remoteIterator =
fs.listFiles(dirPath, true);
        return !remoteIterator.hasNext();
    }

    /**
     * 创建目录
     */
}
```

```

    public static boolean mkdir(Configuration conf, String remoteDir)
throws IOException {
    FileSystem fs = FileSystem.get(conf);
    Path dirPath = new Path(remoteDir);
    boolean result = fs.mkdirs(dirPath);
    fs.close();
    return result;
}

/**
 * 删除目录
 */
    public static boolean rmDir(Configuration conf, String remoteDir)
throws IOException {
    FileSystem fs = FileSystem.get(conf);
    Path dirPath = new Path(remoteDir);
    /* 第二个参数表示是否递归删除所有文件 */
    boolean result = fs.delete(dirPath, true);
    fs.close();
    return result;
}

/**
 * 主函数
 */
    public static void main(String[] args) {
        Configuration conf = new Configuration();
        conf.set("fs.default.name", "hdfs://localhost:9000");
        String remoteDir = "/user/hadoop/input";
        Scanner input = new Scanner(System.in);
        // HDFS 目录
        try {
            /* 判断目录是否存在, 不存在则创建, 存在则删除 */
            if (!HDFSApi7.test(conf, remoteDir)) {
                HDFSApi7.mkdir(conf, remoteDir); // 创建目录
                System.out.println("创建目录: " + remoteDir);
            } else {
                if (HDFSApi7.isDirEmpty(conf, remoteDir)) { // 目录为空
                    HDFSApi7.rmDir(conf, remoteDir);
                    System.out.println("删除目录: " + remoteDir);
                } else { // 目录不为空
                    System.out.println("目录不为空, 是否要强制删除目录
(yes/no)");
                    String ans = input.next();

```

```

        if (ans.equals("yes")) {
            HDFSApi7.rmDir(conf, remoteDir);
            System.out.println("删除目录: " + remoteDir);
        } else {
            System.out.println("目录不为空,不删除: " +
remoteDir);
        }
    }
}
} catch (Exception e) {
    e.printStackTrace();
}
}
}
}

```

实验结果:

创建目录:

1. 目录不存在时, 自动创建目录

```

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls /user/hadoop
Found 2 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:48 /user/hadoop/local.txt
-rw-r--r-- 1 root supergroup 0 2022-10-11 17:01 /user/hadoop/text.txt

(root@yek)~/Hadoop_work
# java HDFSApi7
2022-10-11 17:13:27,250 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultF
S
2022-10-11 17:13:27,349 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
创建目录: /user/hadoop/input

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls /user/hadoop
Found 3 items
drwxr-xr-x - root supergroup 0 2022-10-11 17:13 /user/hadoop/input
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:48 /user/hadoop/local.txt
-rw-r--r-- 1 root supergroup 0 2022-10-11 17:01 /user/hadoop/text.txt

```

删除目录:

1. 当目录为空时, 直接删除目录

```

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls /user/hadoop/input
Found 0 items

(root@yek)~/Hadoop_work
# java HDFSApi7
2022-10-11 17:20:27,357 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultF
S
2022-10-11 17:20:27,455 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
删除目录: /user/hadoop/input

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls /user/hadoop
Found 2 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:48 /user/hadoop/local.txt
-rw-r--r-- 1 root supergroup 0 2022-10-11 17:01 /user/hadoop/text.txt

```

2. 目录不为空时, 用户选定是否除目录

```

(root@yek)-[~/Hadoop_work]
# ~/.hadoop/bin/hdfs dfs -ls /user/hadoop/input
Found 1 items
-rw-r--r-- 1 root supergroup 0 2022-10-11 17:27 /user/hadoop/input/text.txt

(root@yek)-[~/Hadoop_work]
# java HDFSApi7
2022-10-11 17:27:42,066 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
2022-10-11 17:27:42,165 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
目录不为空，是否要强制删除目录(yes/no)
no
目录不为空，不删除： /user/hadoop/input

(root@yek)-[~/Hadoop_work]
# ~/.hadoop/bin/hdfs dfs -ls /user/hadoop/input
Found 1 items
-rw-r--r-- 1 root supergroup 0 2022-10-11 17:27 /user/hadoop/input/text.txt

(root@yek)-[~/Hadoop_work]
# java HDFSApi7
2022-10-11 17:27:57,621 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
2022-10-11 17:27:57,727 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
目录不为空，是否要强制删除目录(yes/no)
yes
删除目录： /user/hadoop/input

(root@yek)-[~/Hadoop_work]
# ~/.hadoop/bin/hdfs dfs -ls /user/hadoop
Found 2 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:48 /user/hadoop/local.txt
-rw-r--r-- 1 root supergroup 0 2022-10-11 17:01 /user/hadoop/text.txt

```

(8) 向 HDFS 中指定的文件追加内容，由用户指定内容追加到原有文件的开头或结尾；

实验代码:调用了(2)题的脚本从 HDFS 下载文件保证文件名的唯一

```

#!/bin/sh

echo "向开头或结尾追加内容"
echo "1. 开头"
echo "2. 结尾"
read ans
if [ $ans = "1" ]
then
    ./dload_from_HDFS.sh $2
    cat $tmp>>$1
    ./bin/hdfs dfs -copyFromLocal -f $1 $2
elif [ $ans = "2" ]
then
    ./bin/hdfs dfs -appendToFile $1 $2
else
    echo "输入错误"
fi
~
~
~
~
~
"append.sh" 17L, 299C written

```

实验结果:传递了两个参数，第一个是本地的文件名，第二个是 HDFS 的文件名

1. 选择追加到结尾

```
[root@master .hadoop]# ./bin/hdfs dfs -cat text.txt
hi
hi
hi
[root@master .hadoop]# echo "hello">text.txt
[root@master .hadoop]# cat text.txt
hello
[root@master .hadoop]# ./append.sh text.txt text.txt
向开头或结尾追加内容
1. 开头
2. 结尾
2
[root@master .hadoop]# ./bin/hdfs dfs -cat text.txt
hi
hi
hi
hello
```

2. 选择追加到开头

```
[root@master .hadoop]# ./append.sh text.txt text.txt
向开头或结尾追加内容
1. 开头
2. 结尾
1
本地产生了新文件text.txt111111
[root@master .hadoop]# ./bin/hdfs dfs -cat text.txt
hello
hi
hi
hi
hello
```

实验 Java 代码:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.io.*;
import java.util.Scanner;

public class HDFSApi8 {
    /**
     * 判断路径是否存在
     */
    public static boolean test(Configuration conf, String path) throws
    IOException {
        FileSystem fs = FileSystem.get(conf);
        return fs.exists(new Path(path));
    }

    /**
     * 追加文本内容
     */
    public static void appendContentToFile(Configuration conf, String
    content, String remoteFilePath)
        throws IOException {
```

```
        FileSystem fs = FileSystem.get(conf);
        Path remotePath = new Path(remoteFilePath);
        /* 创建一个文件输出流,输出的内容将追加到文件末尾 */
        FSDataOutputStream out = fs.append(remotePath);
        out.write(content.getBytes());
        out.close();
        fs.close();
    }

    /**
     * 追加文件内容
     */
    public static void appendToFile(Configuration conf, String
remoteFilePath) throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path remotePath = new Path(remoteFilePath);
        /* 创建一个文件读入流 */
        FileInputStream in = new FileInputStream(localFilePath);
        /* 创建一个文件输出流,输出的内容将追加到文件末尾 */
        FSDataOutputStream out = fs.append(remotePath);
        /* 读写文件内容 */
        byte[] data = new byte[1024];
        int read = -1;
        while ((read = in.read(data)) > 0) {
            out.write(data, 0, read);
        }
        out.close();
        in.close();
        fs.close();
    }

    /**
     * 移动文件到本地
     * 移动后,删除源文件
     */
    public static void moveToLocalFile(Configuration conf, String
remoteFilePath, String localFilePath)
        throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path remotePath = new Path(remoteFilePath);
        Path localPath = new Path(localFilePath);
        fs.moveToLocalFile(remotePath, localPath);
    }
}
```

```

/**
 * 创建文件
 */
public static void touchz(Configuration conf, String remoteFilePath)
throws IOException {
    FileSystem fs = FileSystem.get(conf);
    Path remotePath = new Path(remoteFilePath);
    FSDataOutputStream outputStream = fs.create(remotePath);
    outputStream.close();
    fs.close();
}

//
/**
 * 主函数
 */
public static void main(String[] args) {
    Configuration conf = new Configuration();
    conf.set("fs.default.name", "hdfs://localhost:9000");
    String remoteFilePath = "/user/hadoop/text.txt";
    String content = "新追加的内容\n";
    Scanner input = new Scanner(System.in);
    String choice = input.next();
    // HDFS 文件
    try {
        /* 判断文件是否存在 */
        if (!HDFSApi8.test(conf, remoteFilePath)) {
            System.out.println("文件不存在: " + remoteFilePath);
        } else {
            if (choice.equals("after")) { // 追加在文件末尾
                content = "after 追加内容\n";
                HDFSApi8.appendContentToFile(conf, content,
remoteFilePath);

                System.out.println("已追加内容到文件末尾" +
remoteFilePath);
            } else if (choice.equals("before")) { // 追加到文件开头
                /* 没有相应的 api 可以直接操作, 因此先把文件移动到本地
*/

                /* 创建一个新的 HDFS, 再按顺序追加内容 */
                String localTmpPath = "/user/hadoop/tmp.txt";
                // 移动到本地
                HDFSApi8.moveToLocalFile(conf, remoteFilePath,
localTmpPath);

                // 创建一个新文件

```

```

HDFSApi8.touchz(conf, remoteFilePath);
// 先写入新内容
content = "before 追加内容\n";
HDFSApi8.appendContentToFile(conf, content,
remoteFilePath);

// 再写入原来内容
HDFSApi8.appendToFile(conf, localTmpPath,
remoteFilePath);

System.out.println("已追加内容到文件开头: " +
remoteFilePath);
    }
    }
} catch (Exception e) {
    e.printStackTrace();
}
}
}
}

```

实验结果:

```

(root@yek)~/Hadoop_work
# java HDFSApi8
2022-10-12 22:39:06,242 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
内容追加到原有文件的开头或结尾(before/after)
after
2022-10-12 22:39:10,123 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform..
sing builtin-java classes where applicable
已追加内容到文件末尾 /user/hadoop/text.txt

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -cat /user/hadoop/text.txt
新追加的内容
新追加的内容
新追加的内容
新追加的内容
after追加内容

(root@yek)~/Hadoop_work
# java HDFSApi8
2022-10-12 22:39:52,913 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
内容追加到原有文件的开头或结尾(before/after)
before
2022-10-12 22:40:01,672 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform..
sing builtin-java classes where applicable
已追加内容到文件开头: /user/hadoop/text.txt

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -cat /user/hadoop/text.txt
before追加内容
新追加的内容
新追加的内容
新追加的内容
新追加的内容
after追加内容

```

(9) 删除 HDFS 中指定的文件;


```
[root@master .hadoop]# ./bin/hdfs dfs -ls .
Found 2 items
-rw-r--r--  1 root supergroup          0 2022-10-02 15:54 local.txt
-rw-r--r--  1 root supergroup        21 2022-10-03 11:43 text.txt
[root@master .hadoop]# ./bin/hdfs dfs -rm text.txt
Deleted text.txt
[root@master .hadoop]# ./bin/hdfs dfs -ls .
Found 1 items
-rw-r--r--  1 root supergroup          0 2022-10-02 15:54 local.txt
```

实验 Java 代码:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.io.*;

public class HDFSApi9 {
    /**
     * 删除文件
     */
    public static boolean rm(Configuration conf, String remoteFilePath)
    throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path remotePath = new Path(remoteFilePath);
        boolean result = fs.delete(remotePath, false);
        fs.close();
        return result;
    }

    /**
     * 主函数
     */
    public static void main(String[] args) {
        Configuration conf = new Configuration();
        conf.set("fs.default.name", "hdfs://localhost:9000");
        String remoteFilePath = "/user/hadoop/text.txt";
        // HDFS 文件
        try {
            if (HDFSApi9.rm(conf, remoteFilePath)) {
                System.out.println("文件删除: " + remoteFilePath);
            } else {
                System.out.println("操作失败(文件不存在或删除失败)");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

实验结果:

```
(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls /user/hadoop
Found 2 items
-rw-r--r--  1 root supergroup      0 2022-10-02 15:48 /user/hadoop/local.txt
-rw-r--r--  1 root supergroup 113 2022-10-12 22:40 /user/hadoop/text.txt

(root@yek)~/Hadoop_work
# java HDFSApi9
2022-10-12 23:36:32,302 INFO Configuration.deprecation: fs.default.name is deprecated. Instead
2022-10-12 23:36:32,397 WARN util.NativeCodeLoader: Unable to load native-hadoop library for
sing builtin-java classes where applicable
文件删除: /user/hadoop/text.txt

(root@yek)~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -ls /user/hadoop
Found 1 items
-rw-r--r--  1 root supergroup      0 2022-10-02 15:48 /user/hadoop/local.txt
```

(10) 在 HDFS 中, 将文件从源路径移动到目的路径。

```
[root@master .hadoop]# ./bin/hdfs dfs -ls .
Found 2 items
-rw-r--r--  1 root supergroup      0 2022-10-02 15:54 local.txt
-rw-r--r--  1 root supergroup      0 2022-10-03 12:55 text.txt
[root@master .hadoop]# ./bin/hdfs dfs -mv text.txt text2.txt
[root@master .hadoop]# ./bin/hdfs dfs -ls .
Found 2 items
-rw-r--r--  1 root supergroup      0 2022-10-02 15:54 local.txt
-rw-r--r--  1 root supergroup      0 2022-10-03 12:55 text2.txt
```

实验 Java 代码:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.io.*;

public class HDFSApi10 {
    /**
     * 移动文件
     */
    public static boolean mv(Configuration conf, String remoteFilePath,
String remoteToFilePath) throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path srcPath = new Path(remoteFilePath);
        Path dstPath = new Path(remoteToFilePath);
        boolean result = fs.rename(srcPath, dstPath);
        fs.close();
        return result;
    }

    /**
     * 主函数
     */
    public static void main(String[] args) {
        Configuration conf = new Configuration();
```

```

        conf.set("fs.default.name", "hdfs://localhost:9000");
        String remoteFilePath = "hdfs:///user/hadoop/text.txt"; // 源文件 HDFS 路径
        String remoteToFilePath = "hdfs:///user/hadoop/new.txt"; // 目的 HDFS 路径
        try {
            if (HDFSApi10.mv(conf, remoteFilePath, remoteToFilePath)) {
                System.out.println(" 将文件 " + remoteFilePath + " 移动到 " + remoteToFilePath);
            } else {
                System.out.println("操作失败(源文件不存在或移动失败)");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

实验结果:

```

(root@yek) [~/Hadoop_work]
# ./hadoop/bin/hdfs dfs -ls /user/hadoop
Found 2 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:48 /user/hadoop/local.txt
-rw-r--r-- 1 root supergroup 0 2022-10-13 12:17 /user/hadoop/text.txt

(root@yek) [~/Hadoop_work]
# java HDFSApi10
2022-10-13 12:17:25,744 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
2022-10-13 12:17:25,850 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform
sing builtin-java classes where applicable
将文件 hdfs:///user/hadoop/text.txt 移动到 hdfs:///user/hadoop/new.txt

(root@yek) [~/Hadoop_work]
# ./hadoop/bin/hdfs dfs -ls /user/hadoop
Found 2 items
-rw-r--r-- 1 root supergroup 0 2022-10-02 15:48 /user/hadoop/local.txt
-rw-r--r-- 1 root supergroup 0 2022-10-13 12:17 /user/hadoop/new.txt

```

(二) 编程实现一个类“MyFSDataInputStream”，该类继承“org.apache.hadoop.fs.FSDataInputStream”，要求如下：实现按行读取 HDFS 中指定文件的方法“readLine()”，如果读到文件末尾，则返回空，否则返回文件一行的文本。

实验 Java 代码:

```

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import java.io.*;

public class MyFSDataInputStream extends FSDataInputStream {
    public MyFSDataInputStream(InputStream in) {
        super(in);
    }
}

```

```

    }

    /**
     * 实现按行读取
     * 每次读入一个字符，遇到"\n"结束，返回一行内容
     */
    public static String readline(BufferedReader br) throws IOException
    {
        char[] data = new char[1024];
        int read = -1;
        int off = 0;
        // 循环执行时，br 每次会从上一次读取结束的位置继续读取
        // 因此该函数里，off 每次都从 0 开始
        while ((read = br.read(data, off, 1)) != -1) {
            if (String.valueOf(data[off]).equals("\n")) {
                off += 1;
                break;
            }
            off += 1;
        }
        if (off > 0) {
            return String.valueOf(data);
        } else {
            return null;
        }
    }

    /**
     * 读取文件内容
     */
    public static void cat(Configuration conf, String remoteFilePath)
    throws IOException {
        FileSystem fs = FileSystem.get(conf);
        Path remotePath = new Path(remoteFilePath);
        FSDataInputStream in = fs.open(remotePath);
        BufferedReader br = new BufferedReader(new
        InputStreamReader(in));
        String line = null;
        while ((line = MyFSDataInputStream.readline(br)) != null) {
            System.out.println(line);
        }
        br.close();
        in.close();
        fs.close();
    }

```

```

    }

    /**
     * 主函数
     */
    public static void main(String[] args) {
        Configuration conf = new Configuration();
        conf.set("fs.default.name", "hdfs://localhost:9000");
        String remoteFilePath = "/user/hadoop/text.txt"; // HDFS 路径
        try {
            MyFSDataInputStream.cat(conf, remoteFilePath);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

实验结果:

```

[root@yek]~/Hadoop_work
# ~/.hadoop/bin/hdfs dfs -cat /user/hadoop/text.txt
hello
hello

[root@yek]~/Hadoop_work
# javac MyFSDataInputStream.java

[root@yek]~/Hadoop_work
# java MyFSDataInputStream
2022-10-13 12:45:41,539 INFO Configuration.deprecation: fs.default.name is deprecated
2022-10-13 12:45:41,633 WARN util.NativeCodeLoader: Unable to load native code library: libhdfs.so: /lib64/libhdfs.so: cannot open shared object file: No such file or directory
hello

hello local.txt README.txt text.txt text.txt_3
     bin      text.txt_0 tmp
     share    text.txt_1

```

(三) 查看 Java 帮助手册或其它资料，用“java.net.URL”和“org.apache.hadoop.fs.FsURLConnectionHandlerFactory”编程完成输出 HDFS 中指定文件的文本到终端中。

实验 Java 代码:

```

import org.apache.hadoop.fs.*;
import org.apache.hadoop.io.IOUtils;
import java.io.*;
import java.net.URL;

public class HDFSApi0 {
    static {

```

```

        URL.setURLStreamHandlerFactory(new FsUrlStreamHandlerFactory());
    }

    /**
     * 主函数
     */
    public static void main(String[] args) throws Exception {
        String remoteFilePath = "hdfs:///user/hadoop/text.txt"; // HDFS
文件
        InputStream in = null;
        try {
            /* 通过 URL 对象打开数据流，从中读取数据 */
            in = new URL(remoteFilePath).openStream();
            IOUtils.copyBytes(in, System.out, 4096, false);
        } finally {
            IOUtils.closeStream(in);
        }
    }
}

```

实验结果:

```

(root@yek) - [~/Hadoop_work]
# ~/.hadoop/bin/hdfs dfs -cat /user/hadoop/text.txt
hello
hello

(root@yek) - [~/Hadoop_work]
# java HDFSApi0
2022-10-13 12:59:35,200 WARN util.NativeCodeLoader: Unable to load native-hadoop
sing builtin-java classes where applicable
hello
hello

```

3 实验总结

练习使用利用 shell 脚本编写 Hdfs 的常用命令，加深了我对 shell 脚本的理解，以及对 Hadoop 操作的熟练度，能更快的使用这些命令。通过 Java Api 对 hdfs 进行编程操作，加深了我对 hdfs 文件系统的理解。