hadoop命令

今天是春节前最后一天上班，先祝大家新年快乐。今天依然坚守一线，闲着没事，整理一下之前学习hadoop代码的一些笔记，就先说说hadoop命令的执行过程吧。

不知道有没有人想过【hadoop fs –ls /】这句我们使用频率极高的命令到底是怎么实现的。下面我们就抽丝剥茧，一步步分析它的执行过程。

首先，我们下载源码后找到下面这个路径，因为这个路径就存着hadoop命令本尊。

hadoop-trunk\hadoop-common-project\hadoop-common\src\main\bin

hadoop文件大家打开就知道其实它是一个脚本，发现hadoop作者不仅java编程了得，脚本的编写也相当厉害，（插句题外话现在我们的开发环境，很大人都鄙视写脚本的。但是我见过的高手脚本都是相当厉害的。），我们分析这个脚本不难发现里面最关键的两句，一个是下面这个函数，它的功能就是执行java命令。

hadoop\_java\_exec

另一个重要的变量HADOOP\_CLASSNAME，告诉我们最终java中使用的具体类是什么。

HADOOP\_CLASSNAME=org.apache.hadoop.fs.FsShell

hadoop\_java\_exec并没有在hadoop文件中，但是细心的玩家不难在如下目录中找到。

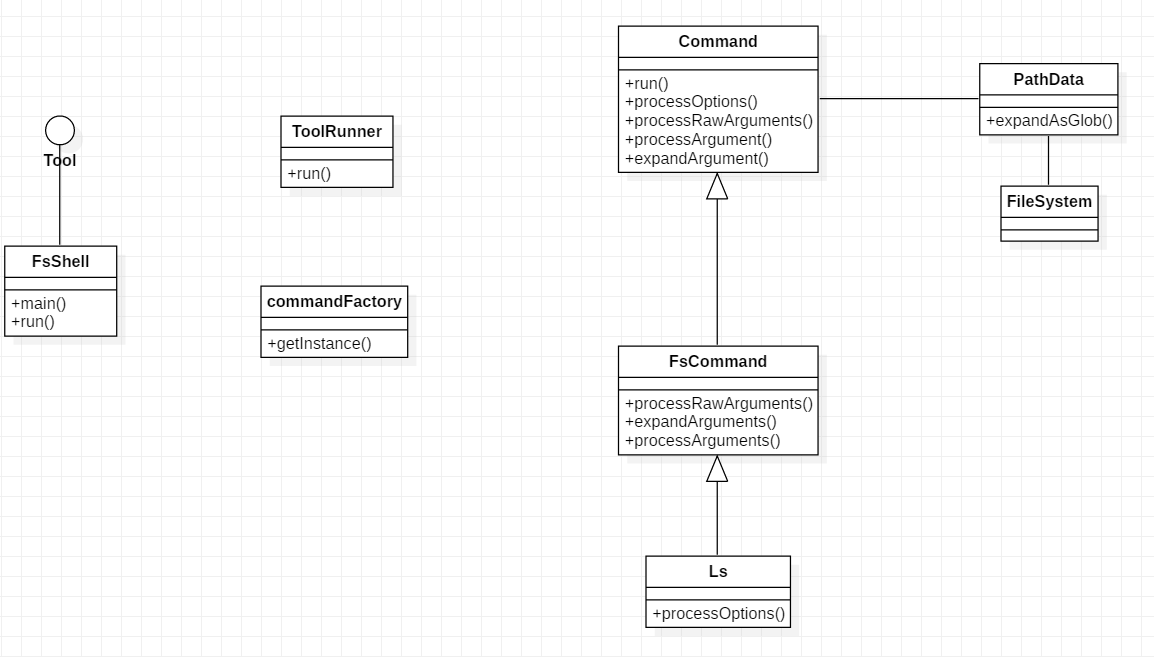
**./hadoop-functions.sh:function**

而这个函数中最重要的就是下面这句，也就解释了hadoop真正的执行方式。

**exec "${JAVA}" "-Dproc\_${command}" ${HADOOP\_OPTS} "${class}" "$@"**

接下来我们就可以进入java代码的学习了。

方便起见我先整理了一下主要用到的几个类图，先简单介绍一下，后面会有一个时序图说明重点方法的调用关系。



现来了解一下FsShell类，这个类有main方法，也就是我们hadoop命令的入口。它通过ToolRunner的run方法，重新生成FsShell对象，并执行run方法。而FsShell的run方法中就引入了commandFactory对象，并找到最终执行ls命令的类Ls.java类。是不是很直接。

简单的都说完了，下面看一下Ls这个类，它的父亲是Fscommand。它的爷爷是Command。这里面其实父亲的事情最少，主要是爷爷和孙子之前玩耍。隔辈亲呀。

Command主要负责调用关系的管理，说明整理命令执行的脉络，而Ls类则负责其中最细节的打印显示信息等具体的实现。Fscommand呢就是打杂的，虽然名字起得很牛x但是功能最弱，从代码行数就可以看出来：

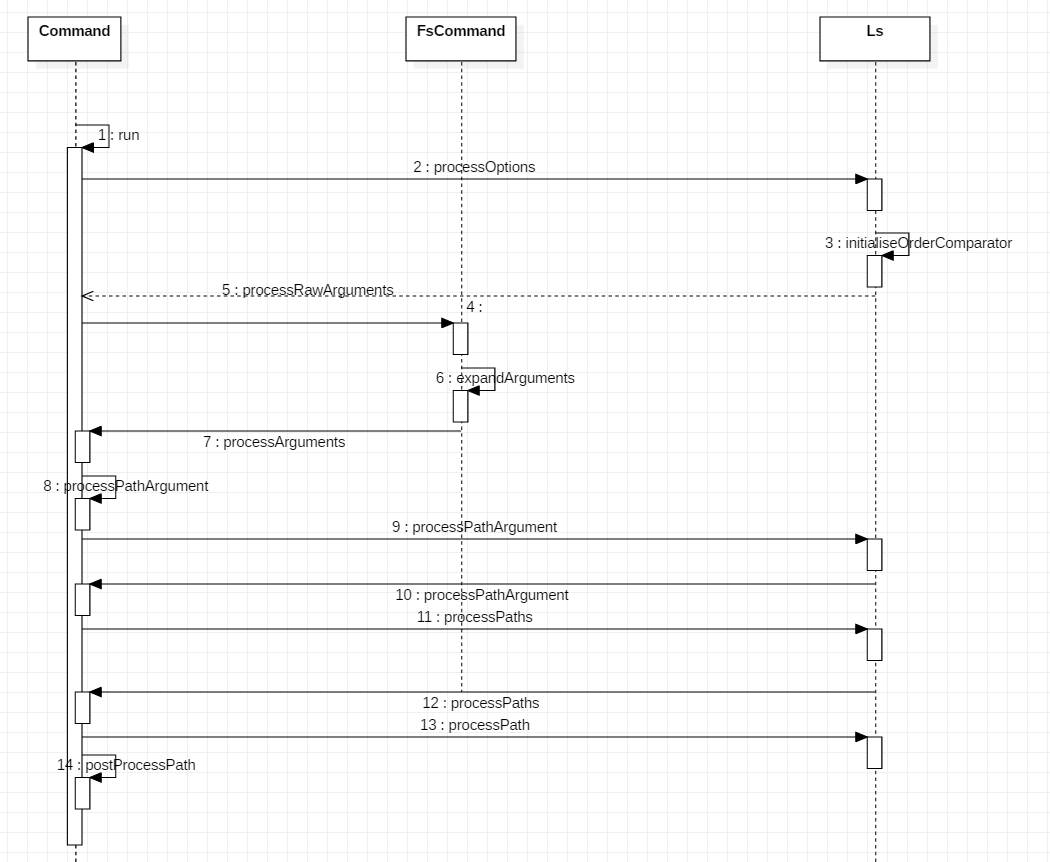
Fscommand:120;

Ls:320;

Command:他喵的500多！！！

感觉就好像现在的中层领导一样，两头受着夹板气。

扯偏了赶紧回来，下面说说具体的调用关系。



其实有了上面的时序图，再对着代码基本就可以看明白了。但是我还是多说几句废话吧，这里面最主要的脉络就是图中2、8、14.包括执行的准备阶段，开使执行和收尾，但是ls这个命令14是空的，所以主要就是2和8了。2也是非核心的东西，所以主要看8就可以了。

这里面Command的expandArgument方法尤为重要，因为这个命令与FileSystem的交互就隐含在这个方法中。而我们真正需要的也就是FileSystem中的信息。这个下次再说吧。今天重点还是在命令解析这块。

行了，通过上面罗里吧嗦的白活，希望能把hadoop的命令解析说清楚，这些对于大佬来讲都是废话，但是希望对小白能有一点帮助，所面还例了一下关键的类和方法关系，感兴趣的朋友可以看看。另外我也是个小白如果有错的地方还请指正。

好了，就写到这里啦，谢谢。如果有人喜欢我后面接着整理。

FsShell

**FsShell shell = newShellInstance();**

**res = ToolRunner.run(shell, argv);**

**ToolRunner**

**GenericOptionsParser parser = new GenericOptionsParser(conf, args);**

**tool.run(toolArgs);**

Tool类

FsShell

**run**

**Command instance = commandFactory.getInstance(cmd);**

**exitCode = instance.run(Arrays.copyOfRange(argv, 1, argv.length));**

**commandFactory**

**registerCommands**

**Ls extends FsCommand ->Command**

**public int run(String...argv) {**

**LinkedList<String> args = new LinkedList<String>(Arrays.asList(argv));**

**try {**

**if (isDeprecated()) {**

**displayWarning(**

**"DEPRECATED: Please use '"+ getReplacementCommand() + "' instead.");**

**}**

**processOptions(args);**

**processRawArguments(args);**

**} catch (CommandInterruptException e) {**

**displayError("Interrupted");**

**return 130;**

**} catch (IOException e) {**

**displayError(e);**

**}**

**return (numErrors == 0) ? exitCode : exitCodeForError();**

**}**

**processOptions**

**processRawArguments**

**processArguments(expandArguments(args));**

**expandArguments**

**processArgument**

**processPathArgument**

**processPaths**

**\* processPath**

**\* postProcessPath**

**processNonexistentPath**

**PathNotFoundException**

**Ls**

**processOptions initialiseOrderComparator**

**processRawArguments**

**expandArguments**

**expandArgument**

**PathData.expandAsGlob**

**processArguments**

**processArguments**

**processArgument**

**processPathArgument**

**processPaths**

**processPath**

**postProcessPath**

**processNonexistentPath**