

需求分析

需求：有大量的文本（文档、网页），需要建立搜索索引

思路分析：

首选将文档的内容全部读取出来，加上文档的名字作为 key，文档的 value 为 1，

组织成这样的一种形式的数据

map 端数据输出

hello-a.txt 1

hello-a.txt 1

hello-a.txt 1

reduce 端数据输出

hello-a.txt 3

代码实现

```
public class IndexCreate extends Configured implements Tool {  
    public static void main(String[] args) throws Exception {  
        ToolRunner.run(new Configuration(), new IndexCreate(), args);  
    }  
    @Override  
    public int run(String[] args) throws Exception {  
        Job job = Job.getInstance(super.getConf(),  
IndexCreate.class.getSimpleName());  
        job.setInputFormatClass(TextInputFormat.class);
```

```

        TextInputFormat.addInputPath(job,new Path("file:///F:\\青软实训大数据离线阶段课程资料\\7、大数据离线第七天\\索引\\inindex"));

        job.setMapperClass(IndexCreateMapper.class);

        job.setMapOutputKeyClass(Text.class);

        job.setMapOutputValueClass(IntWritable.class);

        job.setReducerClass(IndexCreateReducer.class);

        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(IntWritable.class);

        job.setOutputFormatClass(TextOutputFormat.class);

        TextOutputFormat.setOutputPath(job,new Path("file:///F:\\青软实训大数据离线阶段课程资料\\7、大数据离线第七天\\索引\\outindex"));

        boolean bool = job.waitForCompletion(true);

        return bool?0:1;

    }

    public static class IndexCreateMapper extends
Mapper<LongWritable,Text,Text,IntWritable>{

        Text text = new Text();

        IntWritable v = new IntWritable(1);

        @Override

        protected void map(LongWritable key, Text value, Context
context) throws IOException, InterruptedException {

            //获取文件切片

            FileSplit fileSplit = (FileSplit)
context.getInputSplit();

            //通过文件切片获取文件名

            String name = fileSplit.getPath().getName();

            String line = value.toString();

            String[] split = line.split(" ");

            //输出 单词--文件名作为key value 是 1

```

```
        for (String word : split) {  
            ;text.set(word+"---"+name);  
            context.write(text,v);  
        }  
    }  
}  
  
public static class IndexCreateReducer extends  
Reducer<Text,IntWritable,Text,IntWritable>{  
    IntWritable value = new IntWritable();  
    @Override  
    protected void reduce(Text key, Iterable<IntWritable> values,  
Context context) throws IOException, InterruptedException {  
        int count = 0;  
        for (IntWritable value : values) {  
            count += value.get();  
        }  
        value.set(count);  
        context.write(key,value);  
    }  
}  
}
```