Linux程序设计第一次作业

作业1说明

运行作业1的方式为 *./ft.sh*

作业1主要步骤

1. 获得目录参数 *DIR=$1*
2. 处理目录参数，若不是为空，则在最后加一个/（因为后面操作会加上文件名，而空的话代表当前目录，不能加/）。
3. 使用 **ls** 指令打印详细的文件及文件夹信息。
4. 通过管道传入 **grep** ，查找d开头的行数，并传入 **wc** 统计数目，得出文件夹个数。
5. 再使用 **ls** 指令打印文件及文件夹信息。
6. for循环枚举每一个名字，判断是否是可执行文件，统计个数。
7. 使用 **ls -l** 配合 **grep** 获得所有文件名（过滤掉目录）再用 **awk** 去掉详细信息。并定义一个字典数组准备统计。
8. for循环枚举每一个文件名，取得文件后缀（最后一个点之后的字符，对于类似.gitignore的文件，可能会产生歧义）。
9. 判断是否是无后缀文件（去掉后缀后与原文件相同），若无后缀，则累加至无后缀累加器；若有后缀，则在关联数组中将该后缀累加。
10. 将关联数组索引按字典序排序，最后输出各后缀文件的数目。

作业2说明

因为**linux**的思想是一个程序干一件特定的事，所以我将网页获取和之后的处理分成了两个脚本 **crawl.sh** 和 **parse.sh**。

首先使用 ***./crawl.sh*** ，下载的网页将放在**www**文件夹中。

然后使用 ***./deal.sh*** ，分隔出来的单词放在 **words.txt** 中。对**words.txt**统计完，形成一个 **sum[word]** 的数组，再对动词变化和名词变化形式做处理，并且我将转换成原形的日志写到了 **changelist.txt** 中，其中的格式为"单词原形，单词变形"。

最后统计出来的答案存放在 **answer.txt** 中。

作业2主要步骤

1. 使用 **wget** 获取wikipedia中100篇文章的html文件。我发现维基百科有一个链接，随机跳转到维基中一篇文章。并且文章（人工观测下）不会重复，所以我只需要对这个链接wget一百次，分别保存成文件即可。
2. 使用 **grep** 获取带有标签 **<p>** 的那些行，由于维基百科的文章html组织较为有序，故这种方式可行。
3. 通过 **sed** 配合正则表达式去掉html标签。
4. 再使用 **grep** 和正则表达式提取出只由a-z组成的单词，并且通过 **tr** 转成小写。
5. 建立关联数组 **sum[]**，枚举每一个单词，在关联数组中进行累加。

6. 接下来利用语料本身作为词典库，进行变形单词的处理。
7. （**不规则变化**）我从网上搜得不规则动词以及名词的变化表，简单文本处理后存放在 **irregular_nouns.txt**， **irregular_verbs.txt** 中，再运行一个shell脚本 **irreg.sh** ，将其处理成关联数组赋值语句的形式，存放在 **ir_n_map.txt**， **ir_v_map.txt** 中。
8. 在主脚本中，只需定义关联数组 **to[]**，使用 **eval** 命令执行上述两个文件，便生成了一个不规则变化形式到原形的映射表。
9. 对于 **sum[]** 中每个单词，判断其映射后的单词是否存在，存在则将其变化形式的个数累加到原形中。
10. （**规则变化**） 依次判断去掉d，去掉ed，去掉ed以及最后一个双写字母，去掉s，去掉es，去掉es以及最后一个双写字母。看去掉后的单词在 **sum[]** 关联数组中是否存在，存在则将其累加到原形中。
11. （备注）每一次将变化形式累加到原形中，都会记录到 **changelist.txt** 中。
12. 将此时的sum数组按照单词出现的频率 **sort** 排序，并用 **sed** 取出前1000行。