

实验二. Hadoop安装部署

零. 实验前准备

在三台服务器上分别验证

1. Java环境配置验证. 在服务器上测试JDK环境.

```
java -version
```

```
[wyd@wyd-0-202214xxxx ~]$ java -version
java version "1.8.0_341"
Java(TM) SE Runtime Environment (build 1.8.0_341-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.341-b10, mixed mode)
[wyd@wyd-0-202214xxxx ~]$
```

如果测试结果不正确,请参考 [实验1.5 JDK环境配置](#) 重新配置

2. 主机名称验证. 主机名称格式: [姓名缩写-0/1/2-学号](#)

```
hostname
```

```
[wyd@wyd-0-202214xxxx ~]$ hostname
wyd-0-202214xxxx
[wyd@wyd-0-202214xxxx ~]$
```

如果主机名称不符合要求,请参考 [实验1.3 主机名称修改](#) , [1.4 ssh免密配置](#) 重新配置.(最终不符合要求会扣分)

3. ssh免密登陆配置验证. 在服务器上分别执行ssh命令免密登陆三台服务器,例如:

```
ssh wyd-0-202214xxxx

ssh wyd-1-202214xxxx

ssh wyd-2-202214xxxx
```

如果无法满足免密登陆, 请参考 [实验1.4 ssh免密配置](#) 并重新配置

一. 实验介绍

1. 关于本实验

在若干节点上, 安装部署并启动关闭Hadoop分布式集群

2. 实验目的

熟悉Linux命令和CentOS系统。

熟悉在CentOS7.8系统上安装部署Hadoop的流程

熟悉规划Hadoop集群并配置Hadoop相关的配置文件。

熟悉HDFS，YARN集群的自定义配置与启动。

熟悉HDFS，YARN集群的关闭。

3. 实验环境

Linux CentOS 7.8

服务器数目：3

软件包：jdk-8u131-linux-x64.tar.gz + hadoop-2.7.7.tar.gz

二. 实验步骤

1. 安装Hadoop

在0号服务器上，例如 wyd-0-202214xxxx

1. 下载 `hadoop-2.7.7.tar.gz`，并将其通过 `sftp` 或者 `ftp` 或者 `scp` 上传到服务器的 `/opt/software` 目录下。
2. 解压软件包到 `opt/module` 目录下

```
tar -xzvf /opt/software/hadoop-2.7.7.tar.gz -C /opt/module/
```

3. 将Hadoop添加到环境变量(bin与sbin)

1. 打开 `/etc/profile.d/hadoopenv.sh` 文件

```
sudo vim /etc/profile.d/hadoopenv.sh
```

2. 在文件末尾加入以下代码(建议直接复制)

```
#HADOOP_HOME
export HADOOP_HOME=/opt/module/hadoop-2.7.7
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
```

3. 使环境变量生效

```
source /etc/profile
```

4. 测试Hadoop是否安装成功

```
hadoop version
```

```
[wyd@wyd-0-202214xxxx module]$ hadoop version
Hadoop 2.7.7
Subversion Unknown -r c1aad84bd27cd79c3d1a7dd58202a8c3ee1ed3ac
Compiled by stevel on 2018-07-18T22:47Z
Compiled with protoc 2.5.0
From source with checksum 792e15d20b12c74bd6f19a1fb886490
This command was run using /opt/module/hadoop-2.7.7/share/hadoop/common/hadoop-common-2.7.7.jar
```

5. 重启(`hadoop version` 命令不可用时再重启)

```
sync

sudo reboot
```

2. 配置Hadoop集群

在0号服务器上，例如wyd-0-202214xxxx

1. 集群部署规划

	0	1	2
HDFS	<u>NameNode</u>		<u>SecondaryNameNode</u>
	<u>DataNode</u>	<u>DataNode</u>	<u>DataNode</u>
YARN		<u>ResourceManager</u>	
	<u>NodeManager</u>	<u>NodeManager</u>	<u>NodeManager</u>

注:NameNode与SecondaryNameNode不要安装在同一服务器

注:ResourceManager(消耗很多内存)不要和NameNode和SecondaryNameNode安装在同一服务器。

2. 自定义Hadoop配置文件

常见Hadoop端口号在附录

2.1 配置 `core-site.xml` (核心配置文件)

1. 打开 `core-site.xml` 文件

```
vim /opt/module/hadoop-2.7.7/etc/hadoop/core-site.xml
```

2. 修改文件的 `configuration` ,指定NameNode的地址和hadoop数据的存储目录(建议复制后修改)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>

<!--指定NameNode的地址 -->

<!--将hdfs的NameNode节点放在0号服务器的9000端口 -->

    <property>

        <name>fs.defaultFS</name>

        <value>hdfs://wyd-0-202214xxx:9000</value>

    </property>

<!-- 指定hadoop数据的存储目录 -->

    <property>

        <name>hadoop.tmp.dir</name>

        <value>/opt/module/hadoop-2.7.7/data</value>

    </property>

</configuration>

```

2.2 配置 `hdfs-site.xml` (hdfs配置文件)

1. 打开 `hdfs-site.xml` 文件

```
vim /opt/module/hadoop-2.7.7/etc/hadoop/hdfs-site.xml
```

2. 修改文件的 `configuration` ,指定NameNode的WEB UI地址和SecondaryNameNode的WEB UI地址,并设定副本数(建议复制后修改)

```

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>

<!--指定NameNode web端访问地址-->

    <property>

        <name>dfs.namenode.http-address</name>

```

```

        <value>wyd-0-202214xxxx:50070</value>

    </property>

<!--指定Secondary NameNode web端访问地址-->

    <property>

        <name>dfs.namenode.secondary.http-address</name>

        <value>wyd-2-202214xxxx:50090</value>

    </property>

<!--设置副本数为2-->

    <property>

        <name>dfs.replication</name>

        <value>2</value>

    </property>

</configuration>

```

2.3 配置 yarn-site.xml 文件

1. 打开 yarn-site.xml 文件

```
vim /opt/module/hadoop-2.7.7/etc/hadoop/yarn-site.xml
```

2. 修改文件的 configuration ,指定MapReduce走Shuffle,指定ResourceManager的地址,并配置YARN容器的内存管理(建议复制后修改)

```

<?xml version="1.0"?>

<configuration>

<!-- 指定MapReduce走shuffle -->

    <property>

        <name>yarn.nodemanager.aux-services</name>

        <value>mapreduce_shuffle</value>

    </property>

<!-- 指定ResourceManager的地址-->

<!-- 将ResourceManager放在1号服务器上-->

<!-- 其他地方复制,只有这一条需要改-->

    <property>

        <name>yarn.resourcemanager.hostname</name>

        <value>wyd-1-202214xxxx</value>

```

```

    </property>
<!-- 继承环境变量 -->

    <property>

        <name>yarn.nodemanager.env-whitelist</name>

        <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_
        PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>

    </property>
<!-- yarn容器允许分配的最大最小内存 -->

    <property>

        <name>yarn.scheduler.minimum-allocation-mb</name>

        <value>256</value>

    </property>

    <property>

        <name>yarn.scheduler.maximum-allocation-mb</name>

        <value>1024</value>

    </property>
<!-- yarn容器允许管理的物理内存大小 -->

    <property>

        <name>yarn.nodemanager.resource.memory-mb</name>

        <value>1424</value>

    </property>
<!-- 关闭yarn对物理内存和虚拟内存的限制检查 -->

    <property>

        <name>yarn.nodemanager.pmem-check-enabled</name>

        <value>false</value>

    </property>

    <property>

        <name>yarn.nodemanager.vmem-check-enabled</name>

        <value>false</value>

    </property>
</configuration>

```

2.4 配置 `mapred-site.xml` (MapReduce配置文件)

`mapred-site.xml`文件由`mapred-site.xml.template`复制而来

1. 创建并打开 `mapred-site.xml` 文件

```
vim /opt/module/hadoop-2.7.7/etc/hadoop/mapred-site.xml
```

2. 将以下内容复制到 `mapred-site.xml` 文件中,指定MapReduce程序运行在YARN上

```
<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<!-- 指定MapReduce程序运行在YARN上 -->

    <property>

        <name>mapreduce.framework.name</name>

        <value>yarn</value>

    </property>

</configuration>
```

- 2.5 配置 `slaves` (便于群起集群)

1. 打开 `slaves` 文件

```
vim /opt/module/hadoop-2.7.7/etc/hadoop/slaves
```

2. 将自己的服务器 `hostname` 覆盖到 `slaves` 文件中,例如

```
wyd-0-202214xxxx
wyd-1-202214xxxx
wyd-2-202214xxxx
```

- 2.6 将0号服务器的配置同步到1,2 号服务器上

```
rsync -av /opt/module/hadoop-2.7.7 wyd-1-202214xxxx:/opt/module
rsync -av /opt/module/hadoop-2.7.7 wyd-2-202214xxxx:/opt/module
```

- 2.7 在1,2号服务器配置Hadoop环境变量

在1,2号服务器上,例如在wyd-1-202214xxxx和在wyd- 2-202214xxxx服务器上.

1. 分别在1,2号服务器打开 `/etc/profile.d/hadoopenv.sh` 文件

```
sudo vim /etc/profile.d/hadoopenv.sh
```

2. 在文件末尾加入以下代码(建议直接复制)

```
#HADOOP_HOME  
  
export HADOOP_HOME=/opt/module/hadoop-2.7.7  
  
export PATH=$PATH:$HADOOP_HOME/bin  
  
export PATH=$PATH:$HADOOP_HOME/sbin
```

3. 使环境变量生效

```
source /etc/profile
```

4. 验证hadoop安装是否成功

```
hadoop version
```

3. 启动Hadoop集群

1. 格式化Hadoop文件目录

在 0 号服务器上执行, 例如在wyd-0-202214xxxx服务器上

```
hdfs namenode -format
```

2. 运行 `start-dfs.sh` 命令 (打开HDFS)

在 0 号服务器 (NameNode所在节点) 上执行, 例如在wyd-0-202214xxxx服务器上

```
start-dfs.sh
```

```
[wyd@wyd-0-202214xxxx hadoop-2.7.7]$ start-dfs.sh  
Starting namenodes on [wyd-0-202214xxxx]  
wyd-0-202214xxxx: starting namenode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-wyd-nameno  
de-wyd-0-202214xxxx.out  
wyd-1-202214xxxx: starting datanode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-wyd-datano  
de-wyd-1-202214xxxx.out  
wyd-2-202214xxxx: starting datanode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-wyd-datano  
de-wyd-2-202214xxxx.out  
wyd-0-202214xxxx: starting datanode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-wyd-datano  
de-wyd-0-202214xxxx.out  
Starting secondary namenodes [wyd-2-202214xxxx]  
wyd-2-202214xxxx: starting secondarynamenode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-w  
yd-secondarynamenode-wyd-2-202214xxxx.out
```

3. 运行 `start-yarn.sh` 命令 (打开YARN)

在 1号服务器 (ResourceManager所在节点) 上执行, 例如在wyd-1-202214xxxx服务器上


```
start-yarn.sh
```

```
[wyd@wyd-1-202214xxxx hadoop-2.7.7]$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /opt/module/hadoop-2.7.7/logs/yarn-wyd-resourcemanager-wyd-1-202214xxxx.out
wyd-0-202214xxxx: starting nodemanager, logging to /opt/module/hadoop-2.7.7/logs/yarn-wyd-nodemanager-wyd-0-202214xxxx.out
wyd-2-202214xxxx: starting nodemanager, logging to /opt/module/hadoop-2.7.7/logs/yarn-wyd-nodemanager-wyd-2-202214xxxx.out
wyd-1-202214xxxx: starting nodemanager, logging to /opt/module/hadoop-2.7.7/logs/yarn-wyd-nodemanager-wyd-1-202214xxxx.out
```

4. 查看进程是否启动

4.1 在0号服务器,例如wyd-0-202214xxxx服务器上执行 `jps` 命令

```
[wyd@wyd-0-202214xxxx hadoop-2.7.7]$ jps
13415 NodeManager
13514 Jps
13053 NameNode
13183 DataNode
```

4.2 在1号服务器,例如wyd-1-202214xxxx服务器上执行 `jps` 命令

```
[wyd@wyd-1-202214xxxx hadoop-2.7.7]$ jps
11251 NodeManager
11133 ResourceManager
11551 Jps
10991 DataNode
```

4.3 在2号服务器,例如wyd-2-202214xxxx服务器上执行 `jps` 命令

```
[wyd@wyd-2-202214xxxx hadoop-2.7.7]$ jps
10403 DataNode
10580 NodeManager
10502 SecondaryNameNode
10762 Jps
```

5. 对照集群规划查看自己的服务器进程是否与规划一致。

至此, Hadoop的安装部署与启动已经结束。

4. 关闭Hadoop集群

1. 运行 `stop-dfs.sh` 命令 (关闭HDFS)

在0号服务器 (NameNode所在节点) 上执行, 例如在wyd-0-202214xxxx服务器上

```
stop-dfs.sh
```

2. 运行 `stop-yarn.sh` 命令 (关闭YARN)

在1号服务器 (ResourceManager所在节点) 上执行, 例如在wyd-1-202214xxxx服务器上

```
stop-yarn.sh
```

三. 选做

1. 关闭Hadoop集群后, 对HDFS和YARN集群的每个节点的各个服务组件逐一启动, 体会Hadoop集群的搭建过程.

```
#参考指令
```

```
#hdfs
```

```
hadoop-daemon.sh start|stop namenode|datanode|secondarynamenode
```

```
#yarn
```

```
yarn-daemon.sh start|stop resourcemanager|nodemanager
```

2. 在Web UI界面查看Hadoop集群的资源, 查看启动过程中的资源变化. 例如

HDFS

Namenode information

114.116.203.150:50070/dfshealth.html#tab-overview

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

Overview

'wyd-0-202214xxxx:9000' (active)

Started:	Mon Sep 26 16:00:54 CST 2022
Version:	2.7.7, rc1aad84bd27cd79c3d1a7dd58202a8c3ee1ed3ac
Compiled:	2018-07-18T22:47Z by stevel from branch-2.7.7
Cluster ID:	CID-2211ec1e-d665-4c65-9b90-54e17ff045e1
Block Pool ID:	BP-789587212-192.168.0.80-1664118475235

Summary

Security is off.

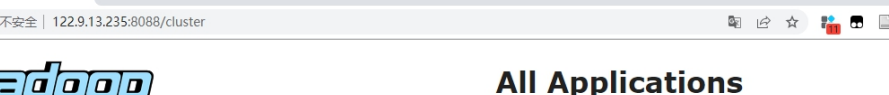
Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 27.5 MB of 60.09 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 41.31 MB of 42.13 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	117.74 GB
DFS Used:	96 KB (0%)
Non DFS Used:	9.74 GB
DFS Remaining:	101.96 GB (86.6%)
Block Pool Used:	96 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	3 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	2022/9/26 16:00:54



The screenshot shows the 'All Applications' page in the Hadoop Distributed File System (HDFS) web interface. The browser address bar shows the URL 'http://122.9.13.235:8088/cluster'. The page features the Hadoop logo and a sidebar with navigation links: Cluster, About Nodes, Node Labels, Applications, NEW, NEW SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, and Scheduler. The main content area displays 'Cluster Metrics' and 'Scheduler Metrics'. The 'Cluster Metrics' table shows various resource usage statistics, including Applications Submitted, Pending, Running, Completed, Containers Running, Memory Used, Memory Total, Memory Reserved, VNodes Used, VNodes Total, VNodes Reserved, Active Nodes, Decommissioned Nodes, and Lost Nodes. The 'Scheduler Metrics' section shows the Scheduler Type as 'Capacity Scheduler' and the Scheduling Resource Type as 'MEMORY'. The Minimum Allocation is set to '<memory:256, vCores:1>' and the Maximum Allocation is set to '<memory:1024, vCores:1>'. The 'Show' dropdown is set to '20' and the 'entries' button is visible. The table below the Scheduler Metrics is empty, displaying 'Showing 0 to 0 of 0 entries'.

All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VNodes Used	VNodes Total	VNodes Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes
0	0	0	0	0	0 B	4.17 GB	0 B	0	24	0	3	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:256, vCores:1>	<memory:1024, vCores:1>

Show: 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
Showing 0 to 0 of 0 entries										

注：在本机访问云服务器的Web端口需先开放端口，可参考华为云的安全组设置。

选做实验提交时,需展示出自己单独启动节点服务组件的过程(不需要每启动一个展示一次,展示出部分过程即可),Web UI界面需展示NameNode, SecondaryNameNode以及ResourceManager的Web UI信息.

四.附录

1. 常见的Hadoop端口地址(Hadoop2.x)

组件	节点	默认端口	配置	用途说明
HDFS	DataNode	50010	dfs.datanode.address	datanode服务端口用于数据传输
HDFS	DataNode	50075	dfs.datanode.http.address	http服务的端口
HDFS	DataNode	50475	dfs.datanode.https.address	https服务的端口
HDFS	DataNode	50020	dfs.datanode.ipc.address	ipc服务的端口
HDFS	NameNode	50070	dfs.namenode.http-address	http服务的端口
HDFS	NameNode	50470	dfs.namenode.https-address	https服务的端口
HDFS	NameNode	8020	fs.defaultFS	接收Client连接的RPC端口，用于获取文件系统metadata信息。
HDFS	journalnode	8485	dfs.journalnode.rpc-address	RPC服务
HDFS	journalnode	8480	dfs.journalnode.http-address	HTTP服务
HDFS	ZKFC	8019	dfs.ha.zkfc.port	ZooKeeper FailoverController，用于NN HA
YARN	ResourceManager	8032	yarn.resourcemanager.address	RM的 applications manager(ASM)端口
YARN	ResourceManager	8030	yarn.resourcemanager.scheduler.address	scheduler组件的IPC端口
YARN	ResourceManager	8031	yarn.resourcemanager.resource-tracker.address	IPC
YARN	ResourceManager	8033	yarn.resourcemanager.admin.address	PC
YARN	ResourceManager	8088	yarn.resourcemanager.webapp.address	http服务端口
YARN	NodeManager	8040	yarn.nodemanager.localizer.address	localizer IPC
YARN	NodeManager	8042	yarn.nodemanager.webapp.address	http服务端口
YARN	NodeManager	8041	yarn.nodemanager.address	NM中 container manager的端口
YARN	JobHistoryServer	10020	mapreduce.jobhistory.address	IPC

2. 常见的Linux命令

2.1 ls命令

ls 命令不仅可以查看 linux 文件夹包含的文件而且可以查看文件权限(包括目录、文件夹、文件权限)查看目录信息等等。

命令格式: ls [选项][目录名]

常用参数

- l : 列出长数据串, 包含文件的属性与权限数据等
- a : 列出全部的文件, 连同隐藏文件 (开头为.的文件) 一起列出来 (常用)
- d : 仅列出目录本身, 而不是列出目录的文件数据
- h : 将文件容量以较易读的方式 (GB, kB等) 列出来
- R : 连同子目录的内容一起列出 (递归列出), 等于该目录下的所有文件都会显示出来

2.2 cd命令

最基本的命令语句, 其他的命令语句要进行操作, 都是建立在使用 cd 命令上的。用于切换当前目录至 dirName。

命令格式: cd [目录名]

2.3 pwd命令

查看"当前工作目录"的完整路径。

命令格式: pwd [选项]

常用参数:

- P :显示实际物理路径, 而非使用连接 (link) 路径
- L :当目录为连接路径时, 显示连接路径

2.4 mkdir命令

用来创建指定的名称的目录, 要求创建目录的用户在当前目录中具有写权限, 并且指定的目录名不能是当前目录中已有的目录。

命令格式: mkdir [选项] 目录

常用参数

- m, --mode=模式, 设定权限<模式> (类似 chmod), 而不是 rwxrwxrwx 减 umask
- p, --parents 可以是一个路径名称。此时若路径中的某些目录尚不存在, 加上此选项后, 系统将自动建立好那些尚不存在的目录, 即一次可以建立多个目录;
- v, --verbose 每次创建新目录都显示信息

--help 显示此帮助信息并退出

--version 输出版本信息并退出

2.5 rm命令

删除一个目录中的一个或多个文件或目录，如果没有使用 `-r` 选项，则 `rm` 不会删除目录。如果使用 `rm` 来删除文件，通常仍可以将该文件恢复原状。

命令格式：rm [选项] 文件

常用参数

-f, --force 忽略不存在的文件，从不给出提示。

-i, --interactive 进行交互式删除

-r, -R, --recursive 指示 `rm` 将参数中列出的全部目录和子目录均递归地删除。

-v, --verbose 详细显示进行的步骤

--help 显示此帮助信息并退出

--version 输出版本信息并退出

2.6 rmdir命令

该命令从一个目录中删除一个或多个子目录项，删除某目录时也必须具有对父目录的写权限。

命令格式：rmdir [选项] 目录

常用参数

-p 递归删除目录 `dirname`，当子目录删除后其父目录为空时，也一同被删除。如果整个路径被删除或者由于某种原因保留部分路径，则系统在标准输出上显示相应的信息。

-v, --verbose 显示指令执行过程

2.7 mv命令

可以用来移动文件或者将文件改名（move (rename) files）。当第二个参数类型是文件时，`mv` 命令完成文件重命名。当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，`mv` 命令将各参数指定的源文件均移至目标目录中。

命令格式：mv [选项] 源文件或目录 目标文件或目录

常用参数

-b : 若需覆盖文件，则覆盖前先行备份

-f : force 强制的意思，如果目标文件已经存在，不会询问而直接覆盖

-i : 若目标文件（destination）已经存在时，就会询问是否覆盖

-u : 若目标文件已经存在，且 source 比较新，才会更新(update)

-t : --target-directory=DIRECTORY move all SOURCE arguments into DIRECTORY，即指定 `mv` 的目标目录，该选项适用于移动多个源文件到一个目录的情况，此时目标目录在前，源文件在后

2.8 cp命令

将源文件复制至目标文件，或将多个源文件复制至目标目录。

命令格式：cp [选项] 源文件 目录 或 cp [选项] -t 目录 源文件

常用参数

-t --target-directory 指定目标目录

-i --interactive 覆盖前询问（使前面的 -n 选项失效）

-n --no-clobber 不要覆盖已存在的文件（使前面的 -i 选项失效）

-f --force 强行复制文件或目录，不论目的文件或目录是否存在

-u --update 使用这项参数之后，只会在源文件的修改时间较目的文件更新时，或是对应的目的文件并不存在，才复制文件

2.9 cat命令

用来显示文件内容，或者将几个文件连接起来显示，或者从标准输入读取内容并显示，它常与重定向符号配合使用。

命令格式：cat [选项] [文件]

常用参数

-A, --show-all 等价于 -vET

-b, --number-nonblank 对非空输出行编号

-e 等价于 -vE

-E, --show-ends 在每行结束处显示 \$

-n, --number 对输出的所有行编号,由1开始对所有输出的行数编号

-s, --squeeze-blank 有连续两行以上的空白行，就代换为一行的空白行

-t 与 -vT 等价

-T, --show-tabs 将跳格字符显示为 ^I

-v, --show-nonprinting 使用 ^ 和 M- 引用，除了 LFD 和 TAB 之外

2.10 more命令

more 命令和 cat 的功能一样都是查看文件里的内容，但有所不同的是more可以按页来查看文件的内容，还支持直接跳转行等功能。

命令格式：more [-dlfpsu] [-num] [+ / pattern] [+ linenum] [file ...]

常用参数

+n 从第n行开始显示

-n 定义屏幕大小为n行

+ / pattern 在每个档案显示前搜寻该字符串（pattern），然后从该字符串前两行之后开始显示

-c 从顶部清屏，然后显示

-d 提示“Press space to continue, 'q' to quit（按空格键继续，按q键退出）”，禁用响铃功能

- l 忽略Ctrl+l (换页) 字符
- p 通过清除窗口而不是滚屏来对文件进行换页, 与-c选项相似
- s 把连续的多个空行显示为一行
- u 把文件内容中的下画线去掉

操作指令

Enter: 向下n行, 需要定义。默认为1行

Ctrl+F: 向下滚动一屏

空格键: 向下滚动一屏

Ctrl+B: 返回上一屏

= : 输出当前行的行号

: f : 输出文件名和当前行的行号

V : 调用vi编辑器

!命令 : 调用Shell, 并执行命令

q : 退出more

2.11 less命令

less 与 more 类似, 但使用 less 可以随意浏览文件, 而 more 仅能向前移动, 却不能向后移动, 而且 less 在查看之前不会加载整个文件。

命令格式: less [参数] 文件

常用参数

- b <缓冲区大小> 设置缓冲区的大小
- e 当文件显示结束后, 自动离开
- f 强迫打开特殊文件, 例如外围设备代号、目录和二进制文件
- g 只标志最后搜索的关键词
- i 忽略搜索时的大小写
- m 显示类似more命令的百分比
- N 显示每行的行号
- o <文件名> 将less 输出的内容在指定文件中保存起来
- Q 不使用警告音
- s 显示连续空行为一行
- S 行过长时间将超出部分舍弃
- x <数字> 将“tab”键显示为规定的数字空格

操作命令

/字符串: 向下搜索“字符串”的功能

?字符串: 向上搜索“字符串”的功能

n: 重复前一个搜索 (与 / 或 ? 有关)

N: 反向重复前一个搜索 (与 / 或 ? 有关)

b 向后翻一页

d 向后翻半页

h 显示帮助界面

Q 退出less 命令

u 向前滚动半页

y 向前滚动一行

空格键 滚动一行

回车键 滚动一页

[pagedown]: 向下翻动一页

[pageup]: 向上翻动一页

2.12 head命令

head 用来显示档案的开头至标准输出中, 默认 head 命令打印其相应文件的开头 10 行。

命令格式: head [参数] [文件]

常用参数

-q 隐藏文件名

-v 显示文件名

-c<字节> 显示字节数

-n<行数> 显示的行数

2.13 tail命令

显示指定文件末尾内容, 不指定文件时, 作为输入信息进行处理。常用查看日志文件。

命令格式: tail [必要参数] [选择参数] [文件]

常用参数

-f 循环读取

-q 不显示处理信息

-v 显示详细的处理信息

-c<数目> 显示的字节数

-n<行数> 显示行数

--pid=PID 与-f合用,表示在进程ID,PID死掉之后结束。

-q, --quiet, --silent 从不输出给出文件名的首部

-s, --sleep-interval=S 与-f合用,表示在每次反复的间隔休眠S秒

2.14 vim命令

vim编辑器是所有Unix及Linux系统下标准的编辑器，它的强大不逊色于任何最新的文本编辑器；vi也是Linux中最基本的文本编辑器，vim就是vi的升级版。

2.14.1 启动vim

在命令行窗口中输入以下命令即可

vim

直接启动vim

vim filename 打开vim并创建名为filename的文件

2.14.2 vim的模式

正常模式（按Esc或Ctrl+[进入） 左下角显示文件名或为空

插入模式（按i键进入） 左下角显示--INSERT--

可视模式（不知道如何进入） 左下角显示--VISUAL--

导航命令

% 括号匹配

2.14.3 插入命令

i 在当前位置前插入

I 在当前行首插入

a 在当前位置后插入

2.14.4 查找命令

/text 查找text，按n键查找下一个，按N键查找前一个。

?text 查找text，反向查找，按n键查找下一个，按N键查找前一个。

:set hlsearch 高亮搜索结果，所有结果都高亮显示，而不是只显示一个匹配。

:set nohlsearch 关闭高亮搜索显示

:nohlsearch 关闭当前的高亮显示，如果再次搜索或者按下n或N键，则会再次高亮。

:set incsearch 逐步搜索模式，对当前键入的字符进行搜索而不必等待键入完成。

:set wrapscan 重新搜索，在搜索到文件头或尾时，返回继续搜索，默认开启。

:set nu 显示行号

2.14.5 撤销和重做

u 撤销 (Undo)

U 撤销对整行的操作

Ctrl + r 重做 (Redo)，即撤销的撤销。

2.14.6 删除命令

x 删除当前字符

3x 删除当前光标开始向后三个字符

X 删除当前字符的前一个字符。X=dh

dl 删除当前字符， dl=x

dh 删除前一个字符

dd 删除当前行

10d 删除当前行开始的10行。

D 删除当前字符至行尾。D=d\$

d\$ 删除当前字符之后的所有字符（本行）

2.14.7 拷贝和黏贴

yy 拷贝当前行

nyy 拷贝当前行开始的n行，比如2yy拷贝当前行及其下一行。

p 在当前光标后粘贴,如果之前使用了yy命令来复制一行，那么就在当前行的下一行粘贴。

shift+p 在当前行前粘贴

2.14.8 退出命令

:wq 保存并退出

ZZ 保存并退出

:q! 强制退出并忽略所有更改

:e! 放弃所有修改，并打开原来文件。

2.14.9 帮助命令

:help or F1 显示整个帮助

:help xxx 显示xxx的帮助，比如 :help i, :help CTRL-[（即Ctrl+[的帮助）。

:help 'number' Vim选项的帮助用单引号括起

:help 特殊键的帮助用<>扩起

:help -t Vim启动参数的帮助用-

: help i_ 插入模式下Esc的帮助，某个模式下的帮助用模式_主题的模式

帮助文件中位于||之间的内容是超链接，可以用Ctrl+]进入链接，Ctrl+o (Ctrl + t) 返回

其他非编辑命令