Linux file permission
Linux file system structure
basic shell scripting
QnA

```
#######################
#!/bin/bash
NAME="John"
echo $NAME
echo "$NAME"
echo "${NAME}!"
##########################
#!/bin/bash
result=$(ls -l)
echo "Result: $result"

########################

#!/bin/bash
# Print a simple message
echo "Hello, World!"
# Print the value of a variable
name="John"
echo "Your name is $name"

#############################

#!/bin/bash
echo "Enter your name: "
read name
echo "Your name is $name"

##############################

#!/bin/bash
car=1
if [ $car -gt 0 ]
then
	echo " the numvber is positive"
else
	echo "the number is not positive"
fi


########################
```

File Conditions

```
[[ -e FILE ]] - Exists
[[ -r FILE ]] - Readable
[[ -h FILE ]] - Symlink
[[ -d FILE ]] - Directory
[[ -w FILE ]] - Writable
[[ -s FILE ]] - Size is > 0 bytes
[[ -f FILE ]] - File
[[ -x FILE ]] - Executable
```

##########################

If statement:

```
if [ -e file.txt ]; then
echo "File exists"
fi
```

###########################

if-else statement:

```
if [ -e file.txt ]; then
echo "File exists"
else
echo "File does not exist"
fi
```

###########################

if-elif-else statement:

```
if [ -f file.txt ]; then
echo "File is a regular file"
elif [ -d file.txt ]; then
echo "File is a directory"
else
echo "File does not "
fi
```

###########################

Basic For Loop

```
#!/bin/bash
# Loop through a list of numbers
for i in 1 2 3 4 5; do
echo $i
done
```

###########################

Basic While Loop

```
#!/bin/bash
# Loop while a condition is true
i=1
while [ $i -le 5 ]; do
echo $i
i=$((i + 1))
done
```

#########################

Back-up Script

```
#!/bin/bash
# Define the source and destination directories
src_dir='/tmp/'
dst_dir='/tmp/var/'
# Copy the source directory to the destination
cp $src_dir $dst_dir
# Confirm that the backup was successful
if [ $? -eq 0 ]; then
echo "Backup successful."
else
echo "Backup failed."
fi

################################
```

System Maintenance

```
#!/bin/bash
# Define the disk usage threshold (in percentage)
threshold=20
# Check the disk usage
disk_usage=$(df -h | awk '{ print $5 }' | grep -v Use | sort -n | tail -1 | cut -d'%' -f1)
# Send an email notification if the disk usage exceeds the threshold
if [ $disk_usage -ge $threshold ]; then
echo "Disk usage is critical ($disk_usage%)." | mail -s "Disk Usage Alert"
user@example.com
fi

##############################

 cat /tmp/test.sh
#!/bin/bash
RED="\033 [1;31m'
CYAN='\033 [1;36m`
NORMAL=`\033 [Om'
GREEN='\033 11:32m`
set =x.
HOST=`uname -n`
#Declaring Falcon process as variables prs_1= pgrep falcond prs
prs_1= `pgrep falcon-sensor`
CONNECTION= `netstat -antlp |grep-w falcon|grep -w ESTABLISHED`
if [[ $prs_1 == "" ||$CONNECTION == "" ]] ; then
# output an error message and go no further
echo -e "$RED Falcon services not loaded completely on SHOST $NORMAL"
exit 30
fi
echo -e "$GREEN Falcon agent installed & services running fine on $HOST SNORMAL"
exit 0

############################
 cat /tmp/test. sh
```

```
#! /bin/bash
hostname-s (uname -n)
size= $(du -sh /datà)
[[ -d /data ]] &&  { echo $size : $hostname;}
exit 0

####################
cat /tmp/test. sh
#! /bin/bash
for user in $ (cat /etc/passwd |cut -d: -f1);
do
echo $user; chage -1 $user' I grep "Password expires";
done | paste -d " " -- | sed 's/Password expires//g' I grep. -v, "never"
```