

Scripting 3: Writing Your Own Scripts

Enhanced addpermplus.sh

1. Include a screenshot of the script running correctly

Your screenshot:

```
jake@itis-3246:~$ ls -la file4
-rw-r--r-- 1 jake jake 0 Feb 13 17:57 file4
jake@itis-3246:~$ ./permplus.sh g a w file4
tests actually work
jake@itis-3246:~$ ls -la file4
-rw-rw-r-- 1 jake jake 0 Feb 13 17:57 file4
jake@itis-3246:~$ ./permplus.sh g r w file4
tests actually work
jake@itis-3246:~$ ls -la file4
-rw-r--r-- 1 jake jake 0 Feb 13 17:57 file4
```

2. Include a screenshot of the script running in an error condition

Your screenshot:

```
jake@itis-3246:~$ ./permplus.sh g b w file4
Second arg must be a|r for add or remove
jake@itis-3246:~$ ./permplus.sh m a w file4
First arg must be one of the following: u|g|o|a
jake@itis-3246:~$
```

3. Include your code for the script. Please ensure that the code you paste here uses a monospaced font and is white.

```
#!/bin/bash
#author Jake Treese <jtreese@uncc.edu>
#this is a script to add or remove permissions to files
#this takes in 4 arguments like this:
#./permplus <arg1> <arg2> <arg3> <filename>

if [[ $# -ne 4 ]]; then
    echo "Need 4 parameters."
    Ex: $0 <u|g|o|a> <a|r> <r|w|x> <filename>"
    exit 1
fi

#test to check if the parameters fall in line
```

```

if [[ $1 != 'u' && $1 != 'g' && $1 != 'o' && $1 != 'a' ]]; then
    echo "First arg must be one of the following: u|g|o|a"
    exit 1
fi

#test to see if second param falls in line
if [[ $2 != 'a' && $2 != 'r' ]]; then
    echo "Second arg must be a|r for add or remove"
    exit 1
fi

#test to see if 3rd param falls in line
if [[ $3 != 'r' && $3 != 'w' && $3 != 'x' ]]; then
    echo "Third arg must be r|w|x for read, write or execute"
    exit 1
fi

if [[ ! -f "$4" ]]; then
    echo "File $4 not found"
    exit 1
fi

echo "tests actually work"

if [[ $2 == 'r' ]]; then
    command="chmod $1-$3 $4"
elif [[ $2 == 'a' ]]; then
    command="chmod $1+$3 $4"
fi

$command

ls -la $4

```

Writing a function

4. Include a screenshot of the script running correctly

Your screenshot:

```
jake@itis-3246:~$ lxc delete badtest
jake@itis-3246:~$ ./ipcont.sh
Testing the function create_container
Creating test
jake@itis-3246:~$ lxc list
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
alive-bluebird	STOPPED			CONTAINER	0
c1	RUNNING	172.16.31.253 (eth0)		CONTAINER	0
test	STOPPED			CONTAINER	0

```
jake@itis-3246:~$
```

5. Include a screenshot of the script running in an error condition

Your screenshot:

```
jake@itis-3246:~$ ./ipcont.sh
Testing the function create_container
Incorrect IP address or formatting
Try again with numbers in this format |000.000.000.000|
jake@itis-3246:~$ vi ipcont.sh
jake@itis-3246:~$
```

6. Include your code for the script. Please ensure that the code you paste here uses a monospaced font and is white.

```
#!/bin/bash
#author jake Treese <jtreese@uncc.edu>
#script to make a container and designate its ip
# takes 2 args: ./ipcont <container name> <container ip>

create_container() {
  if [[ $2 =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]]; then

    lxc init ubuntu:18.04 $1
    lxc network attach lxdbr0 $1 eth0
    lxc config device set $1 eth0 ipv4.address $2
  else
```

```

        echo "Incorrect IP address or formatting"
        echo "Try again with numbers in this format [000.000.000.000]"
    fi
}

echo "Testing the function create_container"
#create_container test adfasasdfasdf

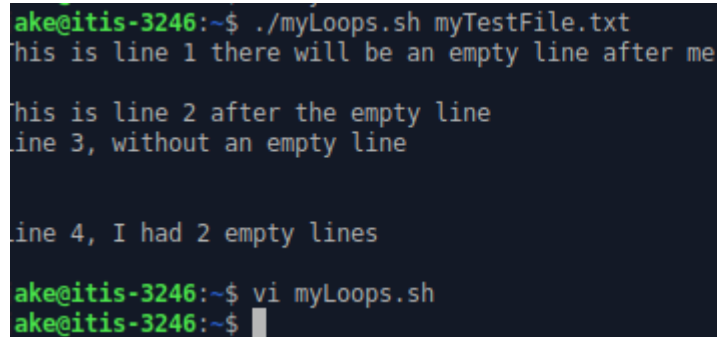
create_container test 172.16.31.254

```

Writing a for loop

7. Include a screenshot of the script running correctly

Your screenshot:



```

ake@itis-3246:~$ ./myLoops.sh myTestFile.txt
this is line 1 there will be an empty line after me

this is line 2 after the empty line
line 3, without an empty line

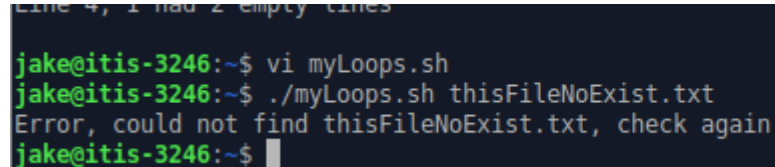
line 4, I had 2 empty lines

ake@itis-3246:~$ vi myLoops.sh
ake@itis-3246:~$

```

8. Include a screenshot of the script running in an error condition

Your screenshot:



```

jake@itis-3246:~$ vi myLoops.sh
jake@itis-3246:~$ ./myLoops.sh thisFileNoExist.txt
Error, could not find thisFileNoExist.txt, check again
jake@itis-3246:~$

```

9. Include your code for the script. Please ensure that the code you paste here uses a monospaced font and is white.

```

#!/bin/bash
#Jake Treese <jtreese@uncc.edu>
#Loop through and read a file
#ignores whitespace and extra lines

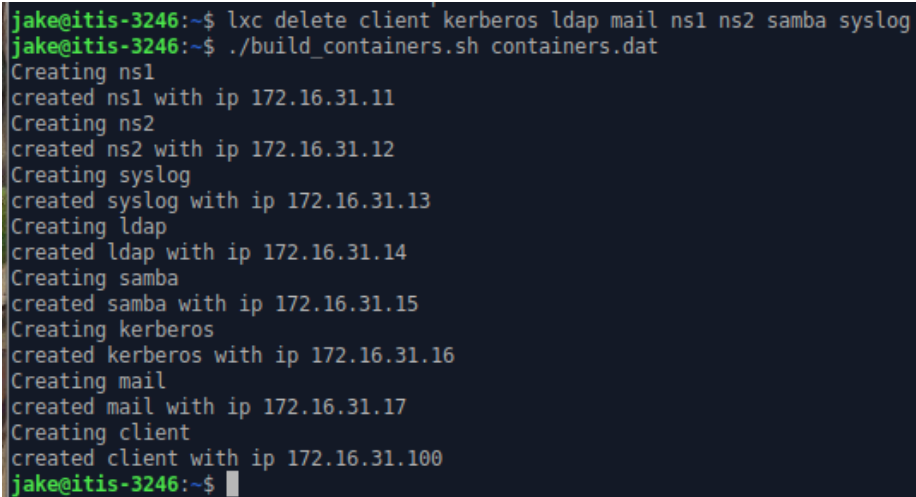
```

```
if [ -f "$1" ]; then
    while IFS= read -r line; do
        echo "$line"
    done < "$1"
else
    echo "Error, could not find $1, check again"
fi
```

Combining functions and for loops

10. Include a screenshot of the script running correctly

Your screenshot:



```
jake@itis-3246:~$ lxc delete client kerberos ldap mail ns1 ns2 samba syslog
jake@itis-3246:~$ ./build_containers.sh containers.dat
Creating ns1
created ns1 with ip 172.16.31.11
Creating ns2
created ns2 with ip 172.16.31.12
Creating syslog
created syslog with ip 172.16.31.13
Creating ldap
created ldap with ip 172.16.31.14
Creating samba
created samba with ip 172.16.31.15
Creating kerberos
created kerberos with ip 172.16.31.16
Creating mail
created mail with ip 172.16.31.17
Creating client
created client with ip 172.16.31.100
jake@itis-3246:~$
```

11. Include a screenshot of the script running in an error condition

Your screenshot:

```

jake@itis-3246:~$ ./build_containers.sh containers.dat
Incorrect IP address or formatting
Creating ns2
created ns2 with ip 172.16.31.12
Creating syslog
created syslog with ip 172.16.31.13
Creating ldap
created ldap with ip 172.16.31.14
Creating samba
created samba with ip 172.16.31.15
Creating kerberos
created kerberos with ip 172.16.31.16
Creating mail
created mail with ip 172.16.31.17
Creating client
created client with ip 172.16.31.100
jake@itis-3246:~$

```

12. Include your code for the script. Please ensure that the code you paste here uses a monospaced font and is white.

```

create_container() {
    if [[ $2 =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]];
    then
        lxc init ubuntu:18.04 $1 </dev/null
        lxc network attach lxdbr0 $1 eth0 </dev/null
        lxc config device set $1 eth0 ipv4.address $2
        echo "created $1 with ip $2"
    else
        echo "Incorrect IP address or formatting"
    fi
}

if [ -f "$1" ]; then
    while IFS= read -r line; do
        create_container $line
    done < "$1"
else
    echo "Error, could not find $1, check again"
    exit 1
fi

```