



# SHELL SCRIPTING

Programs to run in linux environments

# HELLO WORLD

```
#!/bin/bash  
echo "Hello, World!"
```

Same as:

```
#include <stdio.h>  
int main(){  
    printf("Hello, World");  
    Return 0;  
}
```

# VARIABLE

```
#!/bin/bash
```

```
NAME="$1"  
echo "Hello, $1"
```

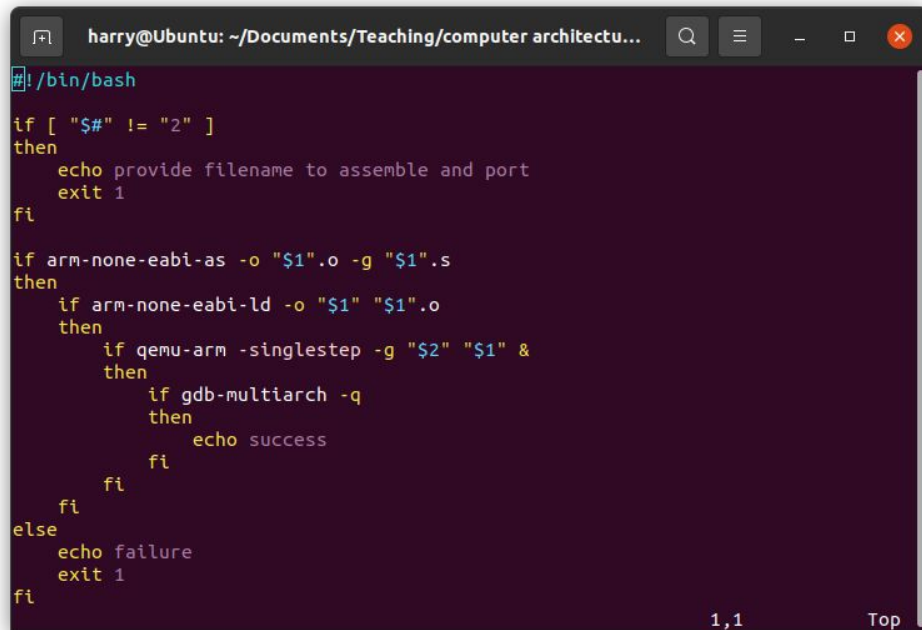
```
-----
```

```
$ ./hello.sh Harry  
Hello, Harry
```

```
$ ./hello.sh Harry Field  
Hello, Harry
```

```
$ ./hello.sh 'Harry Field'  
Hello, Harry Field
```

# REMEMBER THIS?



```
harry@Ubuntu: ~/Documents/Teaching/computer architectu...  
#!/bin/bash  
  
if [ "$#" != "2" ]  
then  
    echo provide filename to assemble and port  
    exit 1  
fi  
  
if arm-none-eabi-as -o "$1".o -g "$1".s  
then  
    if arm-none-eabi-ld -o "$1" "$1".o  
    then  
        if qemu-arm -singlestep -g "$2" "$1" &  
        then  
            if gdb-multiarch -q  
            then  
                echo success  
            fi  
        fi  
    fi  
else  
    echo failure  
    exit 1  
fi
```

1,1 Top

This is terrible, I've made some improvements.

# IMPROVED

```
#!/bin/bash

# e = fail script if any command fails
# u = fail script if reference to any undefined var
set -eu

filename="$1"
port="$2"

arm-none-eabi-as -o "$filename".o -g "$filename".s

arm-none-eabi-ld -o "$filename" "$filename".o

gemu-arm -singlestep -g "$port" "$filename" &

gdb-multiarch -iex "file $filename" -iex "target
remote localhost:$port" -q
```



# TASK

Have a look through the file I have attached to this presentation. This file demonstrates a few different ways you can achieve the same functionality as I have described.

Once you have done this, use your knowledge (and google) to attempt the shell scripting exercise in software tools.