

Forum:[Hack Forums](#)

Thread: [Bash wait for a command to finish then execute another command](#)

friiz1337 - February 2, 2020, 4:48 am

Hello guys.

I have a problem waiting for a specific command within an infinite loop to finish, so another command in another loop to execute. Also the command that I'm waiting to finish is located in a background loop (&). I tried getting the PID of the mv command and then wait for this process to finish but it didn't work to save it as a global variable with process substitution.

Here is an example code of what is happening:

```
Code while true; do
echo "Renaming File"
mv File1 File2
sleep 2
echo "Renaming File again"
mv File2 File1
done &
```

```
while true; do
wait for the mv File2 File1 command to finish
execute another command
done
```

FantaŠec - February 3, 2020, 6:16 pm

The bash script executes sequentially, not in parallel. The functionality you are looking for already exists without special logic you need to create. If you can provide more background on the goal/intent I can help a bit more; however, in what is shown already, bash will execute the Mv command and wait for completion or error before moving on to next line (unless you force bash to not do this using & or other methods).

sed-i - May 14, 2020, 4:09 am

```
Code LOOP1 () {

    while true; do

        echo "Renaming File"
        mv $OLD $NEW

        if [ $? -eq 0 ]; then

            echo "Renaming File again"
            mv $NEW $OLD

            if [ $? -eq 0 ]; then
                echo "execute another command"
            fi
        fi
    done
}
```

```
    else

        echo FAIL

    fi

done

}

LOOP2 () {

    while true; do

        echo "Renaming File"
        mv $OLD $NEW

        if [ $? -eq 0 ]; then

            echo "Renaming File again"
            mv $NEW $OLD

            TMPVAL=$?
            echo "do something else before running the 'execute another command'"

            if [ $TMPVAL -eq 0 ]; then
                echo "execute another command"
            fi

        else

            echo FAIL

        fi

    done

}

old post lol, but prbly very simple.
```