

[639] Linux pipelines

Meenakshi Syamkumar

Learning Objectives

Chain

multiple Linux programs together into a pipeline

Redirect

process output to a file

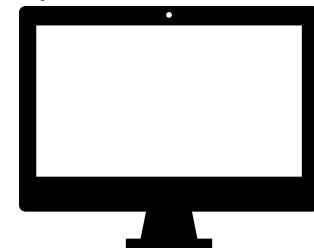
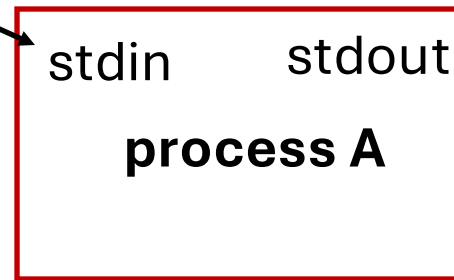
Observe

resource consumption on Linux

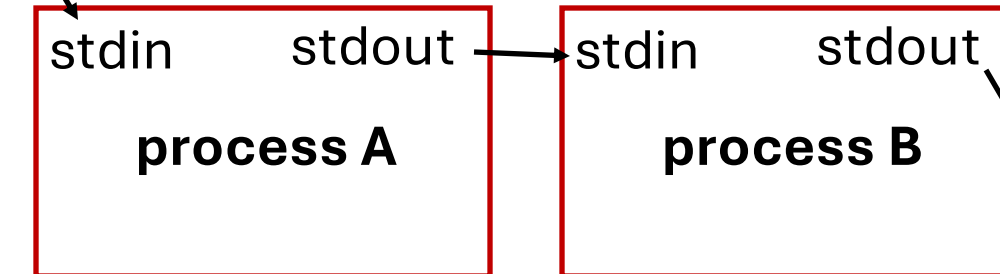
Standard Input and Output (I/O)



Process: instance of a running program



stdout => stdin

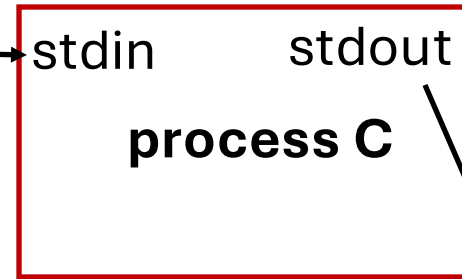
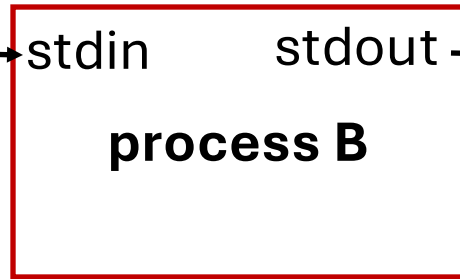
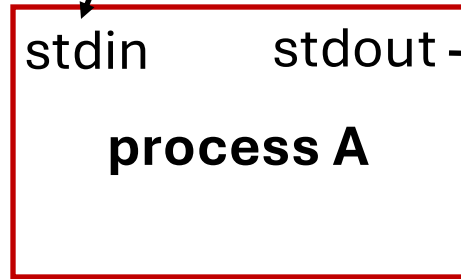


Command:

A | B

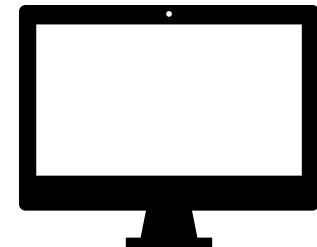


Multiple command chains

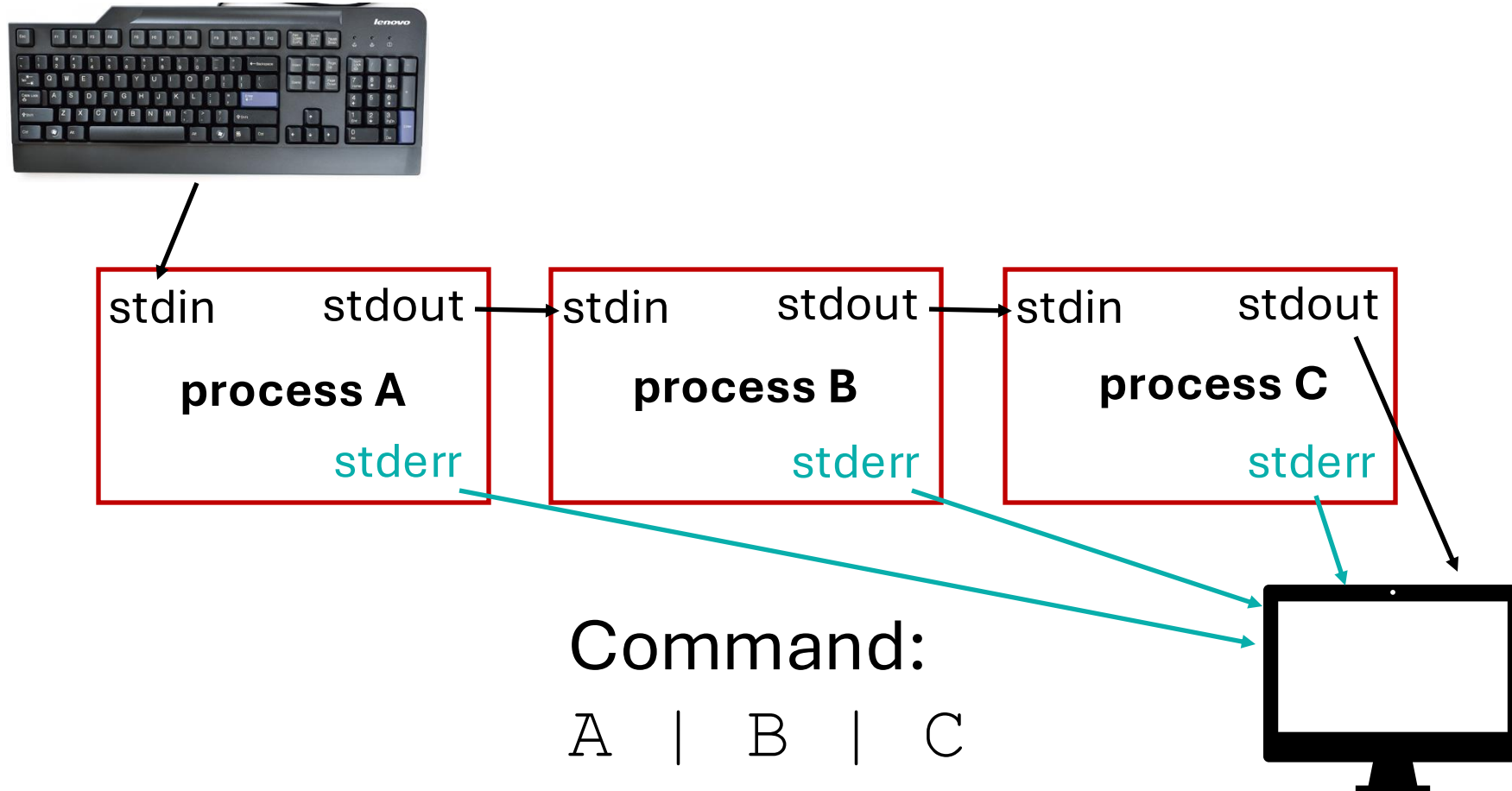


Command:

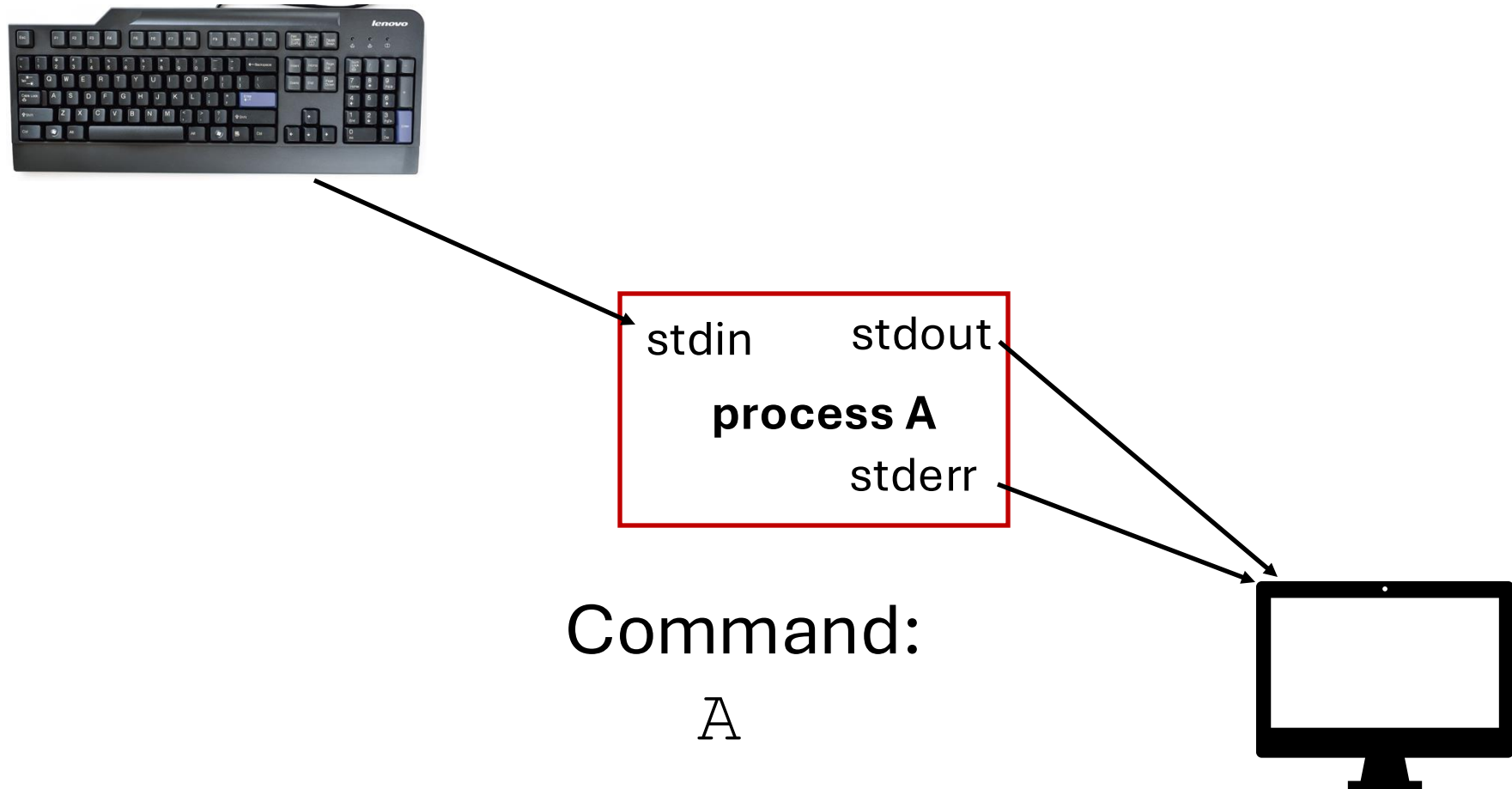
A | B | C



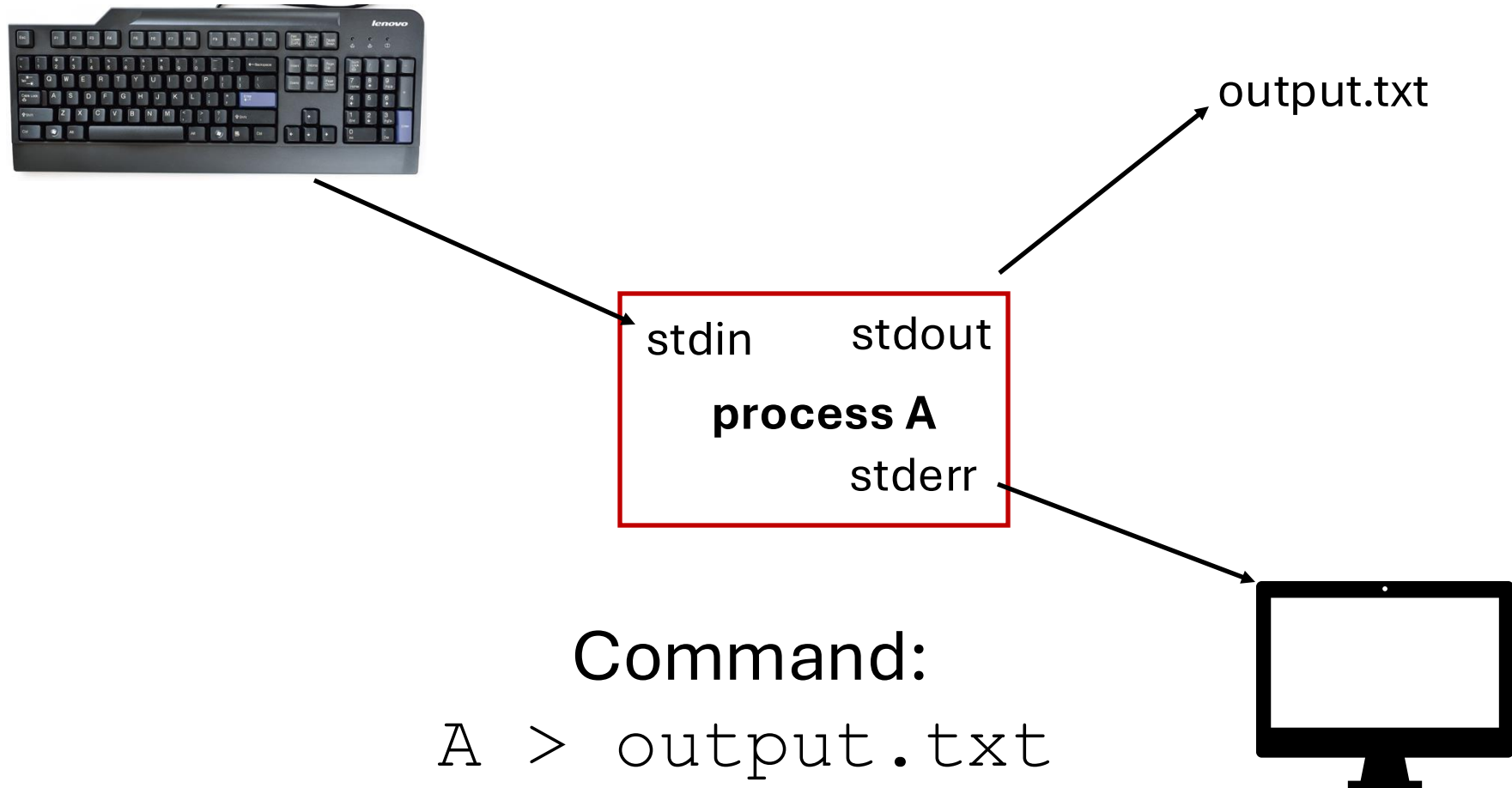
stderr (don't chain)



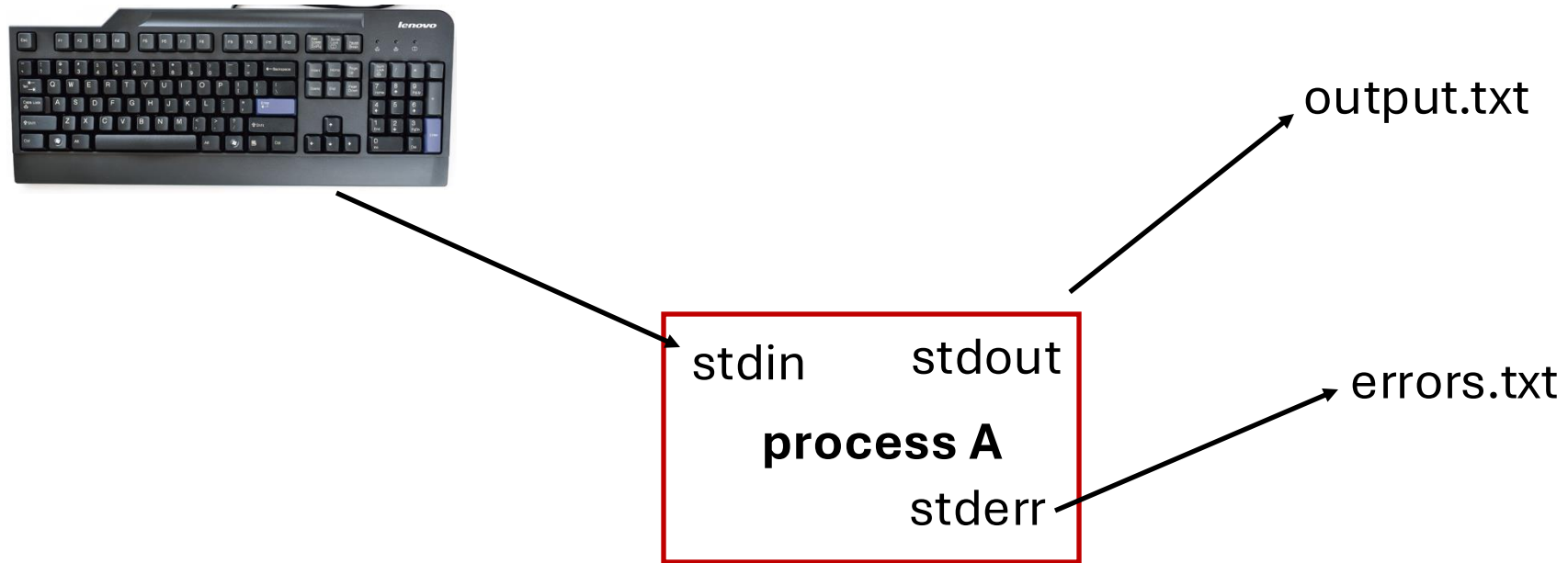
Redirection



Redirection



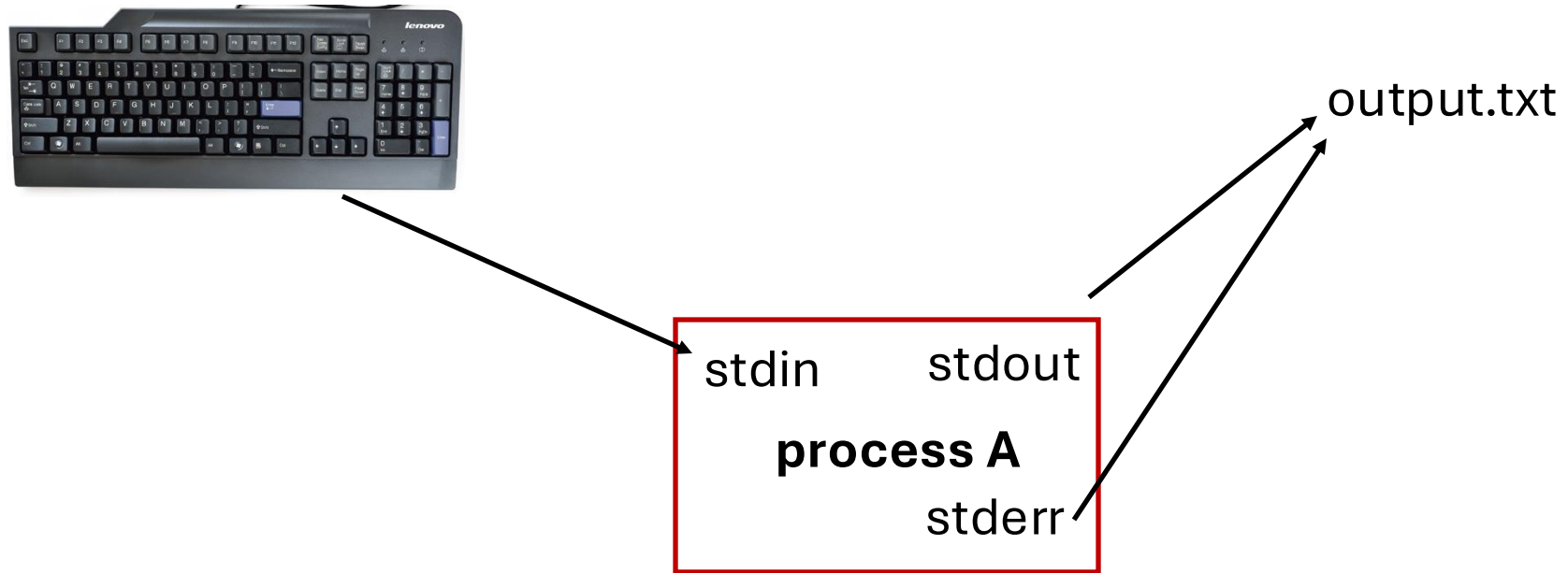
Redirection



Command:

```
A > output.txt 2> errors.txt
```

Redirection

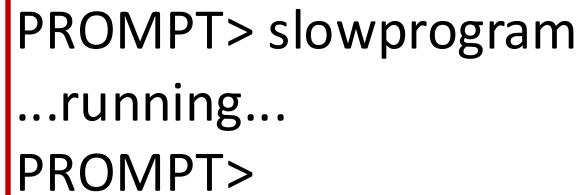


Command:

```
A &> output.txt
```

Async

normally, shells commands are synchronous, meaning you wait for the last command to finish before another prompt appears.



```
PROMPT> slowprogram  
...running...  
PROMPT>
```

A red-bordered box containing terminal output. The first line is 'PROMPT> slowprogram', the second line is '...running...', and the third line is 'PROMPT>'. An arrow points from the text 'normally, shells commands are synchronous...' to the first line of this box.

ampersand at the end runs it in the background and you get a prompt immediately



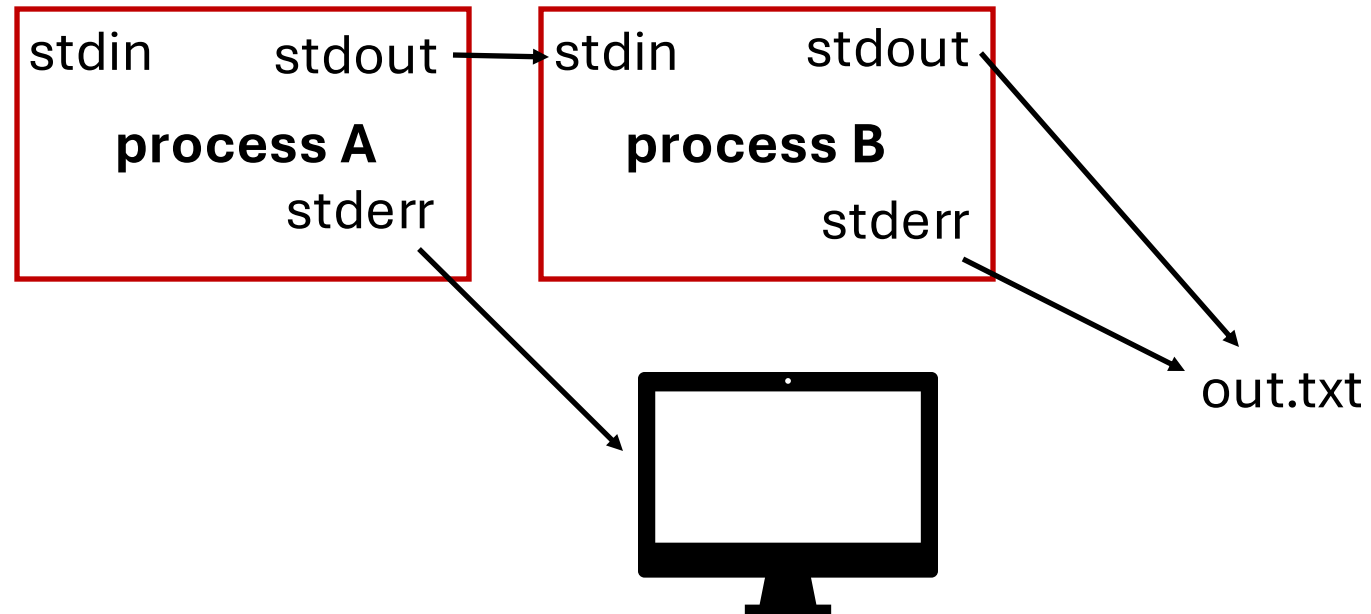
```
PROMPT> slowprogram &  
PROMPT>
```

A red-bordered box containing terminal output. The first line is 'PROMPT> slowprogram &', and the second line is 'PROMPT>'. An arrow points from the text 'ampersand at the end runs it in the background...' to the first line of this box.

Chaining and redirection

Command:

```
A | B &> out.txt &
```



This pipeline will run in the background (perhaps for a long time), and we won't see the output. But we can find it later in the `out.txt` file.