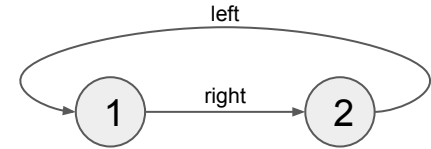
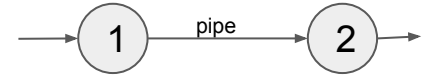


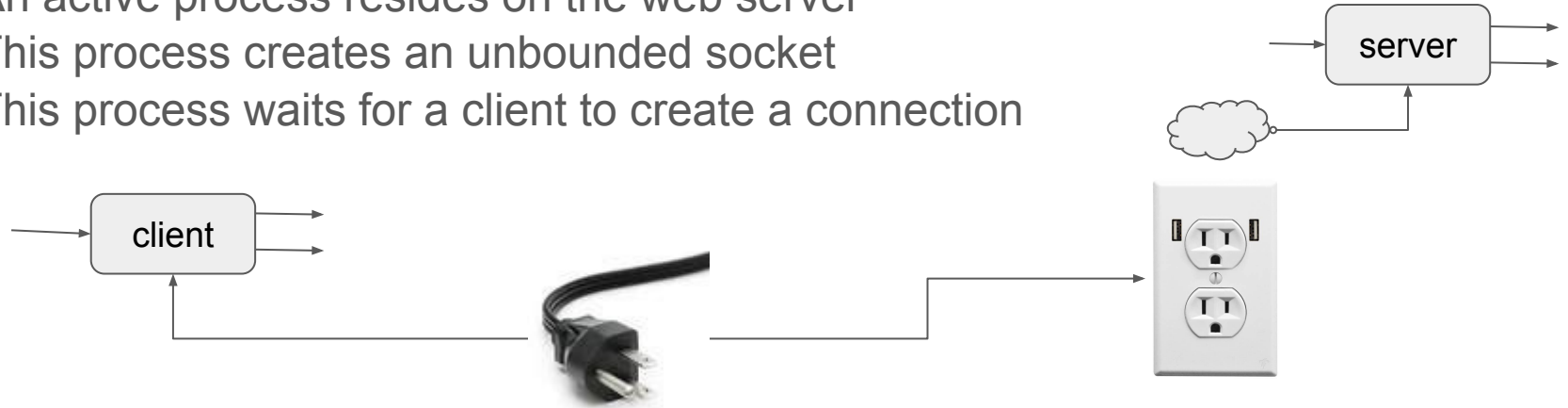
Local Transfer of Information

- This is really just interprocess communication
- One way communication: $\text{prog1} \rightarrow \text{prog2}$
 - mkfifo pipe
 - $\text{prog2} <\text{pipe} \ \& \ \text{prog1} >\text{pipe}$ $\Leftrightarrow \text{prog1} \mid \text{prog2}$
 - rm pipe
- Two way communication: $\text{prog1} \leftarrow \rightarrow \text{prog2}$
 - mkfifo left; mkfifo right;
 - $\text{prog2} <\text{right} >\text{left} \ \&$
 - $\text{prog1} <\text{left} >\text{right}$
 - rm left right
- Issues:
 - This can only be performed within a single server \leftarrow solution: the socket
 - We can't use stdin and stdout for other purposes \leftarrow solution: connect a 3rd file



The Internet is based on Client/Server Model:

- A web server provides content
- An active process resides on the web server
- This process creates an unbounded socket
- This process waits for a client to create a connection



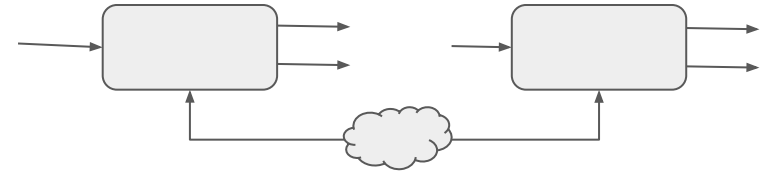
- Once a connection is made, reading/writing to/from the file is performed
 - How? The same way you have been doing it to any other file!



Overview of the Transfer of Information via a Socket

- A socket

- just another type of file
- provides two-way communication
- processes can live on two different machines



- A socket

- A concept at Layer 4 (Transport) of the TCP/IP model
- Establishes a unique connection between two hosts
 - E.g., 130.166.38.161 ← → 130.166.38.209
 - Many possible connections, so add a port number
 - E.g., 130.166.38.161 : 3454 ← → 130.166.38.209 : 443

uniquely names or
identifies a socket

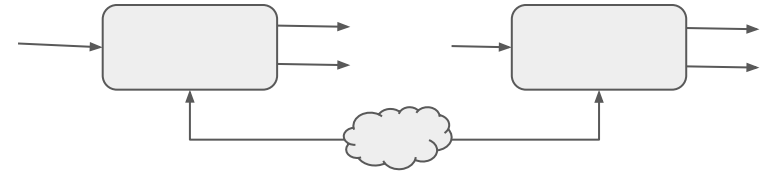
○ → `$ socket -s 2222`

`$ netstat -ln | grep 2222`

Overview of the Transfer of Information via a Socket

- A socket

- just another type of file
- provides two-way communication
- processes can live on two different machines



- A socket

- A concept at Layer 4 (Transport) of the TCP/IP model
- Establishes a unique connection between to hosts
 - E.g., 130.166.38.161 ← → 130.166.38.209
 - Many possible connections, so add a port number
 - E.g., 130.166.38.161 : 3454 ← → 130.166.38.209 : 443

uniquely names or
identifies a socket

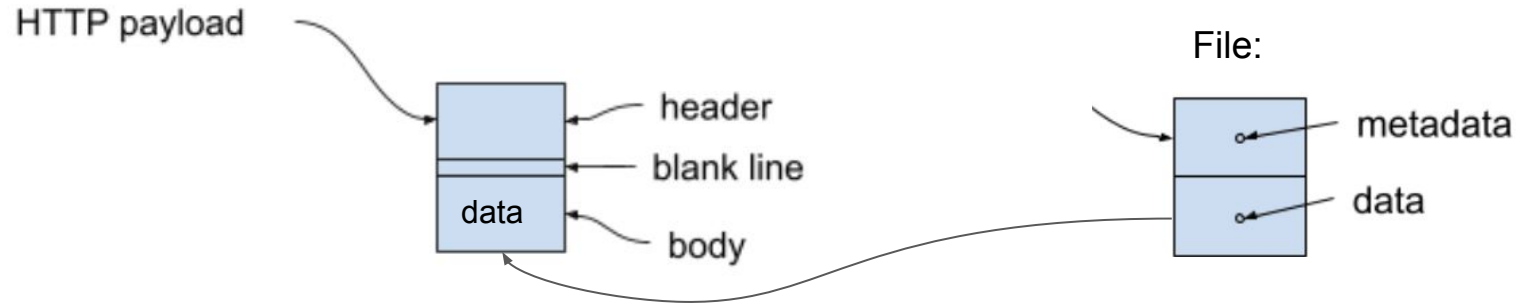
```
$ socket localhost 2222
```

```
$ socket -l 2222
```

```
$ netstat -ln | grep 2222  
$ netstat -n | grep 2222
```

HTTP: HyperText Transfer Protocol: The Payload

- The File's data is placed inside the body of the HTTP payload



- The HTTP payload is transmitted over the socket (the wire)!
- Some of the File's metadata might be placed inside the HTTP header

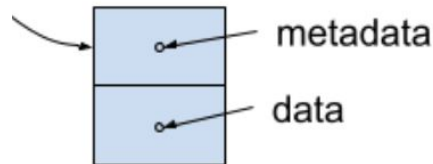
Inside of a file:

```
$ grep bash /usr/share/misc/magic
```

```
0      string/wt      #!\ /bin/bash Bourne-Again shell script text executable
```

- Clues to what is inside:

- The name
- The [file extension](#) <- sometimes advisory!
- Take a peek:
 - `cat filename.txt`
 - `od -c filename.txt`



- The magic:

```
$ cat /usr/share/misc/magic
```

```
$ file file.bash
```

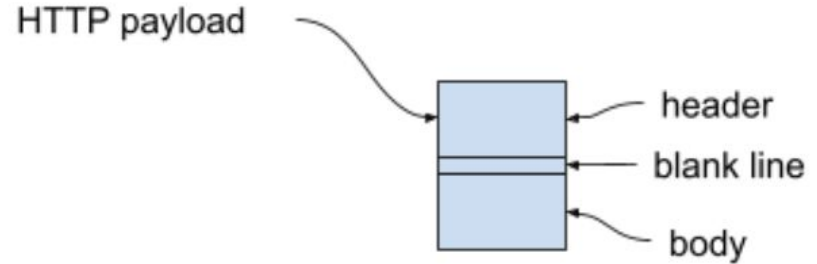
- Advisory?

- Yes, we could ignore this if we want to!

```
$ od # dump files in octal and other formats
```

Inside of a HTTP Payload

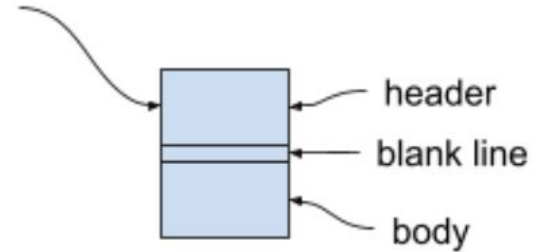
- HTTP Header
 - Request Line:
 - [Set of name/value pairs](#)
- HTTP Body
 - I don't know and I don't care, it's just data
 - But wait, the process cares
- The HTTP Header provides information as to what it is
 - Content-type: text/plain
 - MIME: [Multipurpose Internet Mail Extensions](#)



Inside of a HTTP Payload

- HTTP Header
 - Request / Response Line
 - [Set of name/value pairs](#)
- HTTP Body
 - I don't know and I don't care, it's just data
 - But wait, the process cares
- The HTTP Header provides information as to what it is
 - Content-type: text/plain
 - MIME: [Multipurpose Internet Mail Extensions](#)

HTTP payload



```
$ curl --head www.csun.edu  
$ curl --head https://www.csun.edu
```

```
HTTP/1.0 302 Found  
Location:  
https://www.csun.edu/  
Server: BigIP  
Connection: Keep-Alive  
Content-Length: 0
```

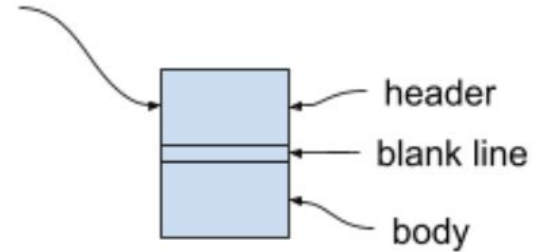

Inside of a HTTP Payload

- HTTP Header
 - Request Line:
 - [Set of name/value pairs](#)
- HTTP Body
 - I don't know and I don't care, it's just data
 - But wait, the process cares
- The HTTP Header provides information as to what it is
 - Content-type: text/plain
 - MIME: [Multipurpose Internet Mail Extensions](#)

\$ curl --head www.csun.edu

\$ curl --head <https://www.csun.edu>

HTTP payload



```
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 24 Mar 2021 17:29:44 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 89644
Content-Language: en
Last-Modified: Wed, 24 Mar 2021 16:51:45 GMT
Expires: Sun, 19 Nov 1978 05:00:00 GMT
```

Writing a CGI Script

- Dynamic Exercise:
 - Create the file: `ssh.sandbox.csun.edu:~steve/public_html/cgi-bin/blah.cgi`
 - Curl the file: `curl --head https://www.sandbox.csun.edu/~steve/cgi-bin/blah.cgi`
 - Open the file: open <https://www.sandbox.csun.edu/~steve/cgi-bin/blah.cgi>

- Output of the script must conform to:
- The request line is provided by the web server
 - content-type: text/html
 - content-type: text/plain
 - content-type: application/pdf

