Project 1
Pipe Hints

Andy Wang

Operating Systems

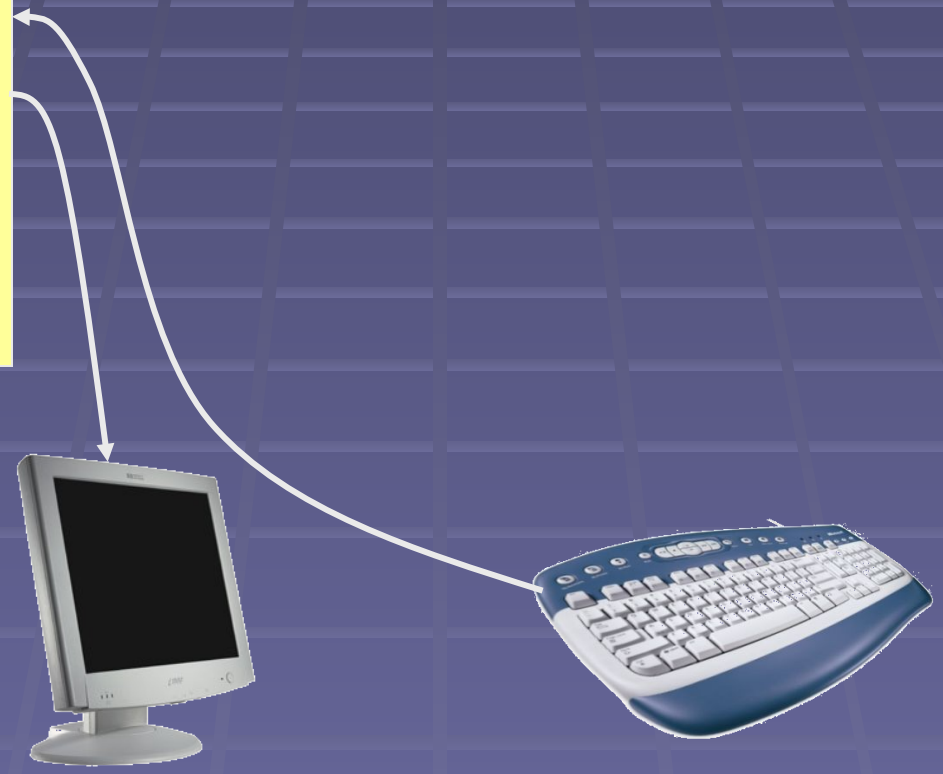
COP 4610 / CGS 5765

Pipe

```
int p1_to_p2[2];
pipe(p1_to_p2);
```

file descriptors
0 stdin
1 stdout
2 stderr

shell

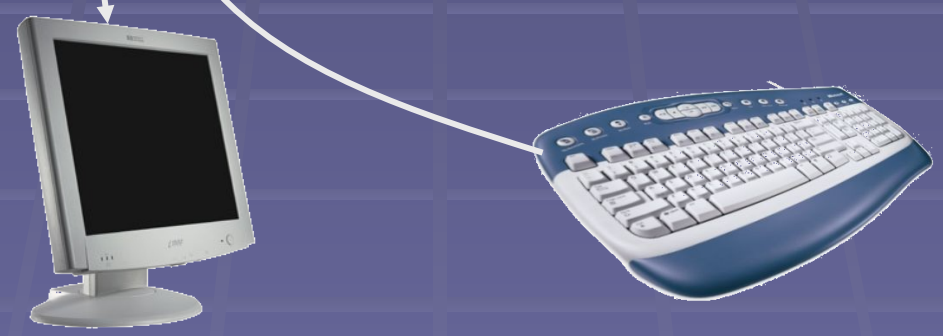


Pipe

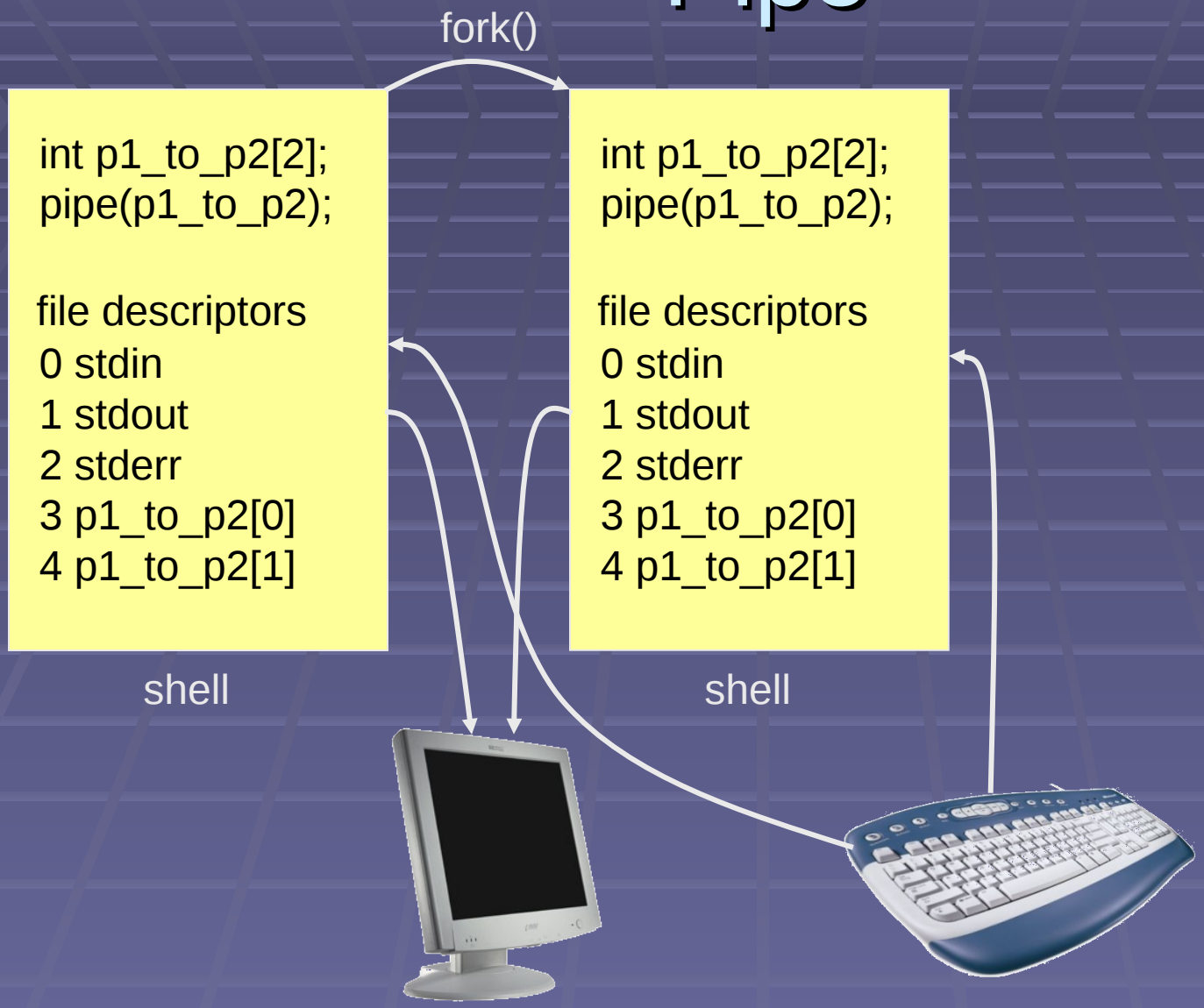
```
int p1_to_p2[2];
pipe(p1_to_p2);
```

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

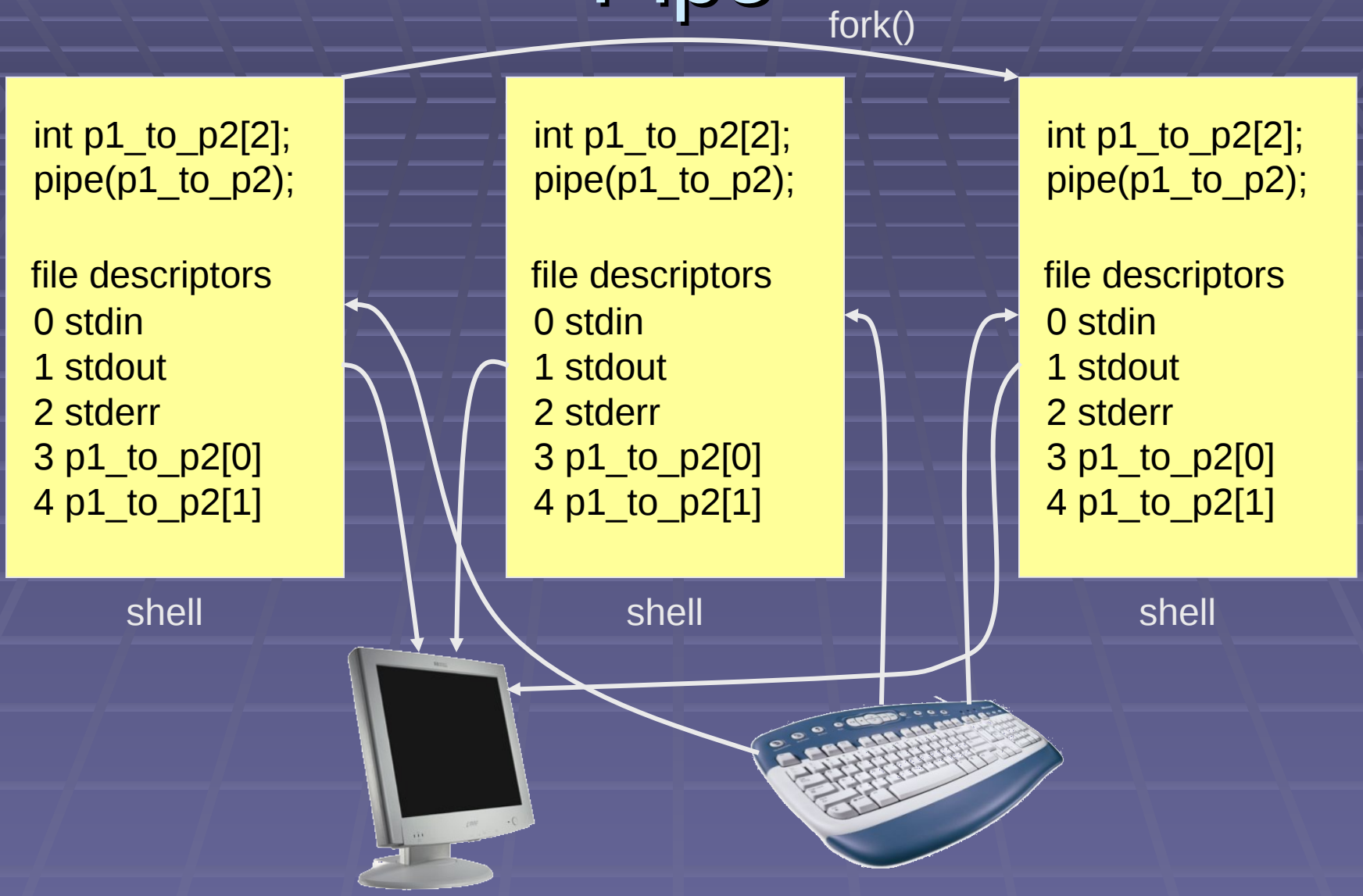
shell



Pipe


fork()
int p1_to_p2[2];
pipe(p1_to_p2);
file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]
shell
int p1_to_p2[2];
pipe(p1_to_p2);
file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]
shell

Pipe


fork()
int p1_to_p2[2];
pipe(p1_to_p2);
file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]
shell
int p1_to_p2[2];
pipe(p1_to_p2);
file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]
shell
int p1_to_p2[2];
pipe(p1_to_p2);
file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]
shell

Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

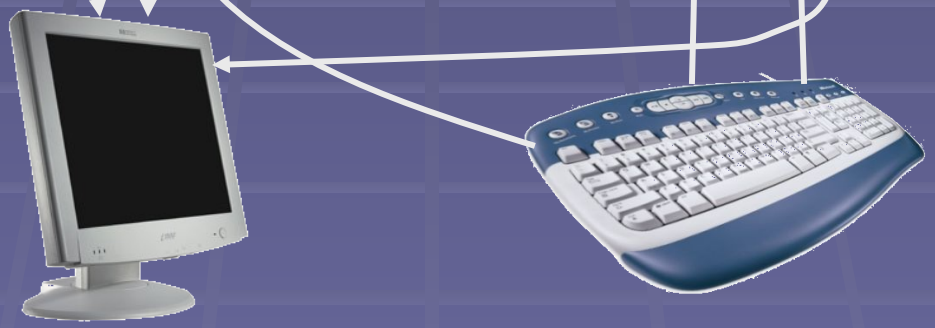
file descriptors
0 stdin
~~1 stdout~~
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell



Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

dup(4)

int p1_to_p2[2];
pipe(p1_to_p2);

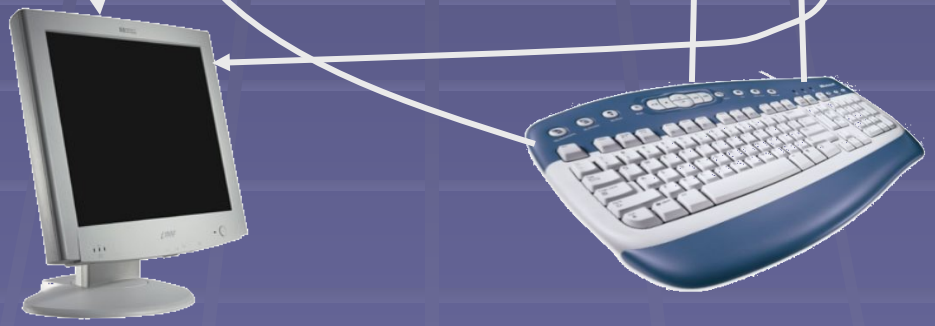
file descriptors
0 stdin
1
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell



Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

dup(4)

int p1_to_p2[2];
pipe(p1_to_p2);

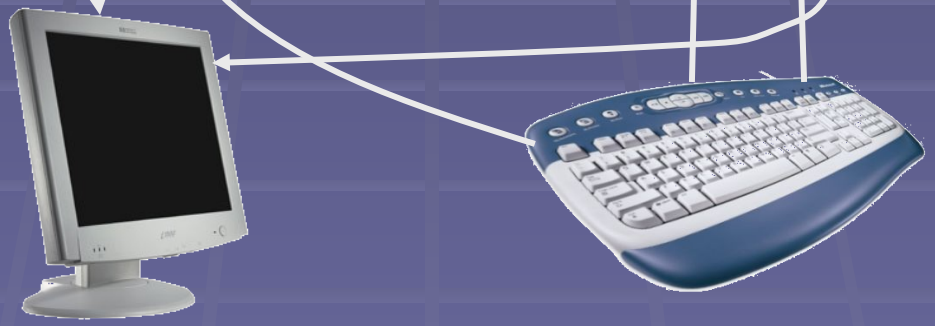
file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell



Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

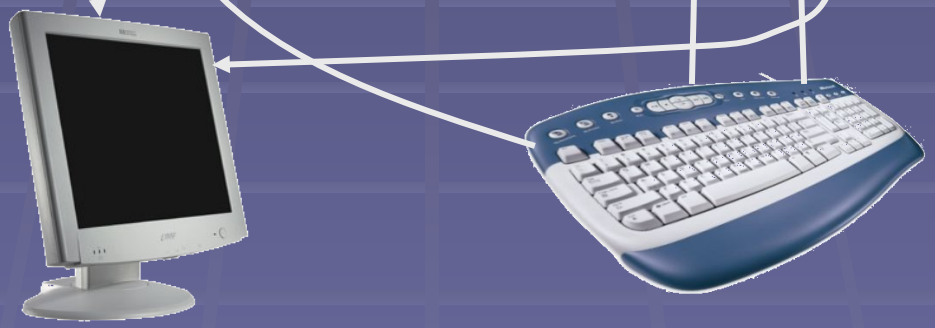
file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr
~~3 p1_to_p2[0]~~
~~4 p1_to_p2[1]~~

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell



Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

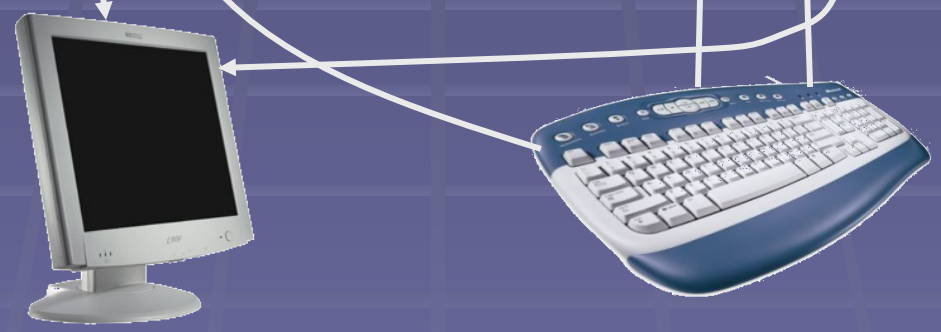
file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell



Pipe

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]
```

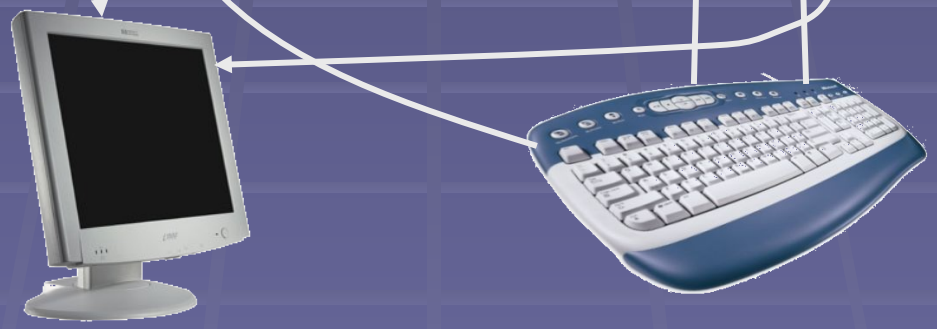
shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

execvp(bin_path,
  argv)
```

shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]
```

shell



Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

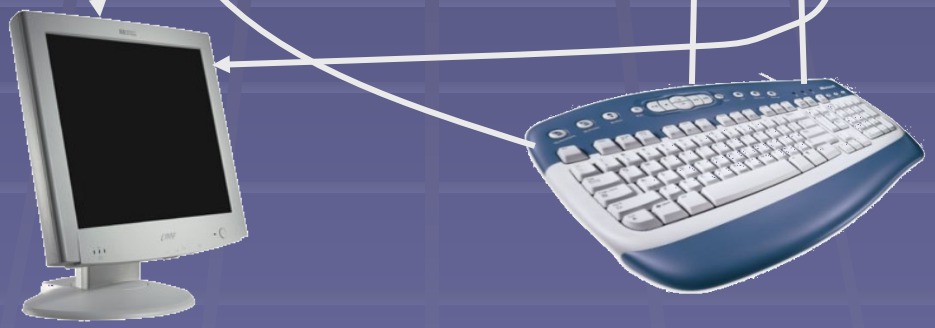
execvp(bin_path,
argv)

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 ~~stdin~~
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell



Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

execvp(bin_path,
argv)

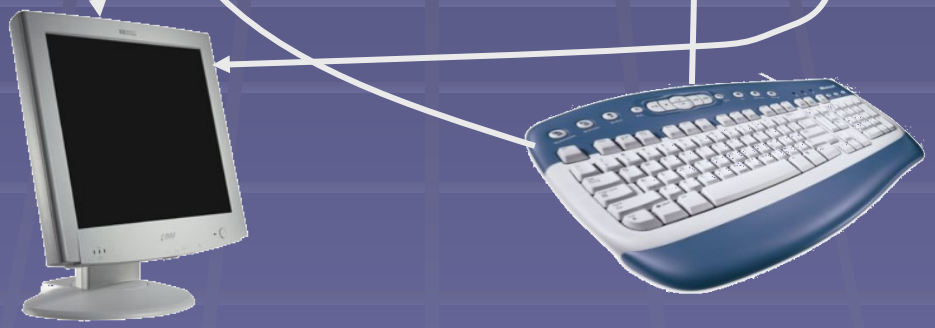
shell

dup(3)

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell



Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

execvp(bin_path,
argv)

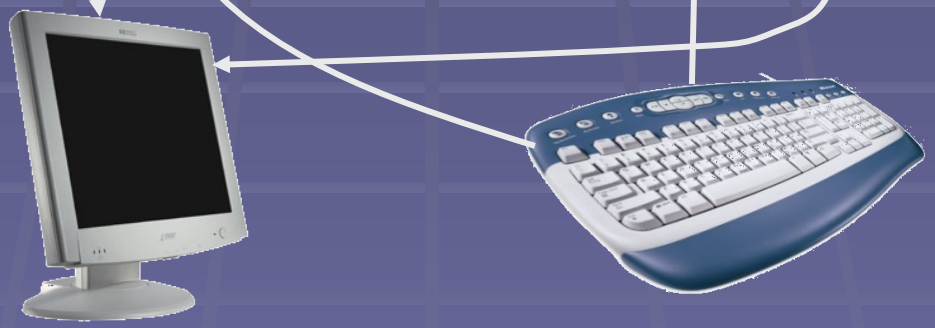
shell

dup(3)

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 p1_to_p2[0]
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell



Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

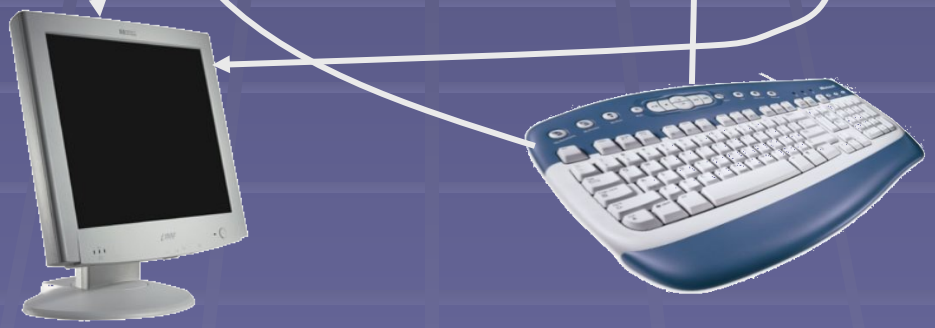
execvp(bin_path,
argv)

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 p1_to_p2[0]
1 stdout
2 stderr
3 ~~p1_to_p2[0]~~
4 ~~p1_to_p2[1]~~

shell



Pipe

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]
```

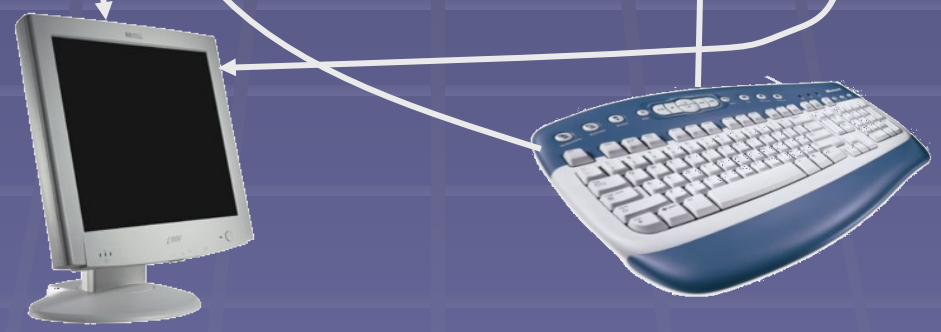
shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

execvp(bin_path,
  argv)
```

shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 p1_to_p2[0]
1 stdout
2 stderr
```

shell



Pipe

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
3 p1_to_p2[0]
4 p1_to_p2[1]
```

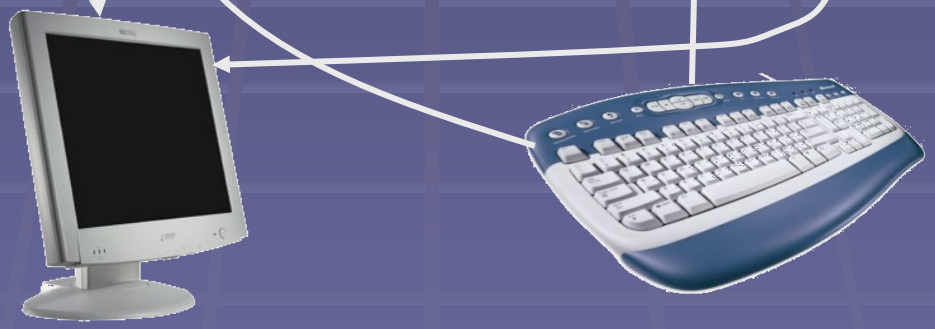
shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

execvp(bin_path,
  argv)
```

shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 p1_to_p2[0]
1 stdout
2 stderr

execvp(bin_path,
  argv)
```

shell



Pipe

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
~~3 p1_to_p2[0]~~
~~4 p1_to_p2[1]~~

shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

execvp(bin_path,
argv)

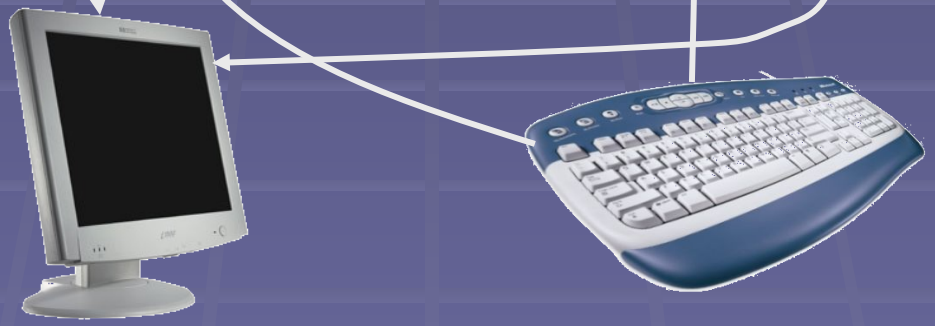
shell

int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 p1_to_p2[0]
1 stdout
2 stderr

execvp(bin_path,
argv)

shell



Pipe

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr
```

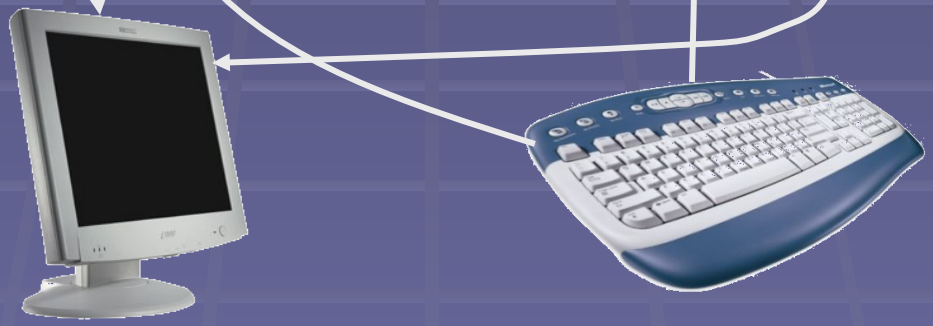
shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

execvp(bin_path,
  argv)
```

shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 p1_to_p2[0]
1 stdout
2 stderr

execvp(bin_path,
  argv)
```

shell



Pipe

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 stdout
2 stderr

waitpid(…)
waitpid(…)
```

shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 stdin
1 p1_to_p2[1]
2 stderr

execvp(bin_path,
  argv)
```

shell

```
int p1_to_p2[2];
pipe(p1_to_p2);

file descriptors
0 p1_to_p2[0]
1 stdout
2 stderr

execvp(bin_path,
  argv)
```

shell