



COMP2432

Tutorial 12 – Past Paper

Past Paper

6. Programming with Pipes (15 points).

Assuming that other statements are correctly written (e.g., definition of variables), *how many processes* are created altogether? *How many possible* outputs can there be when the following program fragment is executed? *List* out those *possible* outputs.

```
pipe(fd1); pipe(fd2); p = fork();  
if (p > 0) strcpy(buf, "hello"); else strcpy(buf, "bye");  
if (p > 0) write(fd1[1], buf, strlen(buf)); else write(fd2[1], buf, strlen(buf));  
if (p > 0) { n = read(fd2[0], buf2, 80); buf2[n] = 0; }  
if (p > 0) p = fork();  
if (p > 0) strcpy(buf, "welcome");  
if (p > 0) write(fd2[1], buf, strlen(buf)); else write(fd1[1], buf, strlen(buf));  
if (p > 0) { n = read(fd1[0], buf, 80); buf[n] = 0; printf("%s\n", buf); }
```

	Parent	Child_1	Child_2
1	<code>pipe(fd1); pipe(fd2); p=fork();</code>		
2	<code>if(p>0) strcpy(buf,"hello");</code>	<code>else strcpy(buf,"bye");</code>	
3	<code>if(p>0) write(fd1[1],buf,strlen(buf));</code>	<code>else write(fd2[1],buf,strlen(buf));</code>	
4	<code>if(p>0) { n=read(fd2[0],buf2,80); buf2[n]=0; }</code>		
5	<code>if(p>0) p=fork();</code>		
6	<code>if(p>0) strcpy(buf,"welcome");</code>		
7	<code>if(p>0) write(fd2[1],buf,strlen(buf));</code>	<code>else write(fd1[1],buf,strlen(buf));</code>	<code>else write(fd1[1],buf,strlen(buf));</code>
8	<code>if(p>0) { n=read(fd1[0],buf,80); buf[n]=0; printf("%s\n",buf); }</code>		



Past Paper

► Possible outputs:

1. hello
2. hellohello
3. hellobye
4. hellohellobye
5. hellobyehello

Past Paper

6. Programming with Pipes (15 points).

There are two lines printed when the following program fragment is executed, assuming that other statements are correctly written (e.g., definition of variables). *How many possible* outputs can there be? *List* out those *possible* outputs. *Give* also one *impossible* output.

```
pipe(fd1); pipe(fd2); p = fork();
if (p > 0) strcpy(buf,"hello"); else strcpy(buf,"bye");
if (p > 0) write(fd1[1],buf,strlen(buf)); else write(fd2[1],buf,strlen(buf));
if (p > 0) strcpy(buf,"welcome"); else strcpy(buf,"farewell");
if (p > 0) write(fd2[1],buf,strlen(buf)); else write(fd1[1],buf,strlen(buf));
if (p > 0) { n = read(fd1[0],buf,80); buf[n] = 0; n = read(fd2[0],buf2,80); buf2[n] = 0;
printf("%s\n %s\n",buf,buf2); }
```

	Parent	Child
1	<code>pipe(fd1); pipe(fd2); p=fork();</code>	
2	<code>if(p>0) strcpy(buf,"hello");</code>	<code>else strcpy(buf,"bye");</code>
3	<code>if(p>0) write(fd1[1],buf,strlen(buf));</code>	<code>else write(fd2[1],buf,strlen(buf));</code>
4	<code>if(p>0) strcpy(buf,"welcome");</code>	<code>else strcpy(buf,"farewell");</code>
5	<code>if(p>0) write(fd2[1],buf,strlen(buf));</code>	<code>else write(fd1[1],buf,strlen(buf));</code>
6	<code>if(p>0) { n=read(fd1[0],buf,80);buf[n]=0;</code>	
7	<code>n=read(fd2[0],buf2,80);buf2[n]=0;</code>	
8	<code>printf("%s\n %s\n",buf,buf2); }</code>	



Past Paper

There are six *possible outputs*:

1. **hello** then next line **welcome**
2. **hello** then next line **byewelcome**
3. **hello** then next line **welcomebye**
4. **hellofarewell** then next line **welcomebye**
5. **hellofarewell** then next line **byewelcome**
6. **farewellhello** then next line **byewelcome**

It is *impossible* to generate the following three viable outputs:

7. **hellofarewell** then next line **welcome**
8. **farewellhello** then next line **welcome**
9. **farewellhello** then next line **welcomebye**