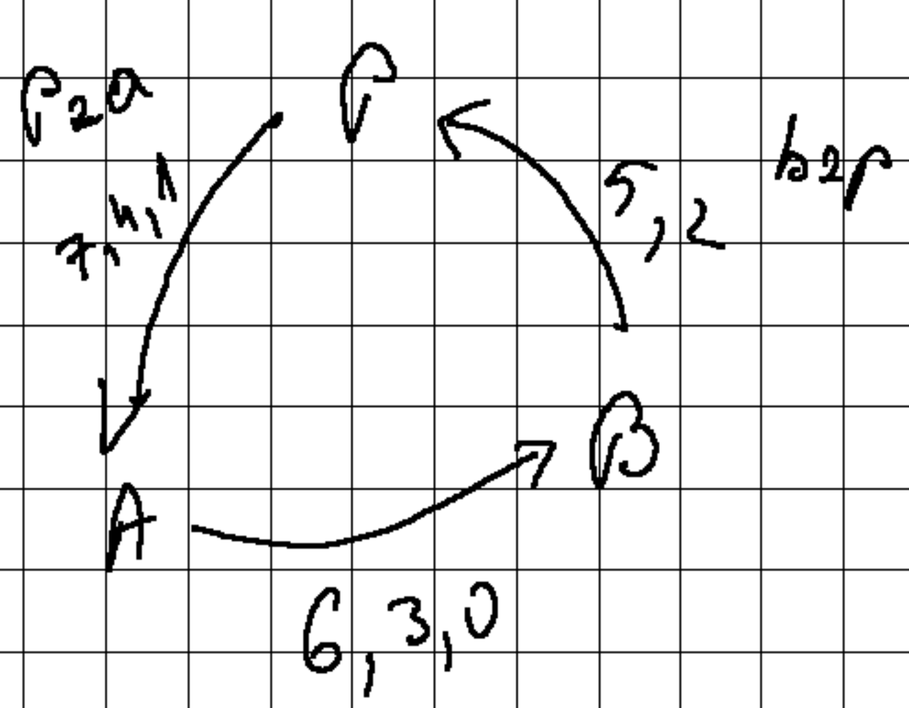
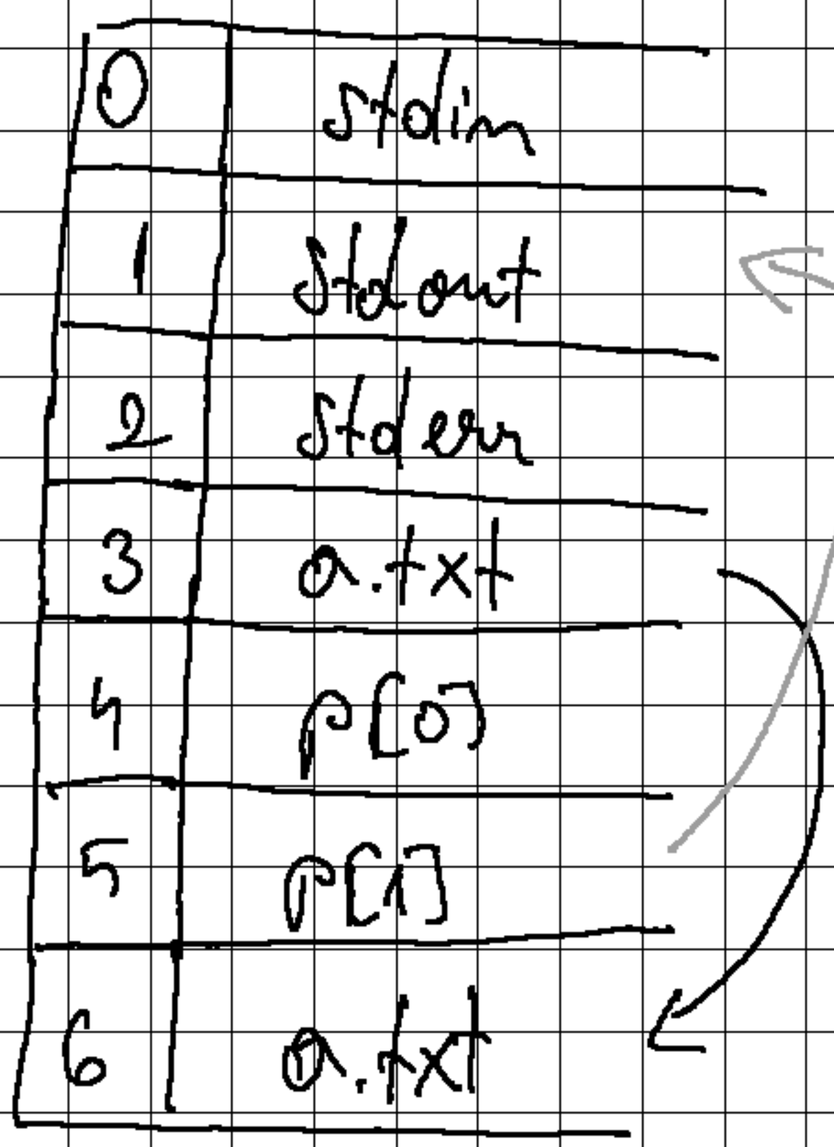



```
int main(...) {
    int p2a[2], a2b[2], b2p[2], n;
    pipe(p2a);
    pipe(a2b);
    pipe(b2p);
    if( fork() == 0){    // code specific for A
        close(p2a[1], a2b[0], b2p[0], b2p[1]);
        while(1){
            if( read(p2a[0], &n, sizeof(int)) <0){
                break;
            }
            if (n<=0){
                break;
            }
            n--;
            write( a2b[1], &n, sizeof(int));
        }
        close(p2a[0], p2b[1];
        exit(0)
    }
```

```
if (fork() == 0) {      // code specific for B
    close( p2a[0] , p2a[1], a2b[1] , b2p[0] );
    while(1) {
        if (read( a2b[0] , &n, sizeof(int) < 0) {
            break;
        }
        if(n <= 0) {
            break;
        }
        n--;
        write( b2p[1], &n, sizeof(int));
    }
    close( a2b[0], b2p[1]);
    exit(0);
}
```

```
close(p2a[0], a2b[0], a2b[1], b2p[1]);
n=7
write(p2a[1], &n, sizeof(int));
while(1){
    if(read(b2p[0], &n, sizeof(int)) < 0){
        break;
    }
    if(n<=0){
        break;
    }
    n--;
    write(p2a[1], &n, sizeof(int));
}
close(b2p[0], p2a[1]);
wait(0);
wait(0);   } 2 child processes => 2 waits
return 0;
}
```

FIFO: Lecture notes (lecture 6)



dup(3) // copies the org from 3

dup2(int old, int new)

dup2(5,1)