

# HW6 SAQ

## SAQ1

```
#include <stdio.h>
#include <unistd.h>

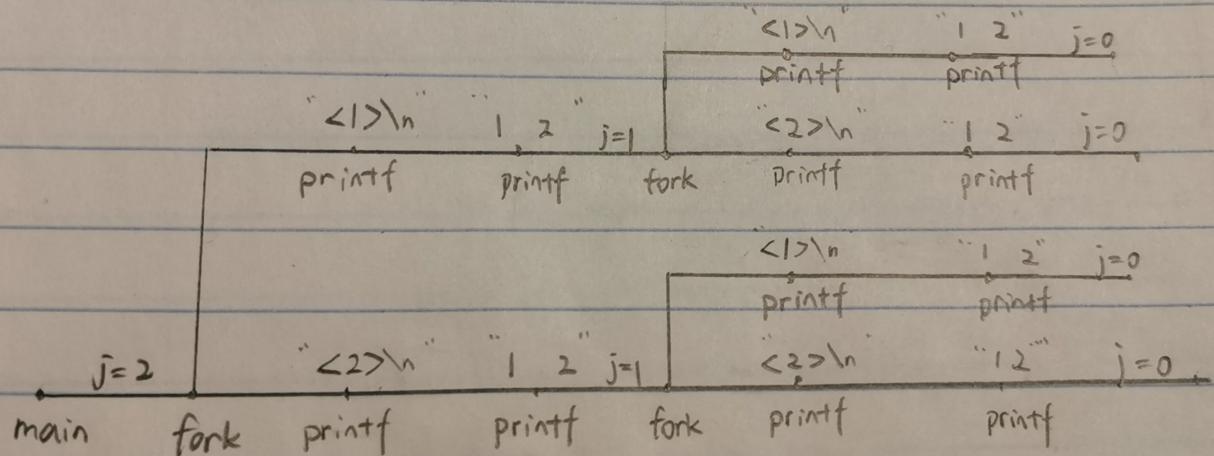
int main()
{
    pid_t id;
    int i;
    int j = 2;
    while (j > 0) {
        id = fork();
        if (id > 0) {
            printf("<1>\n");
        }
        else if (id == 0) {
            printf("<2>\n");
        }
        for (i = 1; i <= 2; i++)
            printf("%d ", i);

        j--;
    }
    return 0;
}
```

The process graph for this code is:

Runqing Cai / runqingc

SAQ1



SAQ2

Consider four processes with the following starting and ending times:

Process	Start Time	End Time
A	5	7
B	2	4
C	3	6
D	1	8

From the figure above, "A && C", "A && D", "B && C", "B && D", "C && D" are concurrent.

"A && B" are not concurrent because A starts at 5, after the B end, which is 4.

## **SAQ3**

Can the init process always terminate any process in the system? Explain.

In Linux, the init process, which is the first process started during the booting of the system (usually having a process ID of 1), plays a special role in the process management. However, while the init process has significant control over user-level processes, especially orphaned ones, it does not have absolute power to terminate any process in the system. Its capabilities are constrained by the Unix permissions model, process states, and the nature of the process (user-level vs. kernel-level).