

OneCompiler

43hq5fzet

AI NEW C RUN

1#include <stdio.h>

2

3int main() {

4 int n, m, i, j, k;

5 n = 5; // Number of processes

6 m = 3; // Number of resources

7

8 int alloc[5][3] = { { 0, 1, 0}, // P0

9 { 2, 0, 0}, // P1

10 { 3, 0, 2}, // P2

11 { 2, 1, 1}, // P3

12 { 0, 0, 2} }; // P4

13

14 int max[5][3] = { { 7, 5, 3}, // P0

15 { 3, 2, 2}, // P1

16 { 9, 0, 2}, // P2

17 { 2, 2, 2}, // P3

18 { 4, 3, 3} }; // P4

19

20 int avail[3] = {3, 3, 2}; // Available resources

21

22 int f[n], ans[n], ind = 0;

23 for (k = 0; k < n; k++) {

24 f[k] = 0;

25 }

26

27 int need[n][m];

28 for (i = 0; i < n; i++) {

29 for (j = 0; j < m; j++)

30 need[i][j] = max[i][j] - alloc[i][j];

31 }

32


STDIN

Input for the program (Optional)




Output:

Following is the SAFE Sequence

P1 -> P3 -> P4 -> P0 -> P2



main.c



Run

Output

Clear

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <string.h>
4 #include <stdlib.h>
5
6 int main() {
7     int fd[2];
8     pid_t pid;
9     char write_msg[] = "Hello from parent!";
10    char read_msg[100];
11
12    // Create the pipe
13    if (pipe(fd) == -1) {
14        perror("Pipe creation failed");
15        return 1;
16    }
17
18    // Fork a child process
19    pid = fork();
20
21    if (pid < 0) {
22        perror("Fork failed");
```

Child received: Hello from parent!

=== Code Execution Successful ===