

Λειτουργικά Συστήματα
Υπολογισμός αθροίσματος με χρήση
συνεργαζόμενων διεργασιών

Ονοματεπώνυμο: Στεφανιώρος Μιχαήλ

ΑΜ: 1072774

Έτος: 4^ο

Κώδικας:

```

1 // Sum calculation with child processes using fork and pipes
2
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <unistd.h>
6
7 int main() {
8     int sums[4];
9     int sum=0, j = 1, n, i, N;
10
11     printf("Give N:");
12     scanf("%d", &N);
13
14     int w = N/4;
15
16     // Create the pipes
17     int fd[4][2];
18     for(i=0; i<4; i++){
19         pipe(fd[i]);
20     }
21
22     // Create 4 child processes
23     for(n=0; n<4; n++) {
24         // After the first child process, change the limits for sum calculation
25         if(n>0){
26             j+=N/4;
27             w+=N/4;
28         }
29         // Child processes
30         if(fork() == 0) {
31             int tmpsum = 0;
32             for (j; j<=w; j++){
33                 tmpsum+=j;
34             }
35             printf("%d\n", tmpsum);
36             close(fd[n][0]); // close read
37             write(fd[n][1], &tmpsum, sizeof(tmpsum));
38             close(fd[n][1]); // close write
39             exit(0);
40         }
41         // Parent process waits for all child processes to finish
42         wait(NULL);
43     }
44     // The parent process takes the four sums and calculates the overall sum
45     for(i=0; i<4; i++){
46         close(fd[i][1]); // close write
47         read(fd[i][0], &sums[i], sizeof(sums[i]));
48         sum+=sums[i];
49     }
50     // Calculate the overall sum: sum = N*(N+1)/2 in steps to avoid overflow
51     int sum2 = N/2;
52     sum2 = sum2*(N+1);
53
54     printf("The total sum calculated from the sums of child processes is: %d\n", sum);
55     printf("The total sum calculated by parent process is: %d", sum2);
56
57     return 0;
58 }
59

```

Επεξήγηση Κώδικα:

Στις πρώτες γραμμές είναι φανερό ότι γίνεται μια απλή αρχικοποίηση των μεταβλητών των αθροισμάτων και αυτών που θα χρησιμοποιηθούν στα for loops.

```

16 // Create the pipes
17 int fd[4][2];
18 for(i=0; i<4; i++){
19     pipe(fd[i]);
20 }

```

Έπειτα, προχωρώ με την δημιουργία των pipes. Στην ουσία φτιάχνω έναν 4x2 πίνακα όπου κάθε γραμμή περιέχει τα read και write κάθε child process. Επέλεξα τα pipes ως τρόπο επικοινωνίας μεταξύ των διεργασιών, διότι η λειτουργία τους είναι built-in, κάνοντάς τα πιο εύκολα σε θέμα προγραμματισμού και έκτασης κώδικα. Στα pipes μπορούν να γράψουν πολλές διεργασίες και μόνο μια μπορεί να διαβάσει. Επομένως, κάθε child process γράφει το άθροισμα που υπολόγισε και τελικά τα διαβάζει η parent process.

```

22 // Create 4 child processes
23 for(n=0;n<4;n++) {
24     // After the first child process, change the limits for sum calculation
25     if(n>0){
26         j+=N/4;
27         w+=N/4;
28     }
29     // Child processes
30     if(fork() == 0) {
31         int tmpsum = 0;
32         for (j; j<=w; j++){
33             tmpsum+=j;
34         }
35         printf("%d\n", tmpsum);
36         close(fd[n][0]); // close read
37         write(fd[n][1], &tmpsum, sizeof(tmpsum));
38         close(fd[n][1]); // close write
39         exit(0);
40     }
41     // Parent process waits for all child processes to finish
42     wait(NULL);
43 }

```

Όπως φαίνεται στην παραπάνω φωτογραφία, οι child processes δημιουργούνται σε ένα for loop. Το γεγονός ότι στο τέλος κάθε loop, κάθε child process κάνει exit(0), βοηθάει ώστε κάθε φορά με την fork() να δημιουργείται μόνο ένα child. Επίσης, η parent δεν κάνει τίποτα μέχρι να τελειώσουν όλες οι θυγατρικές. Όπως είναι φανερό, τα child processes υπολογίζουν τα 4 αθροίσματα και γράφουν τα αποτελέσματα στα pipes.

```

44 // The parent process takes the four sums and calculates the overall sum
45 for(i=0; i<4; i++){
46     close(fd[i][1]); // close write
47     read(fd[i][0], &sums[i], sizeof(sums[i]));
48     sum+=sums[i];
49 }
50 // Calculate the overall sum: sum = N*(N+1)/2 in steps to avoid overflow
51 int sum2 = N/2;
52 sum2 = sum2*(N+1);
53
54 printf("The total sum calculated from the sums of child processes is: %d\n", sum);
55 printf("The total sum calculated by parent process is: %d", sum2);
56
57 return 0;
58 }
59

```

Τέλος, η parent process διαβάζει κάθε άθροισμα από τα pipes, τα αποθηκεύει σε μια λίστα και υπολογίζει το συνολικό άθροισμα.