

Sistemi Operativi – Lab 4 13.11.18 - A.A. 2018/2019 - Prof. L. Sterpone

Goal: gestione preliminare dei processi, albero di generazione dei processi e control flow graph.

Esercizio 1 (albero di generazione dei processi): Rappresentare il grafo di controllo del flusso e l'albero di generazione dei processi dei seguenti tratti di codice. Indicare inoltre l'output prodotto su video. Verificare il risultato predetto tramite esecuzione.

```
A.for (i=1; i<=2; i++)
{
    if (!fork ())
        printf ("%d\n", i);
}
printf ("%d\n", i);
```

```
B.for(i=3; i>1; i--)
{
    if (fork ())
        printf ("%d\n", i);
}
printf ("%d\n", i);
```

```
C.for (i=0; i<2; i++)
{
    if (fork ())
        fork ();
}
printf ("%d\n", i);
```

```
D.for (i=2; i>=1; i--)
{
    if (!fork ())
        printf ("%d\n", -i);
    else
        printf ("%d\n", i);
}
```

Esercizio 2 (albero di generazione dei processi e control flow graph – simile esame 21.09.16):

Rappresentare l'albero di generazione dei processi e il control flow graph del seguente programma C nel caso in cui x sia pari a 5 e n sia pari a 2. Determinare l'ordine di stampa dei messaggi a video e l'azione della system call `system()`.

Come va modificato il codice per poter gestire il caso in cui un processo figlio non è creato ?

Che cosa succede al variare dei parametri n ed x ?

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <sys/wait.h>
6
7 int main (int argc, char *argv[]) {
8     int i, n, x;
9     char str[50];
10
11     n = atoi (argv[1]);
12     x = atoi (argv[2]);
13     printf ("run with n=%d\n", n);
14     fflush (stdout);
15     for (i=0; i<n; i++) {
16         if (fork () > 0) {
17             printf ("%d", n-1);
18             sleep(x);
19         } else {
20             sprintf (str, "%s %d %s", argv[0], n-1, argv[2]);
21             system (str);
22         }
23     }
24
25     exit (0);
26 }
```