

Sistemi Operativi – Lab 6 18.11.19 - A.A. 2019/2020 - Prof. L. Sterpone

Goal: Utilizzo delle system call exec, fork e system. Approfondimento grafo di precedenza.

Esercizio 1 (system call exec):

Considerando il codice seguente, se ne descriva il funzionamento indicando che cosa produce su video e per quale motivo.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main (int argc, char **argv)
{
    int i = 0;
    fprintf (stdout, "%d %d\n", getpid(), ++i);
    execl (argv[0], argv[0], (char *) 0);
    fprintf (stdout, "End program\n");
    return (1);
}
```

Esercizio 2 (system call fork, exec e system):

Riportare l'albero di generazione dei processi a seguito dell'esecuzione del seguente tratto di codice C. Si indichi inoltre che cosa esso produce su video e per quale motivo.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main () {
    char str[100];
    int i;
    fork();
    for (i=0; i<2; i++){
        if (fork()!=0) {
            sprintf (str, "echo system with i=%d", i);
            system (str);
        } else {
            sprintf (str, "exec with i=%d", i);
            execlp ("echo", "myPgrm", str, NULL);
        }
    }
    return (0);
}
```

Esercizio 3 (grafo di precedenza):

Realizzare con le system call fork e wait il grafo di precedenza illustrato nella seguente figura. Ogni processo P produca un messaggio di stampa in cui viene stampato l'indice del processo il PID e il PID del padre. Verificare che le precedenze siano rispettate inserendo delle system call sleep nei vari rami del programma.

