

Prova in itinere (Recupero) 9 Gennaio 2017
(Laboratorio)

Università di Salerno

1. **Codice comportamentale.** Durante questo esame si deve lavorare da soli. Non si può consultare materiale di nessun tipo. Non si può chiedere o dare aiuto ad altri studenti.
2. **Istruzioni.** Rispondere alle domande. Per la brutta usare i fogli posti alla fine del plico (NON si possono usare fogli aggiuntivi); le risposte verranno corrette solo se inserite nello spazio ad esse riservate oppure viene indicata con chiarezza la posizione alternativa.
Per essere accettata per la correzione la risposta deve essere ordinata e di facile lettura.
TUTTE le risposte vanno GIUSTIFICATE. Ciascuna risposta non giustificata vale ZERO.

Nome e Cognome: _____

Matricola: _____

Firma _____

Spazio riservato alla correzione: non scrivere in questa tabella.

1	2	3	Tot
/14	/14	/22	/50

1. 14 punti

Si supponga di mandare in esecuzione il seguente programma:

```
int main()
{
    pid_t p;

    fork();
    p=fork();
    if (p==0) p=fork();
    if (p>0) printf("%d\n", getpid());

    exit(0);
}
```

Usando un albero che mostri l'evoluzione dei processi, dire quanti processi saranno presenti nel sistema durante l'esecuzione.

Quante `printf` saranno eseguite?

2. 14 punti

Si assuma di compilare ed eseguire il seguente programma

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

void exit1(void);
void exit2(void);

int main(void) {
    int pid;
    char array[]="Dopo\n";
    atexit(exit1);

    write(1,"Prima",5);

    if((pid=fork())<0)
        exit(1);

    if(pid==0) {
        atexit(exit2);
        printf("Sono il figlio\n");
        write(1,array,5);
        exit(0);
    }
    wait(NULL);
    write(1,array,5);
    _exit(0);
}

static void exit1(void) {
    printf("Sono il primo handler\n"); }
static void exit2(void) {
    printf("Sono il secondo handler\n"); }
```

Spiegare quale sar l'output del programma, distinguendo tra processo padre e processo figlio, giustificando in maniera esaustiva la risposta.

3. 22 punti

Si consideri il seguente sorgente C `attesa.c`

```
int main(void)
{
    pause();
}
```

e si compili con il comando `gcc attesa.c -o attesa`

Si consideri il seguente codice C

```
void handler(int);

int main(void)
{
    pid_t pid;
    signal(SIGTERM, handler);
    pid=fork();
    if(pid==0) {
        signal(SIGINT, handler);
        execl("./attesa", "attesa", NULL); }
    else pause();
}

void handler(int signum)
{ printf("Eccomi\n"); }
```

(a) Descrivere cosa accade compilando il programma ed eseguendo `./a.out` nella shell.

(b) Supponendo che il processo padre abbia come pid 1237 ed il processo figlio abbia pid 1238, descrivere cosa accade se da una seconda shell si esegue, rispettivamente:

- 1) `kill -SIGTERM 1237`
- 2) `kill -SIGTERM 1238`
- 3) `kill -SIGINT 1237`
- 4) `kill -SIGINT 1238`

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA