Sistemi Operativi: Prof.ssa A. Rescigno **Anno Acc. 2015-2016**

Prova in itinere 26 Aprile 2016

Università di Salerno

1. **Codice comportamentale.** Durante questo esame si deve lavorare da soli. Non si puó consultare materiale di nessun tipo. Non si puó 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	4	5	6	7	8	tot	bonus
/7	/7	/10	/16	/16	/16	/12	/12	/100	/10

NOTA: I primi 4 esercizi riguardano la gestione del sistema operativo UNIX

1. *7 punti*
   Si supponga di mandare in esecuzione il seguente programma:

```
int main(void)
{
pid_t p;

fork();
fork();

p = fork();

fork();

if (p==0) {fork();}

sleep(30);
exit(0);
}
```

   Usando un albero che mostra l'evoluzione dei processi, dire quanti processi sono presenti nel sistema durante i 30 secondi dell'istruzione sleep(30).

2. *7 punti*
   Si supponga di compilare e mandare in esecuzione il codice seguente. Sia 112 il pid di tale processo.

```
static void exit1(void);
static void exit2(void);

main() {
         pid_t pid;
         atexit(exit1);

         pid=fork();
         if(pid==0) {
             atexit(exit2);
             while (getppid() =! 1) ;
             printf("mio padre %d \n", getppid());
             exit(0);
             }
         else {
            printf("esce %d \n", getpid());
            _exit(0);
            }
}

static void exit1(void) {
printf("sono il primo handler");
return;
}
static void exit2(void) {
printf("sono il secondo handler");
return;
}
```

   Dire che cosa si ottiene su standard output, motivando la risposta.

3. *10 punti*
   Si assuma che il segnale `SIGKILL` sia definito come segue

```
#define SIGKILL  25             /* Default action: exit */
```

   Dato il seguente programma C, il cui eseguibile é `a.out`

```
(1)  void handler(int);

(2)  int main(void)
(3)  {
(4)  printf("Ciao \n");
(5)  signal(SIGINT, handler);
(6)  printf("Salut ");
(7)  sleep(30);
(8)  printf("Hello \n");
(9)  exit(0);}

(10)  void handler(int signum)
(11)  { printf("Hola \n"); }
```

   (a) Dire che cosa succede dando `a.out` senza l'arrivo di alcun segnale. Motivare la risposta.

   (b) nell'ipotesi che arrivi il segnale `SIGINT` durante l'esecuzione dello sleep. Motivare la risposta.

(c) nell'ipotesi che arrivi il segnale `SIGKILL` durante l'esecuzione dello sleep. Motivare la risposta.

4. *16 punti*
   Sia `Fattoriale` un eseguibile presente nella cwd che prendendo da linea di comando un intero ne restituisce il fattoriale su standard output. Cioé se a terminale si dá `Fattoriale 4` su standard output si ottiene `24`.

   Scrivere un programma C tale che:
   - genera n processi $P_1,\ P_2, \ldots\ P_n$, dove n é dato da linea di comando;
   - il processo P_i manda in esecuzione `Fattoriale i` utilizzando una delle funzioni `exec`;
   - i valori dei fattoriali devono essere scritti su standard output in ordine crescente, cioé deve essere eseguito prima P_1, poi P_2, poi P_3, etc.

5. *16 punti*
   Quattro processi indipendenti, P_1, P_2, P_3, P_4, possono richiedere anche piú di un CPU burst; appena un CPU burst di un processo ha terminato la propria esecuzione, il processo é pronto per l'esecuzione del successivo CPU burst (che non necessariamente durerá lo stesso tempo). I tempi di arrivo e di esecuzione sono descritti nella seguente tabella:

Processo	Tempo di arrivo	1^o CPU burst	2^o CPU burst	3^o CPU burst
P_1	0	10	6	4
P_2	2	3	2	2
P_3	3	2	-	-
P_4	5	1	1	-

   a) Si tracci il diagramma di Gantt per lo scheduling FCFS e si calcoli il tempo di attesa relativo a ciascun processo.
   b) Lo scheduling generato soffre dell' "effetto convoglio". Motivare la risposta data.

6. *16 punti*
   Si considerino tre processi, due produttori P_1 e P_2 ed un consumatore C_1, ed un buffer in cui é possibile inserire o prelevare 1 messaggio per volta. Sia
   - `enter(messaggio);` la funzione utilizzata dai processi produttori per inserire il messaggio
   - `remove(messaggio);` la funzione utilizzata dal processo consumatore per prelevare il messaggio.

   Inoltre, siano

```
semaphore empty=1;
semaphore full=0;
```

   le variabili condivise (assieme al buffer). Si considerino le seguenti porzioni di codice includenti la sezione critica per l'accesso al buffer da parte di P_1, P_2 e C_1.

	Processo P_1		Processo P_2		Processo C_1
	`while (1) {`		`while (1) {`		`while (1) {`
$a1$	`wait(empty);`	$b1$	`wait(empty);`	$c1$	`wait(full);`
$a2$	`enter(messaggio);`	$b2$	`enter(messaggio);`	$c2$	`remove(messaggio);`
$a3$	`signal(full);`	$b3$	`signal(full);`	$c3$	`signal(empty);`
	`}`		`}`		`}`

   a) Commentare la seguente sequenza di esecuzione:

$$a1, a2, b1, c1, a3, a1, c2$$

   indicando lo stato (in esecuzione, bloccato, pronto) in cui si trovano i tre processi ed il valore delle variabili semaforiche al termine dell'esecuzione di ogni istruzione e prima della eventuale schedulazione del processo successivo.

   b) Indicare quale istruzione sará eseguita successivamente a c2 nella sequenza data, descrivendone le conseguenze.

7. *12 punti*
   Si consideri la seguente successione di riferimenti a pagine:

   1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 3, 2, 7, 6, 3, 2, 1, 2, 3, 6.

   Si supponga che laccesso alle pagine **1, 2 , 3** sia sempre in scrittura.

   Determinare il tempo di accesso effettivo della paginazione su richiesta per LRU con 4 frame, se:
   - il tempo di servizio di un page fault senza salvataggio della pagina avvicendata di 80 millisecondi ($80 * 10^{-3}$ sec),
   - il tempo di servizio di un page fault con salvataggio della pagina avvicendata di 140 millisecondi ($140 * 10^{-3}$ sec)
   - il tempo di accesso alla memoria di 80 microsecondi ($80 * 10^{-6}$ sec),

8. *12 punti*
   Si assuma che un SO usi:
   - 33 bit per un indirizzo fisico
   - 34 bit per un indirizzo logico
   - frame di 2 Kb

   Giusticando le risposte, dire
   (a) Quanti bit uso per l'offset?
   (b) Con quanti bit identifico un frame?
   (c) Con quanti bit identifico una pagina?

   (d) Assumendo che le tabelle delle pagine includono anche il bit di validitá ed il bit di modifica, dire quanto é grande in byte la page table di un processo che usa tutte le pagine.

9. *10 punti (bonus)*
   "Secondo e contorno" é un ristorante specializzato in secondi e contorni. In cucina ci sono due chef: Giulio prepara un secondo mentre, in contemporanea, Giovanni realizza il contorno piú adatto.
   Quando entrambi hanno terminato il loro lavoro, **servono** sia il secondo che il contorno ad un cliente.
   Queste azioni si ripetono per tutto il giorno. Si noti che Giulio non puó servire il secondo finché Giovanni non ha finito di preparare il contorno e viceversa. Descrivere con uno pseudocodice i due processi Giulio e Giovanni utilizzando i semafori.

   NOTA: Nel codice utilizzare la frase
   - "Giulio prepara il secondo" per indicare il momento in cui Giulio prepara;
   - "Giulio consegna il secondo" per indicare il momento in cui Giulio consegna;
   - "Giovanni prepara il contorno" per indicare il momento in cui Giovanni prepara;
   - "Giovanni consegna il contorno" per indicare il momento in cui Giovanni consegna.

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA

FOGLIO DA UTILIZZARE PER LA BRUTTA