Ricordarsi di mettere il proprio nome, cognome, e numero di matricola in tutti i fogli. Motivare sempre le risposte date. Non e' necessario dare risposte molto lunghe, ma e' importante rispondere in modo motivato ed esauriente alle domande poste (in altre parole, molto meglio una frase in piu' che una in meno).

Per avere la sufficienza, e' **necessario svolgere ognuno dei primi 4 esercizi.**

Non sono ammesse macchinette calcolatrici o altre macchine elettroniche; non e' consentito uso di appunti o libri.

Malacopia: consegnare, se necessario **solo** gli esercizi che devono essere corretti (non riportati in bella copia); barrare quindi gli altri

Esercizio 1

1. A cosa serve il Process Control Block?
2. Quali informazioni contiene (indicare almeno 3 dei campi)?
3. Quanti Process Control Block esistono in un momento qualsiasi di attivita' di una macchina?

Risposta(Sketch) PCB rappresentano processi. Contiene informazioni importanti come process ID, process state, CPU state (program counter, register) per riprendere l'attivita' di un processo, informazioni per la gestione della memoria, etc. Il loro numero e' = numero dei processi (in generale c'e' una tabella per i PCB, quindi c'e' un upper bound)

Esercizio 2 Pensando all'architettura di un processore, che cosa indicano i termini pipelining, superscalarita' e multicore?

Esercizio 3

1. Qual'e' lo svantaggi principale del metodo di "file allocation" chiamato "Linked list" ?
2. Quali sono, sempre in questo metodo, i possibili pro e contro che emergono al variare delle dimensioni dei blocchi?

Risposta(Sketch) 1: Nel caso peggiore, un file system deve accedere tutti i blocchi di un file (caso di singly-linked list) per trovare i dati richiesti, o meta' (caso di doubly-linked list). 2: Un blocco grande riduce il numero di operazioni IO richieste per ritrovare un certo record, ma al costo di maggiore frammentazione interna; riducendo la dimensione e' il contrario

Esercizio 4 Riportare il codice di 2 thread che, quando lanciati in parallelo, possano stampare, come tutti e soli output possibili, le sequenze `CAPRI` e `CARPI`.

I thread possono usare operazioni di semaforo e operazioni di stampa di singole lettere (esempio `print R`).

Risposta(Sketch) Varie soluzioni possibili. Esempio

$$T_1 = S.P(); printR; T.V();;$$

$$T_2 = printC; printA; S.V(); printP; T.P(); print\, I$$

con inizialmente tutti i semafori a 0.

Esercizio 5 Si consideri il protocollo di mutual exclusion per i 2 processi mostrati sotto:

```
                                    // Shared variables
                                    boolean c1 = false, c2 = false;


// Process 1                                        // Process 2
<init1>;                                            <init2>;
while(true) {                                       while(true) {
     c1 = true;                                          c2 = true;
     while (c2) { };                                     while (c1) { };
     <crit1>;                                            <crit2>;
     c1 = false;                                         c2 = false;
}                                                   }
```

dove `crit1` e `crit2` sono sezioni critiche. Il protocollo fornisce una soluzione al problema di Sezione Critica? (Come al solito, motivare la risposta)

Risposta(Sketch) No, puo' portare a deadlock (e' l'algoritmo 2 visto a lezione)

Esercizio 6 Un hard disk ha la capienza di 2^{26} byte, ed e' formattato in blocchi da 1024 byte. Viene adottata una allocazione indicizzata dello spazio su disco.

1. Quanti bit occorrono per memorizzare il numero di un blocco?
2. Si consideri un file A della dimensione di 800 K byte. Un solo blocco indice e' sufficiente a indirizzare tutti i blocchi di A? Quanti accessi al disco sono necessari per leggere l'ultimo blocco di A?

Risposta(Sketch)

1. Ci vogliono 16 bit (2 byte) per memorizzare in numero di un blocco, ottenuti con $2^{26}/2^{10} = 2^{16}$
2. Un blocco indice puo' contenere $1024/2 = 514$ puntatori a blocco.

   Un solo blocco indice non e' sufficiente a indirizzare tutti i blocchi di A. Se si assume una allocazione indicizzata a schema concatenato, con un secondo blocco indice possiamo indirizzare tutti i blocchi de file. Poiche' e' gia' in RAM il numero del primo blocco indice, e' necessario leggere il primo blocco indice per recuperare il valore del secondo blocco indice, leggere il secondo blocco indice per sapere il numero dell'ultimo blocco del file, leggere questo blocco, per un totale di 3 letture in RAM (stesso risultato si ottiene assumendo una allocazione indicizzata a due livelli).

Esercizio 7
1. In un sistema in cui la memoria e' gestita con la paginazione, supponiamo che gli indirizzi logici siano di 31 bit, e che nella tabella di pagine si usino 8 byte per scrivere il numero di un frame. Quale dimensione minima dovrebbero avere le pagine di questo sistema per essere certi di non dover ricorrere ad una paginazione a piu' livelli?
2. Se in un sistema non si vuole usare una paginazione a piu' livelli, quale soluzione alternativa si puo' adottare?

Risposta(Sketch)

1. Poniamo 31 = m + n (m = bit usati per scrivere un numero di pagina, n = bit usati per scrivere l'offset).

   Allora il numero di entry della PT piu' grande, moltiplicato per la dimensione di una entry deve poter essere contenuto in una pagina /frame, ossia: $2^m \times 2^3 \leq 2^n$. Da cui: $m + 3 \leq n$. Poiche' $m = 31 - n$, risolvendo il semplice sistema si ha $n = 17$, ossia le pagine devono almeno essere grandi $2^{17} = 128$ Kbyte.

2. Una inverted page table

Esercizio 8 In un sistema con memoria virtuale, supponiamo che il grado di utilizzazione del processore (cpu) e del disco per la paginazione siano i seguenti:

1. cpu 15%, disco 95%
2. cpu 95%, disco 10%
3. cpu 20%, disco 5%

Ognuno di questi 3 casi: rappresenta una situzione ottimale o no? Perche'? Se la situazione non e' ottimale, cosa e' ragionevole faccia il sistema operativo?

Risposta(Sketch) 1. Thrashing: ridurre il livello di multiprogrammazione 2 okay 3. si puo' aumentare la multiprogrammazione

Esercizio 9 Per i processi indicati nella tabella sotto, disegnare un diagramma di Gant che illustri la loro esecuzione usando:

- First-Come First Served
- Shortest Job First preemptive
- Round Robin (quantum = 2)

Indicare anche, nel sistema con RR, il waiting time medio.

Processi	Tempo di arrivo	Tempo di esecuzione
A	0.000	4
B	2.001	7
C	3.001	2
D	3.002	2

Risposta(Sketch)
1. [A] 4 [B] 11 [C] 13 [D]15 2. [A]4 [C]6 [D]8 [B]15 3. [A]2[A]4[B]6[C]8[D]10[B]12[B]14[B]15
Waiting time : 0+6+3+5=14/4= 3.50

Esercizio 10 1. In protocolli di sicurezza, descrivere il significato dei termini: confidentiality; message integrity; end-point authentication.

2. Descrivere i passi che portano alla determinazione della chiave pubblica e privata nell'algoritmo di RSA.