Quesito 1 (punti 7). Cinque processi *batch*, identificati dalle lettere $A - E$ rispettivamente, arrivano all'elaboratore agli istanti $0, 2, 4, 6, 9$ rispettivamente. Tali processi hanno un tempo di esecuzione stimato di $4, 5, 6, 3, 2$ unità di tempo rispettivamente. Per ognuna delle seguenti politiche di ordinamento:

1. FCFS (un processo per volta, fino al completamento)
2. Round Robin (a divisione di tempo, senza priorità e con quanto tempo di ampiezza 2)
3. Round Robin (a divisione di tempo, con priorità e prerilascio e quanto di tempo di ampiezza 2)
4. SJF (senza considerazione di valori di priorità espliciti[1] e con prerilascio)

determinare, trascurando i ritardi dovuti allo scambio di contesto: (i) il tempo medio di risposta; (ii) il tempo medio di *turn around*; (iii) il tempo medio di attesa.
Ove la politica di ordinamento in esame consideri i valori di priorità, tali valori, mantenuti staticamente per l'intera durata dell'esecuzione, sono rispettivamente: $3, 5, 5, 2, 3$ (con 5 valore maggiore).
Nel caso di arrivi simultanei di processi allo stato di pronto, fatta salva l'eventuale considerazione del rispettivo valore di priorità, si dia la precedenza ai processi usciti dallo stato di esecuzione rispetto a quelli appena arrivati.

Quesito 2 (punti 7).
Dato il sistema descritto dalla seguente rappresentazione insiemistica delle assegnazioni di risorsa, dove $\mathcal{P}$ denota l'insieme dei processi, $\mathcal{R}$ l'insieme delle risorse R_i^j di indice i e molteplicità j, $\mathcal{E}(\mathcal{P})$ l'insieme delle richieste di accesso a risorse in $\mathcal{R}$ emesse da processi in $\mathcal{P}$ e *attualmente pendenti*, e $\mathcal{E}(\mathcal{R})$ l'insieme degli accessi *attualmente soddisfatti* di processi in $\mathcal{P}$ a risorse in $\mathcal{R}$:

$$\mathcal{P} = \{P_1; P_2; P_3; P_4; P_5; P_6\} \tag{1}$$
$$\mathcal{R} = \{R_1^2; R_2^1; R_3^2; R_4^1\} \tag{2}$$
$$\mathcal{E}(\mathcal{P}) = \{P_2 \to R_2; P_3 \to R_4; P_4 \to R_1; P_5 \to R_4; P_6 \to R_3\} \tag{3}$$
$$\mathcal{E}(\mathcal{R}) = \{R_1 \to (P_1, P_2); R_2 \to P_3; R_3 \to (P_4, P_5); R_4 \to P_6\} \tag{4}$$

si analizzi il grafo di allocazione delle risorse, determinando se il sistema i trovi attualmente in situazione di stallo o meno. Successivamente, si studi l'evoluzione dello stato del sistema ove il processo P_1 richiedesse accesso alla risorsa R_2 a partire dalla situazione data.

Quesito 3 (punti 7). Discutere *concisamente* le cause e le conseguenze della frammentazione a livello di partizione.
Descrivere il funzionamento concettuale di una utilità di deframmentazione.
Fornire almeno una ragione per la quale sistemi operativi diversi decidono diversamente rispetto al fornire o meno all'utente applicativo utilità di deframmentazione.

Quesito 4 (punti 4).
[4.A]: Dato un sistema di *swapping* e una memoria con zone disponibili di ampiezza: $10, 4, 20, 18, 17, 9, 12, 15$ KB, indicare quale area venga prescelta dalla politica first-fit a fronte della richiesta di caricamento di un segmento di ampiezza 12 KB:
1: 15 KB
2: 18 KB
3: 20 KB
4: 17 KB.

[4.B]: Sia dato un sistema di memoria con indirizzi virtuali suddivisi in 4 campi: a, b, c, d, i primi 3 dei quali siano utilizzati per indirizzare tre livelli gerarchici di tabelle delle pagine e il 4^o campo rapprenti l'*offset* entro la pagina selezionata. Indicare dall'ampiezza di quali campi dipende il numero di pagine indirizzate nel sistema:
1: da quella di tutti e quattro i campi
2: da quella del campo d
3: da quella del campo a e d
4: da quelle dei campi a, b, c.

[1] Esclusi ovviamente i valori di priorità impliciti determinati dalla durata dei processi.

[4.C]: Sia data memoria dotata di 4 *page frame*, inizialmente libere, e 8 pagine di memoria virtuale. Utilizzando la politica FIFO per il rimpiazzo delle pagine, indicare quanti *page fault* si verifichino a fronte della stringa di riferimenti: 0172327103:
1: 4
2: 3
3: 0
4: 2.

[4.D]: Quando una interruzione o una chiamata di sistema trasferiscono il controllo al sistema operativo l'area di *stack* necessaria per ospitare il calcolo risiede preferibilmente:
1: in un'area distinta e separata dallo *stack* del processo interrotto
2: nell'area *stack* del processo interrotto
3: in una zona di memoria privata del sistema operativo e staticalmente preallocata allo scopo
4: in un *page frame* liberato specificamente allo scopo.

Quesito 5 (punti 7). Sia dato un sistema di gestione della memoria principale basato su segmentazione con indirizzi logici e fisici espressi su 16 *bit*. Si consideri la tabella dei segmenti riportata di seguito, ove il prefisso $0x$ denota l'uso di notazione esadecimale:

Segmento	**Base**	**Limite**
$0x0$	$0x0219$	$0x600$
$0x1$	$0x2300$	$0x014$
$0x2$	$0x0090$	$0x100$
$0x3$	$0x1327$	$0x580$
$0x4$	$0x1952$	$0x096$
$\ldots$	$\ldots$	$\ldots$
$0xF$	$\ldots$	$\ldots$

Tabella 1: Tabella dei segmenti.

Si mostri graficamente la mappa di memoria corrispondente, indicando anche il suo grado percentuale di occupazione.
Si fornisca l'indirizzo fisico corrispondente ai seguenti indirizzi virtuali espressi in notazione decimale:

a.	$< 0, 1984 >$
b.	$< 1, 18 >$
c.	$< 2, 250 >$
d.	$< 3, 1400 >$
e.	$< 4, 112 >$

Indicare infine in modo nel quale i segmenti di indice $0 - 4$ possano essere mappati su disco, assumendo blocchi di ampiezza 1 KB, calcolando anche la quantità percentuale di memoria *sprecata* di conseguenza.

Soluzione 1 (punti 7).

- FCFS (un processo per volta, fino al completamento)

```
processo A  AAAA                    LEGENDA DEI SIMBOLI
processo B  --bbBBBBB               - non ancora arrivato
processo C  ----cccccCCCCCC         x (minuscolo) attesa
processo D  ------ddddddddDDD       X (maiuscolo) esecuzione
processo E  ---------eeeeeeeeeEE

CPU         AAAABBBBBCCCCCCDDDEE
coda        ..bbccccddddddeee..
            ......dddeeeeee.....
```

processo	tempo di risposta	tempo di attesa	tempo di *turn-around*
A	0	0	0+4=4
B	2	2	2+5=7
C	5	5	5+6=11
D	9	9	9+3=12
E	9	9	9+2=11
medie	5,00	5,00	9,00

- Round Robin (a divisione di tempo, senza priorità e con quanto di ampiezza 2)

```
processo A  AAAA                    LEGENDA DEI SIMBOLI
processo B  --bbBBbbBBbbbbbbB       - non ancora arrivato
processo C  ----ccCCccccCCccccCC    x (minuscolo) attesa
processo D  ------ddddDDdddddD      X (maiuscolo) esecuzione
processo E  ---------eeeeeEE        . coda vuota

CPU         AAAABBCCBBDDCCEEBDCC
coda        ..bbccbbddcceebbdc..
            ......ddcceebbddc...
            .........ebbddcc....
```

processo	tempo di risposta	tempo di attesa	tempo di *turn-around*
A	0	0	0+4=4
B	2	2+2+6=10	10+5=15
C	2	2+4+4=10	10+6=16
D	4	4+5=9	9+3=12
E	5	5	5+2=7
medie	2,60	6,80	10,80

- Round Robin (a divisione di tempo, con priorità e prerilascio e quanto di tempo di ampiezza 2)

```
processo A  AAaaaaaaaaaaaAA         LEGENDA DEI SIMBOLI
processo B  --BBBBbbB               - non ancora arrivato
processo C  ----ccCCcCCCC           x (minuscolo) attesa
processo D  ------ddddddddddDDD     X (maiuscolo) esecuzione
processo E  ---------eeeeeeEE       . coda vuota

CPU         AABBBBCCBCCCCAAEEDDD
coda        ..aaccbbcaaaaeedd...
            ....aaaaaeeeedd.....
            ......ddddddd.......
```

processo	risposta	tempo di attesa	*turn-around*
A	0	0+11=11	11+4=15
B	0	0+2=2	2+5=7
C	2	2+1=3	3+6=9
D	11	11+0=11	11+3=14
E	6	6+0=6	6+2=8
medie	3,80	6,60	10,60

- SJF (senza considerazione di valori di priorità espliciti e con prerilascio)

```
processo A  AAAA                        LEGENDA DEI SIMBOLI
processo B  --bbBBBBB                   - non ancora arrivato
processo C  ----ccccccccccCCCCCC        x (minuscolo) attesa
processo D  ------ddddddDDD             X (maiuscolo) esecuzione
processo E  ---------EE

CPU         AAAABBBBBEEDDDCCCCCC
coda        ..bbccdddddccc......
            ......ccccc.........
```

processo	risposta	tempo di attesa	*turn-around*
A	0	0	0+4=4
B	2	2	2+5=7
C	10	10	10+6=16
D	5	5	5+3=8
E	0	0	0+2=2
medie	3,40	3,40	7,40

Soluzione 2 (punti 7). La figura 1 riporta la versione grafica della rappresentazione insiemistica data.

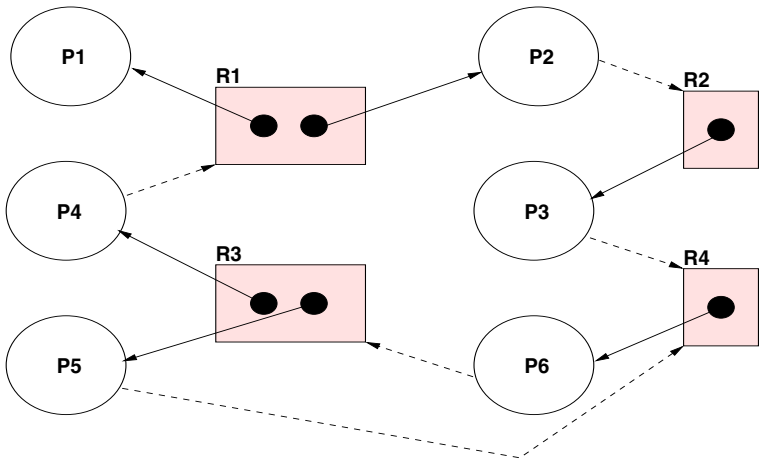

Figura 1: Grafo di allocazione delle risorse per il sistema dato, dove la freccia tratteggiata diretta da P_i a R_j denota una relazione in $\mathcal{E}(\mathcal{P})$ e la freccia solida diretta da R_j a P_i denota una relazione in $\mathcal{E}(\mathcal{R})$.

Allo stato si distinguono due percorsi chiusi:
percorso 1:

$$\{P_5 \rightarrow R_4 \rightarrow P_6 \rightarrow R_3 \rightarrow P_5\} \tag{5}$$

percorso 2:

$$\{P_2 \rightarrow R_2 \rightarrow P_3 \rightarrow R_4 \rightarrow P_6 \rightarrow R_3 \rightarrow P_4 \rightarrow R_1 \rightarrow P_2\} \tag{6}$$

in entrambi i quali è presente almeno una risorsa completamente impegnata ma a molteplicità > 1 e quindi con almeno un processo potenzialmente non implicato nel percorso chiuso. In situazioni di questo genere *non si può* trarre alcuna conclusione a priori sullo stato di stallo del sistema, ma occorre analizzare il problema dato nel suo specifico dettaglio.
Il percorso 1 condivide la risorsa R_3 a molteplicità > 1 con il percorso 2, il che comporta la stessa caratteristica per entrambi: o sono entrambi bloccati o si potranno liberare successivamente. Lo stato di stallo del percorso 1 dipende dall'evoluzione della risorsa R_3, cioè dalla possibilità che essa si liberi indipendentemente dai processi P_5 e P_6 implicati nel percorso. Lo stato della risorsa R_3 però dipende anche dal percorso 2, e quindi dall'evoluzione della risorsa R_1 (l'altra risorsa a molteplicità > 1). La risorsa R_1 è in possesso del processo P_1, che *non è incluso* nei due percorsi chiusi, e che quindi può procedere regolarmente e prima o poi rilasciare R_1. Quando ciò avverrà, P_4 potrà accedere alla risorsa R_1, e quindi riprendere ad avanzare. Al proprio completamento, P_4 rilascerà una molteplicità della risorsa R_3, che potrà essere devoluta al processo P_6, che potrà così a sua volta avanzare fino al proprio completamento. Infine verrà rilasciata anche la risorsa R_4, che potrà essere assegnata al processo P_5 o P_3. La sequenza di completamento e di rilascio procede poi a catena, consentendo la ripresa di tutti gli altri processi in attesa. Si puo pertanto concludere che la situazione proposta *non è di stallo*.
Qualora invece il processo P_1 richiedesse la risorsa R_1, partendo dalla situazione iniziale, questo creerebbe un nuovo percorso chiuso che manterrebe sospeso anche P_1, così determinando lo stallo completo del sistema.

Soluzione 3 (punti 7). Il fenomeno della frammentazione a livello di partizione è conseguenza diretta dell'allocazione di blocchi non contigui a *file*. Questo tipo di frammentazione produce un percepibile scadimento delle prestazioni a causa dei movimenti meccanici del disco necessari per il reperimento di blocchi non contigui di dati.
Le utilità di deframmentazione operano per porre rimedio a questo problema. Il loro funzionamento cerca di raggruppare i blocchi dati di ciascun *file* appartenente alla partizione quanto più contiguamente possibile. Per farlo tali utilità spostano progressivamente blocchi dati su zone libere della partizione così da liberare aree abbastanza grandi per allocarvi *file* interi.
Come è facile capire questa attività sul disco è molto delicata, impegnativa e incompatibile con ogni altra attività di sistema che operi autonomamente su zone della partizione. Per questo motivo, sistemi operativi con strategie di allocazione dei *file* minimamente ragionevoli non hanno convenienza a esporre all'utente applicativo una utilità di questo tipo, mentre possono prevedere l'impiego di strategie di deframmentazione in fasi non operative.

Soluzione 4 (punti 4).

Quesito	Risposta
[4.A]	3
[4.B]	4
[4.C]	Nessuna
[4.D]	1

La risposta corretta al quesito **[4.C]** è 6, che risulta da 4 *page fault* iniziali e 2 successivi, che si verificano alla 1^a richiesta della pagina 3 e alla 2^a richiesta della pagina 0.

Soluzione 5 (punti 7). La zona di memoria occupata dai segmenti di indice $0-4$ specificati dalla tabella 1 è rappresentata in figura 2.
Si noti che poiché non conosciamo le specifiche relativa ai rimanenti 11 segmenti indicizzabili dal sistema, ne ignoriamo la presenza nel rispondere alle domande poste dal quesito.
Come vediamo in figura l'area di memoria occupata dai segmenti in questione ha base minore all'indirizzo 0 e limite superiore all'indirizzo 8980. Poiché l'ampiezza complessiva dei segmenti in tale zona misura 3370 *Byte* ne risulta che un grado di occupazione pari al $37,53\%$.
In tabella 2 vediamo la risoluzione degli indirizzi virtuali specificati dal quesito.

indirizzo virtuale	indirizzo fisico
$< 0, 1984 >$	Errore: indirizzo illegale.
$< 1, 18 >$	$8960 + 0018 = 8978$
$< 2, 250 >$	$0144 + 0250 = 0394$
$< 3, 1400 >$	$4903 + 1400 = 6303$
$< 4, 112 >$	$6482 + 0112 = 6594$

Tabella 2: Risoluzione fisica degli indirizzi virtuali specificati dal quesito.

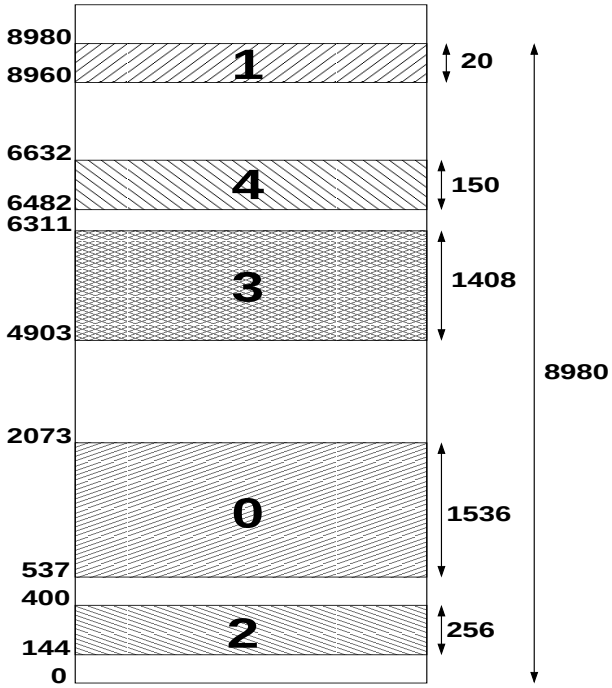

Figura 2: Rappresentazione grafica della memoria occupata dai segmenti definiti dalla tabella 1 del quesito.

In tabella 3 infine vediamo la mappatura dei segmenti dati su blocchi di disco di ampiezza 1 KB, ove naturalmente alloche-remo segmenti distinti su blocchi distinti. Dall'informazione in tabella si evince uno spreco di memoria pari al 53%.

Segmento	Ampiezza	# Blocchi occupati
0	1536 B	2
1	20 B	1
2	256 B	1
3	1408 B	2
4	150 B	1
Totale	3370 B	7

Tabella 3: Mappatura su disco dei segmenti dati.