

Esercizio 1 (4 punti): Consideriamo la seguente tabella delle pagine, in cui tutti i numeri presenti sono decimali. Assumiamo che la dimensione delle pagine sia **1024** byte. Assumiamo anche che tutte le numerazioni comincino da 0 (e che tutti gli indirizzi siano al byte e non alla parola).

Virtual page number	Page frame number
0	2
1	0
2	not allocated
3	1

- L'indirizzo virtuale **857** corrisponde all'indirizzo fisico:
- L'indirizzo virtuale **1699** corrisponde all'indirizzo fisico:

Esercizio 2 (6 punti): Definire la dimensione dei blocchi e la loro occupazione per soddisfare le richieste di memoria ricevute in un sistema di memoria di dimensione 2Megabyte organizzata secondo la tecnica del Buddy System, dove ogni partizione ha dimensione minima di 32 byte.

1. Processo A richiede 100 byte

Processo	A				
Dim. segmento	128				

2. Processo B richiede 770 byte

Processo		
Dim. Segmento		

3. Processo C richiede 180 byte

Processo		
Dim. Segmento		

4. Processo D richiede 500 byte

Processo		
Dim. Segmento		

5. Processo B rilascia la memoria

Processo		
Dim. Segmento		

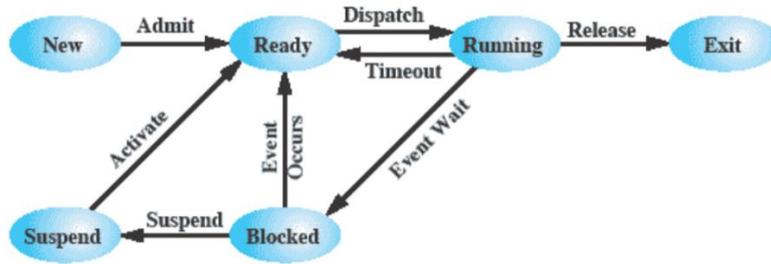
6. Processo A rilascia la memoria

Processo		
Dim. Segmento		

7. Processo F richiede 220 byte

Processo		
Dim. Segmento		

Esercizio 3 (10 punti): Completare la seguente tabella per mostrare come viene realizzata l'esecuzione dei processi specificati di seguito, considerando il diagramma seguente



L'ordine con cui vanno considerati i passaggi da uno stato all'altro – cioè *la priorità con cui viene eseguito il passaggio* – è il seguente:

1) ready, 2) new, 3) blocked, 4) running, 5) suspended.

N.B. se un processo running trova la coda ready piena va in blocked per un Δt , se trova la coda blocked piena, il primo blocked in coda va in suspended e la coda scorre per fare spazio al nuovo blocked.

Considerate un sistema con 4 processi:

- P1 consiste di 15 istruzioni – l'istruzione 5 è una richiesta di I/O e richiede $2\Delta t$
- P2 consiste di 10 istruzioni – l'istruzione 8 è una richiesta di I/O e richiede $2\Delta t$
- P3 consiste di 15 istruzioni – l'istruzione 7 è una richiesta di I/O e richiede $1\Delta t$
- P4 consiste di 10 istruzioni – l'istruzione 3 è una richiesta di I/O e richiede $1\Delta t$

Inserire "-" se non si vuole inserire un valore

Δt	Processo running (istruzioni in esecuzione)	Motivo interruzione (<i>T per timeout - C per completed - IO input/output</i>)	Coda dei processi ready (si entra da sinistra e si esce da destra)		Coda dei processi blocked (si entra da sinistra e si esce da destra)		Processi suspended
1	P1 (1 - 10)	IO istr. 5	- (-)	P2 (-)			
2							
3							
4							
5							
6							
7							
8							

Esercizio 4 (6 punti): Considerate la seguente sequenza di riferimenti alle pagine: 3, 1, 3, 4, 0, 4, 1, 3. Riempire le seguenti tabelle per mostrare l'allocazione di frame (**inserire i numeri di pagina**) e l'occorrenza di fault (**inserire F per fault e N per no**) nel caso vengano usati gli algoritmi di sostituzione OPT e FIFO.

pagina	3	1	3	4	0	4	1	3
OPT								
fault								

pagina	3	1	3	4	0	4	1	3
FIFO								
fault								

Parte 2 – Esercizi sulla bash

N.B. Questa parte va consegnata spedendola per email a massini@di.uniroma1.it (o caricandola sulla penna USB della professoressa)

Esercizio preliminare (3 punti)

Creare la directory **cognome-matricola** (usando il proprio cognome e la propria matricola) e dentro essa scrivere **uno script bash** denominato **crea-ambiente** che dovrà essere eseguito stando nella **directory cognome-matricola** ed eseguire le seguenti azioni:

- Crea la directory **esercizi** all'interno della directory **cognome-matricola**
- Crea i file **esercizio1**, **esercizio2** ed **esercizio3** all'interno della directory **esercizi**
- Crea la directory **esame** e all'interno di essa crea i file **fileA**, **fileB**
- Crea la directory **risultati** e all'interno di essa crea le directory **risultati1**, **risultati2**, e **risultati3**

Eeguire **crea-ambiente** stando nella directory **cognome-matricola**

Esercizio 1 (9 punti) - Nel file **esercizio1**, scrivere uno script **bash** che esegue le seguenti azioni (**dovrà essere eseguito stando nella directory **esercizi****):

- Usando un ciclo **for**
 - stampa dieci multipli di 2 a partire da 10 nel file **multipli** da posizionare nella directory **risultati1** specificando nella prima riga del file che cosa contiene
 - stampa le potenze di 3 minori di 20000 nel file **potenze** da posizionare nella directory **risultati2** specificando nella prima riga del file che cosa contiene
- Copia il contenuto del file **multipli** in un nuovo file chiamato **mul-pot** da posizionare nella directory **risultati2**, poi ci appende il contenuto del file **potenze**

N.B. Eseguire lo script **esercizio1** stando nella directory **esercizi**.

Esercizio 2 (10 punti)

Con un qualunque editor scrivere nel **fileA** (creato con il precedente esercizio) il nome dei mesi dell'anno con il relativo numero di giorni e il nome di 2 ipotetici amici nati in quel mese e il relativo giorno di nascita. Ogni linea del file sarà formattata nel seguente modo:

mese(spazio)giornidel mese *a* numero(spazio)amico1(spazio)giorno di nascita1 *a* numero(spazio)amico2 (spazio)giorno di nascita2 *a* numero

Esempio: gennaio 31 anno 13 marco 25

Nel file **esercizio2**, scrivere uno script **bash** che dovrà essere eseguito stando nella **directory **esercizi**** e che esegue le seguenti azioni

- Copia **fileA** presente nella directory **esame** nel file **amici** e lo posiziona nella directory **risultati2**
- Usando il comando **awk** cercare tutti i mesi che hanno 30 giorni e per ognuno di essi stampare la sequenza amico1 giorno di nascita1 mese in un nuovo file denominato **trenta** da collocare nella directory **risultati2**
- Usando il comando **awk** cercare i giorni di nascita maggiori di 15 e stampare la sequenza mese amico2 giorno di nascita2 in un nuovo file denominato **quindici** da collocare nella directory **risultati2**
- Estrarre dal file amici il nome di amico1 per tutti i mesi e stamparli uno per riga in un nuovo file denominato **amici-tutti** insieme, aggiungere poi al file amici-tutti il nome di amico2 per tutti i mesi e stamparli uno per riga. Infine ordinare il file **amici-tutti** e mettere il risultato in un nuovo file denominato **amici-ordinati**

N.B. Eseguire lo script **esercizio2** stando nella directory **esercizi**.

Esercizio 3 (8 punti)

Con un qualunque editor creare i seguenti file e metterli nella directory **esame**:

- file **date.dat** contenente la data compleanni di almeno tre familiari ognuna in una riga
- file **animali.dat** con il nome di quattro animali ognuno in una riga
- file **fiori.dat** contenente il nome di quattro fiori ognuna in una riga.

Nel file **esercizio3**, scrivere uno script *bash* che dovrà essere eseguito stando nella **directory esercizi** e che esegue le seguenti azioni

- controlla se nella directory ci sono file con estensione .doc e se non ci sono scrive su schermo: «Nessun file .doc»
- controlla se nella directory ci sono file con estensione .dat e per ogni file .dat appende il contenuto in un nuovo file chiamato **nuovo.txt** scrivendo su schermo il nome del file che si aggiunge

Eseguire lo script **esercizio3** stando nella directory **esercizi**.