Esame di Sistemi operativi

30 gennaio 2024 – Prof.ssa Massini

Cognome …… Nome …… matricola ……

Per superare l'esame occorre ottenere:

- Parte 1 - domande ed esercizi: almeno 18 punti su 30
- Parte 2 - programmazione bash: almeno 18 su 30

Parte 1 – Domande ed esercizi

Domanda a risposta aperta 1 *(4 punti)*
Descrivere brevemente la **gestione della memoria** da parte del sistema operativo

Domanda a risposta aperta 2 *(4 punti)*

Descrivere brevemente i principali compiti e le caratteristiche di un sistema operativo.

Le seguenti domande a risposta multipla valgono 1 punto

1. Quale dei seguenti elementi non è una delle parti che definiscono un **processo**?
   a) Il codice macchina da eseguire
   b) L'insieme delle risorse detenute
   c) Il processore
   d) Lo stato o modalità

2. Quale delle seguenti affermazioni è vera per l'architettura di **Von Neumann**?
   a) Comprende: processore, memoria, dispositivi di input/output e bus
   b) Comprende solo processore e memoria
   c) Comprende: processore, bus, memoria principale e memoria secondaria
   d) Comprende: memoria, dispositivi di input/output e bus

3. Quale dei seguenti elementi non fa parte del **Process Control Block** (PCB)?
   a) L'identificatore del processo
   b) Lo stato o modalità
   c) Il codice macchina da eseguire
   d) Il puntatore alla tabella delle pagine

4. Quale tra le seguenti transizioni NON è possibile nel **modello di processo** a cinque stati:
   a) da ready a running
   b) da ready a blocked
   c) da running a ready
   d) da running a blocked

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 **2048** byte. Assumiamo anche che tutte le numerazioni comincino da 0.

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

- L'indirizzo virtuale **3993** corrisponde all'indirizzo fisico:

- L'indirizzo virtuale **1472** 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 80 byte

Processo
Dim. segmento

2. Processo B richiede 600 byte

Processo	
Dim. Segmento	

3. Processo C richiede 200 byte

Processo	
Dim. Segmento	

4. Processo D richiede 400 byte

Processo	
Dim. Segmento	

5. Processo B rilascia 600 byte

Processo	
Dim. Segmento	

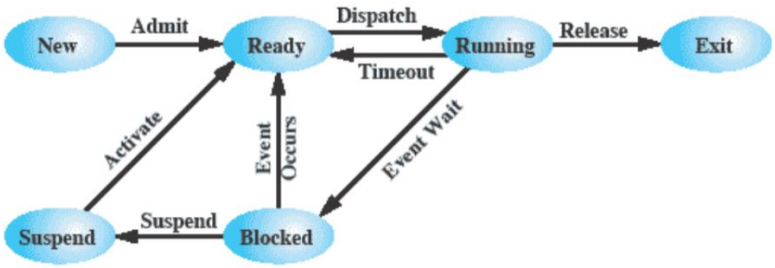
6. Processo A rilascia 80 byte

Processo	
Dim. Segmento	

7. Processo F richiede 240 byte

Processo	
Dim. Segmento	

Esercizio 3 (8 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 **2Δt**
- P2 consiste di 20 istruzioni – l'istruzione 12 è una richiesta di I/O e richiede **2Δt**
- P3 consiste di 15 istruzioni – l'istruzione 7 è una richiesta di I/O e richiede **1Δt**
- P4 consiste di 25 istruzioni – l'istruzione 3 è una richiesta di I/O e richiede **1Δt**

Inserire "-" se non si vuole inserire un valore

Δt	**Processo running** (istruzioni in esecuzione)	Motivo interruzione (***T*** *per timeout* **-** ***C*** *per completed* **-** ***IO*** *input/output*)	**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							

Parte 2 – Esercizi sulla bash

N.B. Questa parte va consegnata spedendola per email a massini@di.uniroma1.it

Esercizio preliminare (3 punti)

Scrivere **uno script *bash*** denominato **crea-ambiente** che dovrà essere eseguito stando nella **directory home** ed esegue le seguenti azioni:

- Crea la directory **cognome-matricola** (usando il proprio cognome e la propria matricola)
- 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*** e ***fileB***
- Crea la directory ***risultati*** e all'interno di essa crea le directory ***risultati1***, ***risultati2***, e ***risultati3***

Esercizio 1 (8 punti)

Con un qualunque editor scrivere nel ***fileB*** (creato con il precedente esercizio) il titolo e almeno 8 righe di una poesia o di una canzone.

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

- Sposta il file ***fileB*** presente nella directory ***esame*** nel file ***testo*** e lo posiziona nella directory ***risultati1***
- Usando il file ***testo*** produrre le seguenti informazioni e metterle nel nuovo file chiamato ***info-testo*** da posizionare nella directory ***risultati1***:
    - Titolo della poesia/canzone
    - numero di righe
    - numero di parole
    - numero caratteri della riga più lunga
    - la seconda e la terza parola di ogni verso della poesia/canzone
    - la poesia/canzone per intero

Esercizio 2 (9 punti)

Con un qualunque editor scrivere nel ***fileA*** (creato con il precedente esercizio) il nome di quattro ipotetici esami previsti nella sessione gennaio-febbraio e di quattro esami previsti nella sessione giugno-luglio con il relativo giorno, mese ed orario.
Ogni linea del file sarà formattata nel seguente modo:
giorno *a numero*(spazio)mese *a lettere*(spazio)ore(spazio)orario *a numero*(spazio)esame
Esempio: 3 febbraio ore 10:30 geometria

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

- Copia ***fileA*** presente nella directory ***esame*** nel file ***appelli*** e lo posiziona nella directory ***risultati2***
- Usando il comando `awk` cercare gli esami che si terranno nel mese di gennaio e stampare la sequenza esame, ora e giorno in un nuovo file denominato ***gennaio*** da collocare nella directory ***risultati2*** per tutti gli esami che soddisfano la condizione
- Usando il comando `awk` cercare gli esami che si terranno nel mese di febbraio e stampare la sequenza esame e **ora incrementata di 2** in un nuovo file denominato ***febbraio*** da collocare nella directory ***risultati2*** per tutti i esami che soddisfano la condizione

- Estrarre il nome degli esami di giugno e di luglio e stampare il nome dell'esame e il mese in un nuovo file denominato ***estivi***. Ordinare tale file e mettere il risultato in un nuovo file denominato ***estivi-ordinati***

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

Esercizio 3 (10 punti)

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

* file **mesi.dat** contenente i mesi dell'anno in righe diverse
* file **date.dat** con almeno tre date a vostra scelta nel formato che preferite in righe diverse
* file **giorni.dat** contenente i giorni della settimana in righe diverse

Nel file ***esercizio3***, scrivere uno script *bash* (che dovrà essere eseguito stando nella **directory *esercizi***) 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 crea un nuovo file con estensione .txt e ci appende il contenuto del file .dat scrivendo su schermo il nome del file che si aggiunge

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