Esame Sistemi Operativi - Operating Systems Exam
2023/02/03

Ex 1 (2.0 points)

Italiano

Si indichino quali delle seguenti affermazioni sono corrette relativamente ai file system. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Indicate which of the following statements referred to file systems are correct. Note that incorrect answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ⬜ Il file system a grafo di tipo ciclico non permette la condivisione di elementi. The filesystem of type cyclic graph does not allow sharing of entries.
2. ✅ Le system call read() e write() possono essere utilizzate per leggere e scrivere testo, caratteri e sequenze di byte. The system calls read() and write() can be used to read and write text, characters, and sequences of bytes.
3. ⬜ La system call open() ha esattamente tre parametri. The system call open() has exactly three parameters.
4. ⬜ La system call read() ritorna il numero di oggetti letti, come nel caso della funzione fread(). The system call read() returns the number of objects read, as in the case of the function fread().
5. ✅ Quando un file viene creato attraverso la funzione open(), contestualmente si possono impostare i permessi del file creato. When a file is created using the open() system call, the permissions of the created file can be set at the same time.

Ex 2 (2.0 points)

Italiano

Indicare quanti caratteri ‘A’, ‘B’, ‘C’ e ‘D’ sono visualizzati su standard output dal seguente programma. Se ad esempio fossero visualizzati 6 caratteri ‘A’, 3 caratteri ‘B’, 5 caratteri ‘C’ e 20 caratteri ‘D’, riportare la risposta con esattamente il seguente formato: A=6 B=3 C=5 D=20.

English

Indicate how many characters ‘A’, ‘B’, ‘C’ and ‘D’ are printed on standard output by the following program. If for example the displayed characters are 6 characters ‘A’, 3 characters ‘B’, 5 characters ‘C’, and 20 characters ‘D’, report the answer exactly in the following format: A=6 B=3 C=5 D=20.

```
int main () {
  int i;
  int pid;

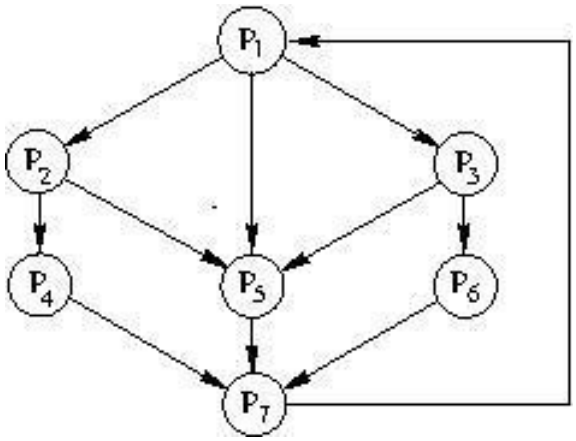
  for (i=1; i<=2; i++) {
    pid = fork ();
    fork();
    printf("A\n");
    if (pid==0) {
      fork();
      printf("D\n");
    }
    execlp("echo", "B", "C", NULL);
    printf("B\n");
  }
  return 1;
}
```

Risposta. Answer.
A=4 B=0 C=6 D=4

Ex 3 (5.5 points)

Italiano
Realizzare il seguente grafo di precedenza, utilizzando il minimo numero di semafori. I processi rappresentati devono essere processi ciclici (con corpo del tipo while(1)). Utilizzare le primitive init, signal, wait e destroy. Indicare gli eventuali archi superflui e riportare il corpo dei processi (P1, ..., P7) e l'inizializzazione dei semafori.

English
Implement the following precedence graph, using the minimum number of semaphores. The processes represented must be cyclical (with a body of type while(1)). Use the primitives init, signal, wait, and destroy. Indicate any unnecessary arcs and report the body of the processes (P1, ..., P7), and the initialization of the semaphores.



Soluzione. Solution.
The edge P1->P5 is redundant and can be ignored.

```
Initialization:
init (s1, 1); init (s2, 0); init (s3, 0);  ... ; init (s7, 0);

Termination:
destroy (s1); ...; destroy (s7);

P1
while (1) {
  wait (s1);
  printf ("P1\n");
  signal (s2);
  signal (s3);
  // REDUNDANT signal (s5);
}

P2
while (1) {
  wait (s2);
  printf ("P2\n");
  signal (s4);
  signal (s5);
}

P3
```

```
while (1) {
  wait (s3);
  printf ("P3\n");
  signal (s5);
  signal (s6);
}

P4
while (1) {
  wait (s4);
  printf ("P3\n");
  signal (s7);
}

P5
while (1) {
  wait (s5);
  // REDUNDANT from P1 wait (s5);
  wait (s5);
  printf ("P7\n");
  signal (s7);
}

P6
while (1) {
  wait (s6);
  printf ("P6\n");
  signal (s7);
}

P7
while (1) {
  wait (s7);
  wait (s7);
  wait (s7);
  printf ("P7\n");
  signal (s1);
}
```

Ex 4 (3.0 points)

Italiano

Si supponga che il seguente programma venga eseguito con il valore 3 sulla linea di comando. Si riporti esattamente l'output generato. Si prega di riportare la risposta su un'unica riga, indicando i vari messaggi e valori di output separati da un unico spazio. Non inserire nessun altro carattere nella risposta.

English

Suppose that the following program is executed passing the value 3 on the command line. Report the exact output generated. Please, report the response on one line, indicating all the output messages and values separated by a single space. Do not insert any other character in the response.

```
sem_t s1, s2;

void *tA (void *);
void *tB (void *);

void *tA (void *p){
  pthread_t thA;
```

```
  long int n;

  pthread_detach (pthread_self ());

  sem_wait (&s1);
  n = (long int) p;
  printf ("tA%ld ", n);
  sem_post (&s2);
  n--;
  if (n>0)
    pthread_create (&thA, NULL, tA, (void *) n);

  pthread_exit (NULL);
}

void *tB (void *p){
  pthread_t thB;
  long int n;

  pthread_detach (pthread_self ());

  sem_post (&s1);
  sem_wait (&s2);
  n = (long int) p;
  printf ("tB%ld ", n);
  n--;
  if (n>0)
    pthread_create (&thB, NULL, tB, (void *) n);

  pthread_exit (NULL);
}

int main (int argc, char *argv[]) {
  pthread_t thA, thB;
  long int n;

  n = atoi (argv[1]);

  sem_init (&s1, 0, 0);
  sem_init (&s2, 0, 0);
  setbuf (stdout, 0);
  pthread_create (&thA, NULL, tA, (void *) n);
  pthread_create (&thB, NULL, tB, (void *) n);

  pthread_exit (NULL);
}
```

Risposta. Answer.
tA3 tB3 tA2 tB2 tA1 tB1

Ex 5 (2.0 points)

Italiano
Dato il seguente elenco di comandi, si indichino quali delle seguenti affermazioni sono corrette dopo averli eseguiti. Si osservi che risposte errate implicano una penalità nel punteggio finale.
English

Given the following list of commands, indicate which of the following statements are correct after their execution. Note that incorrect answers imply a penalty in the final score.

```
echo "str1" > f1
ln f1 f2
echo "str2" >> f2
cat f1
cat f2
ln -s f3 f4
echo "str3" > f4
rm f3
cat f4 #First
echo "str4" >> f3
cat f4 #Second
mkdir dir
ln dir x
```

Scegli una o più alternative. Choose one or more options.

1. ⬜ Il comando "`cat f1`" stampa solo "`str1`". The command "`cat f1`" prints only "`str1`".
2. ⬜ Il comando "`cat f2`" stampa solo "`str2`". The command "`cat f2`" prints only "`str2`".
3. ✅ Il primo comando "`cat f4`" (identificato da `#First`) stampa un errore. The first command "`cat f4`" (identified by `#First`) prints an error.
4. ✅ Il secondo comando "`cat f4`" (identificato da `#Second`) stampa solo "`str4`". The second command "`cat f4`" (identified by `#Second`) prints only "`str4`".
5. ✅ Se alla fine del programma viene rimosso il file `f1`, il suo contenuto non viene perso perché ancora accessibile tramite `f2`. If at the end of the program the file `f1` is removed, its content is not lost because its content is still accessible through `f2`.
6. ✅ La dimensione del file `f4` è pari alla lunghezza del path originale, i.e., "f3" uguale a 2 byte. The size of file `f4` is equal to the length of the original path, i.e., "f3" equal to 2 bytes.
7. ⬜ La directory "`dir`" può essere acceduta anche tramite "`x`". The directory "`dir`" can be accessed also by means of "`x`".

Ex 6 (2.0 points)

Italiano

Si faccia riferimento all'utilizzo dei segnali in ambiente UNIX/Linux. Si indichi quali delle seguenti affermazioni sono corrette. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Refer to the use of signals in a UNIX/Linux environment. Indicate which of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ✅ Attraverso la system call signal() si può decidere di ignorare ALCUNI segnali. By using the system call signal(), you can decide to ignore SOME signals.
2. ✅ L'uso di funzioni non rientranti all'interno di un signal handler può portare a risultati non predicibili. The use of non-reentrant functions within a signal handler can lead to unpredictable results.
3. ⬜ Quando un processo riceve un segnale esso viene SEMPRE terminato dal sistema operativo. When a process receives a signal it is ALWAYS terminated by the operating system.
4. ⬜ Nella sincronizzazione di processi, le primitive semaforiche sem_post() e sem_wait() sono EQUIVALENTI e possono essere SOSTITUITE dalle funzioni kill() e pause(). In process synchronization, the primitives sem_post() and sem_wait() are EQUIVALENT and may be REPLACED by the functions kill() and pause().
5. ✅ L'esecuzione di un signal handler a seguito della ricezione di un segnale da parte di un processo può portare a race conditions. The execution of a signal handler after the reception of a signal by a process can lead to race conditions.

6. ⬜ La ricezione del segnale SIGKILL può essere ignorata. The reception of the SIGKILL signal can be ignored.
7. ✅ La ricezione del segnale SIGUSR2 può essere ignorata. The reception of the signal SIGUSR2 can be ignored.

Ex 7 (2.0 points)

Italiano

Relativamente all'implementazione delle pipe in un sistema operativo UNIX/Linux, si indichino quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Regarding implementing pipes in a UNIX/Linux operating system, indicate which of the following statements are correct. Note that wrong answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ✅ L'uso della system call write() su una pipe in cui tutti gli estremi di lettura sono stati chiusi provoca la generazione di uno specifico segnale. The use of the system call write() on a pipe where all reading ends are close, causes the generation of a specific signal.
2. ⬜ L'uso della system call write() su una pipe che è piena restituisce un errore. The use of the system call write() on a full pipe returns an error.
3. ✅ La system call write() eseguita su una pipe piena è normalmente bloccante. The system call write() executed on a full pipe is normally blocking.
4. ✅ La system call read() eseguita su una pipe vuota è normalmente bloccante. The system call read() executed on an empty pipe is normally blocking.
5. ✅ Se il numero di byte scritti su una pipe è minore o uguale a PIPE_BUF, essi vengono scritti in modo atomico. If the number of bytes written in a pipe is less or equal to PIPE_BUF, they are written atomically.
6. ⬜ PIPE_BUF rappresenta il numero massimo di byte che possono essere contenuti in una pipe. PIPE_BUF represents the maximum number of bytes that can be stored in a pipe.
7. ⬜ L'uso della system call read() su una pipe in cui tutti gli estremi di scrittura sono stati chiusi provoca la generazione di un segnale di tipo SIGPIPE. The use of the system call read() on a pipe, in which all writing ends have been closed, causes the generation of a SIGPIPE signal.

Ex 8 (2.0 points)

Italiano

Relativamente alla differenza tra thread e processi, si indichino quali delle seguenti affermazioni sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale..

English

Regarding the difference between threads and processes, indicate which of the following statements are correct. Note that wrong answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ✅ I processi hanno spazi di indirizzamento separati, i thread di un processo condividono lo stesso spazio di indirizzamento. Processes have separate address spaces, the threads of a same process share the same address space.
2. ⬜ I tempi di creazione di un processo e di un thread sono equivalenti. The times necessary to create a process and a thread are equivalent.
3. ✅ La schedulazione dei kernel thread è gestita dal kernel del sistema operativo. The scheduling of kernel threads is managed by the kernel of the operating system.
4. ⬜ La schedulazione degli user thread è gestita dal kernel del sistema operativo. The scheduling of user threads is managed by the kernel of the operating system.
5. ✅ L'operazione di context switching dei thread è più veloce di quella dei processi. Context switching of threads is faster than the one of processes.
6. ⬜ Nel caso di user thread, il parallelismo è effettivo, cioè possono esserci più thread eseguiti contemporaneamente. In the case of user threads, execution is actually parallel, i.e., there can be multiple threads running concurrently.

7. ✅ Gli user thread sono gestiti tramite una libreria utente. User threads are managed by means of a user library.

Ex 9 (2.0 points)

Italiano

Dire quali delle seguenti risposte relative agli algoritmi di scheduling sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Indicate which of the following statements referred to scheduling algorithms are correct. Note that incorrect answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ✅ In riferimento all'algoritmo round robin (RR), la riduzione del "quanto temporale" comporta una riduzione del tempo di risposta (response time) medio. With reference to the round robin (RR) algorithm, the reduction of the "time quantum" implies a reduction in the average response time.
2. ⬜ In riferimento all'algoritmo round robin (RR) l'incremento del "quanto temporale" comporta una riduzione del throughput. With reference to the round robin (RR) algorithm, the increase of the "time quantum" implies a reduction of the throughput.
3. ⬜ A parità di altri indici qualitativi, un algoritmo di schedulazione con turnaround time maggiore è migliore. If the other qualitative indices are equal, a scheduling algorithm with a longer turnaround time is better.
4. ✅ Uno dei principali problemi nell'implementazione dell'algoritmo shortest-job first (SJF) è la stima del CPU burst time dei processi coinvolti. One of the main drawbacks in the implementation of the shortest-job first (SJF) algorithm is the estimation of the CPU burst time of the involved processes.
5. ✅La media esponenziale (exponential average) per la stima del CPU burst time da peso maggiore ai campioni più recenti. The exponential average for estimating CPU burst time gives greater weight to more recent samples.

Ex 10 (2.0 points)

Italiano

Dire quali delle seguenti risposte relative al deadlock sono vere. Si osservi che risposte errate implicano una penalità nel punteggio finale.

English

Indicate which of the following statements referred to deadlock are correct. Note that incorrect answers imply a penalty in the final score.

Scegli una o più alternative. Choose one or more options.

1. ⬜ Una risorsa con possibilità di prelazione, quale ad esempio la CPU, può portare a deadlock. A resource with the possibility of preemption, such as the CPU for example, can lead to deadlock.
2. ✅ Se in un programma concorrente le uniche funzioni bloccanti sono le funzioni `sem_wait()`, e tutte sono sostituite con la funzione `sem_trywait()`, questo programma non può essere soggetto a deadlock. If in a concurrent program the only blocking functions are `sem_wait()`, and all are substituted with the function `sem_trywait()`, this program cannot be subject to deadlock.
3. ✅Siano A, B, C, D dei semafori inizializzati a 1. Il thread T1 esegue il codice `sem_wait(A); sem_wait(B); sem_wait(D); sem_post(A); sem_post(B); sem_post(D);`. Il thread T2 esegue il codice `sem_wait(C); sem_wait(D); sem_post(D); sem_post(C);`. Infine, il thread T3 esegue il codice `sem_wait(B); sem_wait(D); sem_post(B); sem_post(D);`. Se non ci sono altri thread che usano questi semafori, si può affermare che questi thread non sono soggetti a deadlock. Let A, B, C, D semaphores initialized to 1. The thread T1 executes the code `sem_wait(A); sem_wait(B); sem_wait(D); sem_post(A); sem_post(B); sem_post(D);`. The thread T2 executes the code `sem_wait(C); sem_wait(D); sem_post(D); sem_post(C);`. Finally, the thread T3 executes the code `sem_wait(B); sem_wait(D); sem_post(B); sem_post(D);`. If there are no other threads using these semaphores, it can be stated that these threads are not subject to deadlock.

4. ⬜ Le condizioni necessarie per il deadlock sono mutua esclusione, possesso e attesa, prelazione e l'assenza di attesa circolare. The necessary conditions to have a deadlock are mutual exclusion, hold and wait, preemption, and no circular wait.

Ex 11 (3.0 points)

Italiano

Lo stato di un sistema con 5 processi e 3 tipologie di risorse è definito dalle matrici riportate alla fine della domanda. Si supponga il processo P1 effettui una richiesta per le risorse {1, 1, 1}. Si stabilisca se la richiesta può essere soddisfatta permettendo al sistema di rimanere in uno stato sicuro. In caso affermativo si risponda "YES" e si riporti la sequenza di esecuzione dei processi. Ad esempio, si risponda "YES 5 4 3 2 1" nel caso in cui la sequenza sicura sia P5, P4, P3, P2, P1. In caso contrario, si risponda "NO".

English

The state of a system with 5 processes and 3 types of resources is defined by the matrices reported at the end of the question. Suppose that process P1 requests the resources {1, 1, 1}. Indicate whether the request can be granted, allowing the system to remain in a safe state. In the affirmative case, respond with "YES" and report the sequence in which the processes will be executed. For instance, respond "YES 5 4 3 2 1" in the case the safe sequence is P5, P4, P3, P2, P1. Otherwise, respond with "NO".

Processo **Process**	**Assegnate** **Allocation**	**Massimo** **Max**	**Disponibile** **Available**
P1	0 0 0	4 3 5	4 3 3
P2	0 0 1	4 4 2	
P3	0 2 0	3 2 2	
P4	1 0 1	2 5 4	
P5	1 1 0	3 4 2	

Fasi operative. Workflow.

```
Step  0;                                                    Disponibili:  3  2  2
Step:  1   RunProc:  3   Necessita':  3  0  2    -->      Disponibili:  3  4  2
Step:  2   RunProc:  5   Necessita':  2  3  2    -->      Disponibili:  4  5  2
Step:  3   RunProc:  2   Necessita':  4  4  1    -->      Disponibili:  4  5  3
Step:  4   RunProc:  4   Necessita':  1  5  3    -->      Disponibili:  5  5  4
Step:  5   RunProc:  1   Necessita':  3  2  4    -->      Disponibili:  6  6  5
```

Risposta. Answer,

YES 3 5 2 4 1

Ex 12 (3.0 points)

Italiano

Si consideri il seguente insieme di processi schedulati con un quanto temporale di 10 unità di tempo. Rappresentare mediante diagramma di Gantt l'esecuzione di tali processi utilizzando gli algoritmi First Come First Served (FCFS) e Priority Scheduling (PS), al fine di calcolare il tempo di terminazione di ciascun processo e il tempo di attesa medio. Tra i due, si riporti lo scheduling più vantaggioso dal punto di vista del tempo di attesa medio. Si prega di riportare la risposta su un'unica riga, indicando i tempi di terminazione di P1, P2, P3, P4 e P5 seguiti dal tempo di attesa medio. Separare i numeri con un unico spazio. Riportare il tempo di attesa medio con 1 sola cifra decimale. Non inserire nessun altro carattere nella risposta. Esempio di risposta corretta: 20 23 11 45 67 30.5

English

Consider the following set of processes, which are scheduled with a temporal quantum of 10 units. Represent using a Gantt diagram the execution of these processes using the First Come First Served (FCFS) and Priority Scheduling (PS) algorithms, in order to compute the termination time of each process, and the average waiting time. Compare the two schedulings, and report only the best one in terms of average waiting time. Please, write your answer on a single line, indicating the termination times of P1, P2, P3, P4 and P5 followed by the average

waiting time. Separate the numbers with a single space. Report the average waiting time with a single decimal digit. Do not enter any other character in the response. Example of correct answer: 20 23 11 45 67 30.5

Processo Process	TempoArrivo ArrivalTime	BurstTime	Priorità Priority
P1	0	26	1
P2	0	23	3
P3	3	15	5
P4	3	13	4
P5	6	11	2

Fasi operative. Workflow.

FCFS:
P1: from= 0 to = 25 end = 26 waitingTime = 0 tournaroundTime = 26
P2: from= 26 to = 48 end = 49 waitingTime = 26 tournaroundTime = 49
P3: from= 49 to = 63 end = 64 waitingTime = 46 tournaroundTime = 61
P4: from= 64 to = 76 end = 77 waitingTime = 61 tournaroundTime = 74
P5: from= 77 to = 87 end = 88 waitingTime = 71 tournaroundTime = 82
EndTime: 26 49 64 77 88
AverageWaitingTime: (0 + 26 + 46 + 61 + 71) / 5 = 40.80
AverageTournaroudTime: (26 + 49 + 61 + 74 + 82) / 5 = 116.80
GanttChart:
11111111111111111111111112222222222222222222222333333333333333344444444444455555555555
PS:
P1: from= 0 to = 25 end = 26 waitingTime = 0 tournaroundTime = 26
P5: from= 26 to = 36 end = 37 waitingTime = 20 tournaroundTime = 37
P2: from= 37 to = 59 end = 60 waitingTime = 37 tournaroundTime = 57
P4: from= 60 to = 72 end = 73 waitingTime = 57 tournaroundTime = 70
P3: from= 73 to = 87 end = 88 waitingTime = 70 tournaroundTime = 82
EndTime: 26 37 60 73 88
AverageWaitingTime: (0 + 20 + 37 + 57 + 70) / 5 = 36.80
AverageTournaroudTime: (26 + 37 + 57 + 70 + 82) / 5 = 108.80
GanttChart:
11111111111111111111111115555555555522222222222222222222224444444444444333333333333333

Risposta. Answer.

26 60 88 73 37 36.8

Ex 13 (5.0 points)

Italiano

Il file `/etc/passwd`, contenente informazioni essenziali per il login degli utenti in un sistema UNIX, è composto da righe con il seguente formato:

```
USERNAME:PASSWORD:UID:GID:INFO:HOME:SHELL
```

Ad esempio:

```
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
hunter:x:1000:1000:hunter:/home/hunter:/bin/bash
```

Si scriva uno script BASH che, ricevuto il nome di un utente sulla riga di comando (e.g., `hunter`), cancelli l'albero della sua home directory (campo `HOME`) e la corrispondente riga del file `/etc/passwd`. Verificare il passaggio del corretto numero di parametri.

English

The file `/etc/passwd`, which stores essential information required during login of users in a UNIX system, is composed of lines with the following format:

`USERNAME:PASSWORD:UID:GID:INFO:HOME:SHELL.`

For instance:

```
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
hunter:x:1000:1000:hunter:/home/hunter:/bin/bash
```

Write a BASH script that, given a user's name as a command line argument (e.g., `hunter`), deletes that user's home folder (HOME field) and then removes the corresponding line from the `/etc/passwd` file. Verify that the correct number of arguments are passed to the script.

Soluzione. Solution.

```
#!/bin/bash

################################################################################
# Version 1: Using grep
################################################################################

PASSWD_FILE="/etc/passwd"

# Check if the correct parameters have been passed
if [ $# -lt 1 ]; then
  echo "Usage: ./purge_user.sh <username>"
  exit 1
fi

# Get passwd line for the specified user
line=$(cat $PASSWD_FILE | grep -e "^$1:")

# Handle user name not found
if [ $? -eq 1 ]; then
      echo "Username $1 not found in $PASSWD_FILE"
      exit 1
fi

# Remove home folder of the user
HOME_FOLDER=$(echo $line | cut -d ":" -f 6)
rm -rf $HOME_FOLDER

# Remove user line from passwd
cat $PASSWD_FILE | grep -v "^$1:" > "tmp_passwd"
mv "tmp_passwd" $PASSWD_FILE

#!/bin/bash

################################################################################
# Version 2: Using a while read loop
################################################################################

PASSWD_FILE="/etc/passwd"
```

```
# Check if the correct parameters have been passed
if [ $# -lt 1 ]; then
  echo "Usage: ./purge_user2.sh <username>"
  exit 1
fi

# Scan content of the passwd file line-by-line
FOUND=0
while read line; do

      # Check user name
      USERNAME=$(echo $line | cut -d ":" -f 1)
      if [ "$USERNAME" == "$1" ]; then
      FOUND=1

      # Remove home folder of the user
      HOME_FOLDER=$(echo $line | cut -d ":" -f 6)
      rm -rf $HOME_FOLDER
      else
      echo $line >> "tmp_passwd"
      fi
done < $PASSWD_FILE

# Handle user name not found
if [ $FOUND -eq 0 ]; then
      echo "Username $1 not found in $PASSWD_FILE"
      exit 1
fi

# Update passwd file
mv "tmp_passwd" $PASSWD_FILE


#!/bin/bash

################################################################################
# Version 3: Using read -a
################################################################################

PASSWD_FILE="/etc/passwd"

# Check if the correct parameters have been passed
if [ $# -lt 1 ]; then
  echo "Usage: ./purge_user.sh <username>"
  exit 1
fi

# Scan content of the passwd file line-by-line
FOUND=0
while read -a line; do

      # Check user name
      if [ "${line[0]}" == "$1" ]; then
      FOUND=1
```

```
        # Remove home folder of the user
        rm -rf ${line[5]}
        else
        echo ${line[*]} >> "tmp_passwd"
        fi
done < $PASSWD_FILE

# Handle user name not found
if [ $FOUND -eq 0 ]; then
        echo "Username $1 not found in $PASSWD_FILE"
        exit 1
fi

# Update passwd file
mv "tmp_passwd" $PASSWD_FILE
```