### PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA

#### SISTEMAS OPERATIVOS

2da práctica (tipo a) (Segundo semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 1 h. 50 min.

Nota: La presentación, la ortografía y la gramática influirán en la calificación.

Puntaje total: 20 puntos

<u>Pregunta 1</u> (6 puntos – 30 min.) Su respuesta debe estar en la carpeta INF239\_0781\_P2\_P1\_Buzón de la Práctica 2 en PAIDEIA antes de las 09:40. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser <*su código de 8 dígitos*>\_21.txt. Por ejemplo, 20171903\_21.txt.

"bzip2 es un programa libre desarrollado bajo licencia BSD que comprime y descomprime ficheros usando los algoritmos de compresión de Burrows-Wheeler y de codificación de Huffman. El porcentaje de compresión alcanzado depende del contenido del fichero a comprimir, pero por lo general es mucho mejor al de los compresores basados en el algoritmo LZ77/LZ78 (gzip, compress, WinZip, pkzip, ...). Como contrapartida, bzip2 emplea más memoria y más tiempo en su ejecución."

```
$ ls -il /bin/{bz*,bunz*}
2621443 -rwxr-xr-x 3 root root 39144 set 5
                                             2019 /bin/bunzip2
2621443 -rwxr-xr-x 3 root root 39144 set
                                          5
                                             2019 /bin/bzcat
2621589 lrwxrwxrwx 1 root root
                                   6 set
                                          5
                                             2019
                                                              -> bzdiff
                                          5
                                             2019 /bin/bzdiff
2621542 -rwxr-xr-x 1 root root
                                2227 set
                                                  /bin/bzegrep -> bzgrep
2621593 lrwxrwxrwx 1 root root
                                          5
                                             2019
                                   6 set
                                4877 set
                                          4
                                             2019 /bin/bzexe
2621543 -rwxr-xr-x 1 root root
2621603 lrwxrwxrwx 1 root root
                                   6 set
                                          5
                                             2019
                                                  /bin/bzfgrep -> bzgrep
2621549 -rwxr-xr-x 1 root root
                                3775 set
                                             2019 /bin/bzgrep
2621443 -rwxr-xr-x 3 root root 39144 set
                                             2019
                                                  /bin/bzip2
2621556 -rwxr-xr-x 1 root root 18584 set
                                          5
                                             2019 /bin/bzip2recover
                                          5
2621605 lrwxrwxrwx 1 root root
                                   6 set
                                             2019
                                                  /bin/bzless -> bzmore
2621587 -rwxr-xr-x 1 root root
                                1297 set
                                          5
                                             2019 /bin/bzmore
```

- a) (1 punto) bzip2 comprime los archivos pero también puede descomprimirlos. ¿Cómo?
- b) (2 puntos) bunzip2 descomprime los archivos pero también puede comprimirlos. ¿Cómo?
- c) (1 punto) bzcmp compara dos archivos byte por byte, mientras que bzdiff hace lo mismo pero línea por línea. Y el programa bzcmp es el enlace simbólico al programa bzdiff. Lo mismo sucede con los programas bzegrep y bzfgrep que ambos son los enlaces simbólicos al programa bzgrep pero estos tres programas buscan de manera diferente las expresiones regulares. Entonces, ¿cómo los programas bzdiff y bzgrep saben qué les piden?
- d) (2 puntos) Realmente no existen 3 archivos ejecutables bunzip2, bzcat, bzip2, es un solo archivo ejecutable de tamaño 39144 bytes que tiene 3 nombres. Si usted en su programa está creando un proceso hijo y el proceso hijo cambia su programa por bzcat para descomprimir los archivos a la salida estándar usando la llamada al sistema execlp indicando la ruta /bin/bunzip2, ¿cuál será la línea de su código con execlp?



La práctica ha sido preparada por VK con LibreOffice Writer en Linux Mint 20.2 "Uma"

Profesor del curso: V. Khlebnikov

Lima, 24 de septiembre de 2021

## PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA

#### SISTEMAS OPERATIVOS

2da práctica (tipo a) (Segundo semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 1 h. 50 min.

Nota: La presentación, la ortografía y la gramática influirán en la calificación.

Puntaje total: 20 puntos

<u>Pregunta 2</u> (6 puntos – 30 min.) Su respuesta debe estar en la carpeta INF239\_0781\_P2\_P2\_Buzón de la Práctica 2 en PAIDEIA antes de las 10:20. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser <*su código de 8 dígitos*>\_22.txt. Por ejemplo, 20171903\_22.txt.

This code could be used in an e-commerce application that supports transfers between accounts. It takes the total amount of the transfer, sends it to the new account, and deducts the amount from the original account.

```
1  $transfer_amount = GetTransferAmount();
2  $balance = GetBalanceFromDatabase();
3
4  if ($transfer_amount < 0) { FatalError("Bad Transfer Amount"); }
5
6  $newbalance = $balance - $transfer_amount;
7  if (($balance - $transfer_amount) < 0) { FatalError("Insufficient Funds"); }
8
9  SendNewBalanceToDatabase($newbalance);
NotifyUser("Transfer of $transfer_amount succeeded.");
NotifyUser("New balance: $newbalance");</pre>
```

A race condition could occur between ... (a) where? (1 punto)

Suppose the balance is initially 100.00. An attack could be constructed as follows:

The attacker makes two simultaneous calls of the program, CALLER-1 and CALLER-2. Both callers are for the same user account.

CALLER-1 (the attacker) is associated with PROGRAM-1 (the instance that handles CALLER-1).

CALLER-2 is associated with PROGRAM-2.

CALLER-1 makes a transfer request of 80.00.

. . .

CALLER-2 makes a transfer request of 1.00.

. . .

- b) (2 puntos) ¿Cuál será el valor final del balance?
- c) (3 puntos) Presente el escenario (la secuencia de ejecución y los valores de las variables) para cada valor final posible usando la notación P1.6 (newbalance = ...) para indicar la ejecución de la línea 6 del Programa 1.



La práctica ha sido preparada por VK con LibreOffice Writer en Linux Mint 20.2 "Uma"

Profesor del curso: V. Khlebnikov

Lima, 24 de septiembre de 2021

# PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA

#### SISTEMAS OPERATIVOS

2da práctica (tipo a) (Segundo semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 1 h. 50 min.

Nota: La presentación, la ortografía y la gramática influirán en la calificación.

Puntaje total: 20 puntos

<u>Pregunta 3</u> (8 puntos – 30 min.) Su respuesta debe estar en la carpeta INF239\_0781\_P2\_P3\_Buzón de la Práctica 2 en PAIDEIA antes de las 11:00. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser <su código de 8 dígitos>\_23.txt. Por ejemplo, 20171903\_23.txt.

a) (4 puntos) Suppose processes A and B that share the queue both try to add new nodes 5 and 6 respectively to the queue at about the same time.

```
Shared queue: head -> 1 -> 2 -> 3 -> 4 -> null
```

to add a new node x to the queue:

What is the result of a race condition? Describe the way to obtain this result.

**b)** (4 puntos) (Algorithms for Mutual Exclusion by M. Raynal) Finding a software solution to the mutual exclusion problem became something of a challenge to computer scientists in the sixties, leading to the publication of many false solutions, including the following which was proposed by Hyman (1966).

The two processes P0 and P1, which compete for access to their critical sections, share the following variables:

```
var flag: array [0..1] of boolean; turn: 0..1;
```

initialized respectively to false and to some arbitrary value.

The protocol proposed for Pi is then (where i = 0 or  $1, j = i + 1 \mod 2$ ):

```
flag[i] ← true;
while turn ≠ i do
    while flag[j] do nothing enddo;
    turn ← i;
    enddo;
< critical section >;
flag[i] ← false;
```

(2 puntos) Explique el caso cuando solamente uno de 2 procesos pretende entrar en su sección crítica mientras que el otro no lo pretende. La variable *turn* puede tener el valor arbitrario.

(2 puntos) Explique el caso cuando ambos procesos, simultáneamente, pretenden entrar en sus secciones críticas.



Profesor del curso: V. Khlebnikov

La práctica ha sido preparada por VK con LibreOffice Writer en Linux Mint 20.2 "Uma"

Lima, 24 de septiembre de 2021