PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS
Examen 1
(Segundo semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 3 horas
Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**
Puntaje total: 20 puntos

Pregunta 1 (5 puntos – 30 min.)

El archivo de su respuesta debe estar en **INF239_0781_Ex1_P1_Buzón** del **Examen 1** en PAIDEIA **antes de las 08:45**. Por cada 3 minutos de retardo son -2 puntos.
El nombre de su archivo debe ser <*su_código_de_8_dígitos*>`_11.txt`. Por ejemplo, `20202912_11.txt`.

a) (2 puntos) Here are some questions for practicing unit conversions:

- How long is a nanoyear in seconds?
- Micrometers are often called microns. How long is a megamicron?
- How many bytes are there in a 1-PB memory?
- The mass of the earth is 6000 yottagrams. What is that in kilograms?

b) (3 puntos) Este es un programa de un mandato en *shell* que está disponible en Unix y en sistemas operativos de tipo Unix, en Microware OS-9, DOS (por ejemplo, 4DOS, FreeDOS), Microsoft Windows (por ejemplo, 4NT, Windows PowerShell), ReactOS. Para GNU/Linux este programa fue escrito por Mike Parker, Richard Stallman, and David MacKenzie. Para FreeDOS fue desarrollado por Jim Hall. También fue portado al sistema operativo IBM i. Aquí se presenta la versión de este programa para MINIX 3:

```
/* . . . . . .                    Author: Paul Polderman */

#include <sys/types.h>
#include <fcntl.h>
#include <signal.h>
#include <unistd.h>
#include <stdlib.h>
#include <minix/minlib.h>

#define MAXFD       18
#define CHUNK_SIZE  4096

int fd[MAXFD];

int main(int argc, char **argv);

int main(argc, argv)
int argc;
char **argv;
{
  char iflag = 0, aflag = 0;
  char buf[CHUNK_SIZE];
  int i, s, n;

  argv++;
  --argc;
  while (argc > 0 && argv[0][0] == '-') {
        switch (argv[0][1]) {
```

```
            case 'i':           /* Interrupt turned off. */
                iflag++;
                break;
            case 'a':           /* Append to outputfile(s), instead of
                                 * overwriting them. */
                aflag++;
                break;
            default:
                std_err("Usage: ..... [-i] [-a] [files].\n");
                exit(1);
        }
        argv++;
        --argc;
  }
  fd[0] = 1;                    /* Always output to stdout. */
  for (s = 1; s < MAXFD && argc > 0; --argc, argv++, s++) {
        if (aflag && (fd[s] = open(*argv, O_RDWR)) >= 0) {
                lseek(fd[s], 0L, SEEK_END);
                continue;
        } else {
                if ((fd[s] = creat(*argv, 0666)) >= 0) continue;
        }
        std_err("Cannot open output file: ");
        std_err(*argv);
        std_err("\n");
        exit(2);
  }

  if (iflag) signal(SIGINT, SIG_IGN);

  while ((n = read(0, buf, CHUNK_SIZE)) > 0) {
        for (i = 0; i < s; i++) write(fd[i], buf, n);
  }

  for (i = 0; i < s; i++)       /* Close all fd's */
        close(fd[i]);
  return(0);
}
```

¿Para que se usa el mandato que está implementado por este programa?

Preparado por VK
con LibreOffice Writer en Linux Mint 20.2 “Uma”

Profesor del curso: (0781) V. Khlebnikov

Lima, 20 de octubre de 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS
Examen 1
(Segundo semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 3 horas
Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**
Puntaje total: 20 puntos

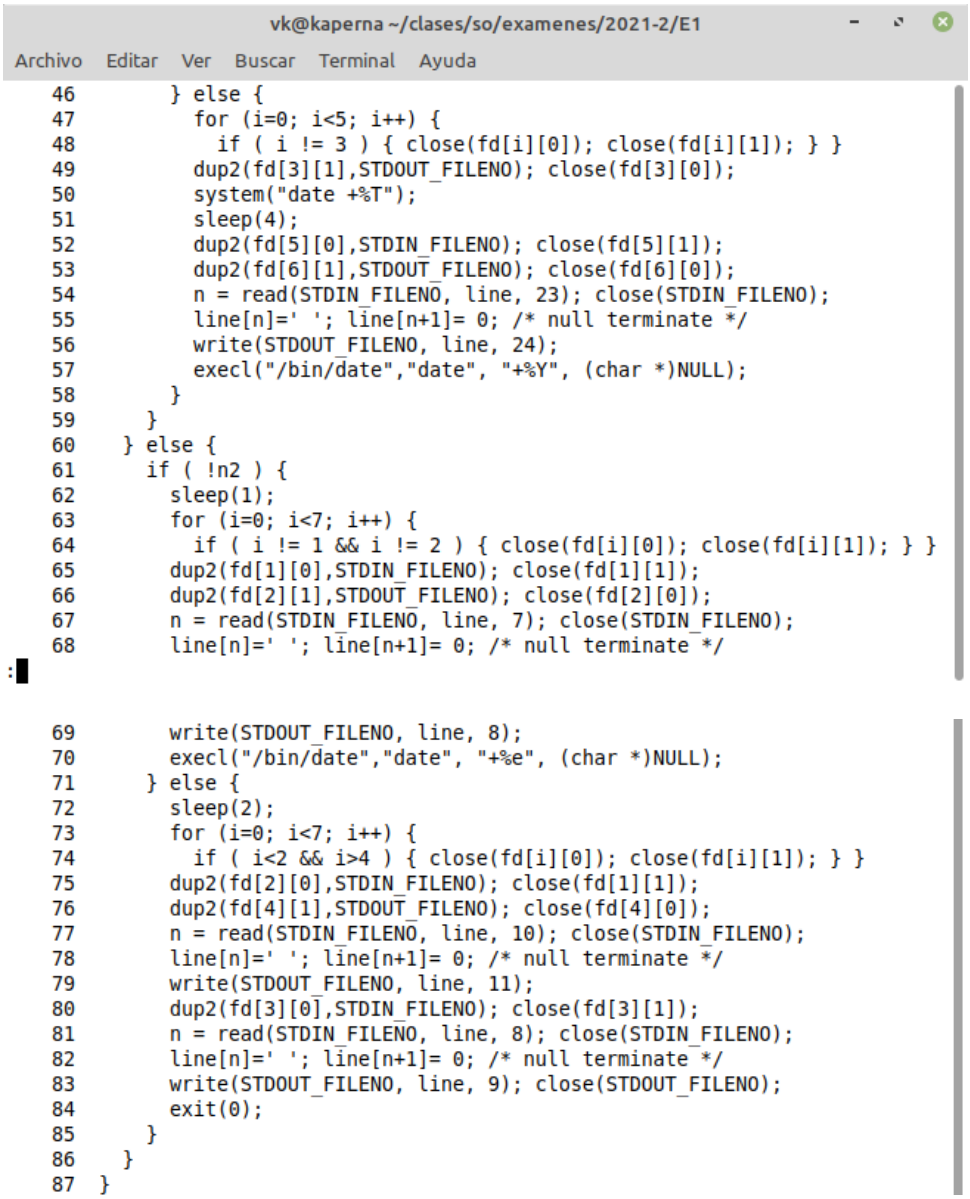
Pregunta 2 (5 puntos – 30 min.)

El archivo de su respuesta debe estar en **INF239_0781_Ex1_P2_Buzón** del **Examen 1** en PAIDEIA **antes de las 09:30**. Por cada 3 minutos de retardo son -2 puntos.
El nombre de su archivo debe ser <*su_código_de_8_dígitos*>`_12.txt`. Por ejemplo, `20202912_12.txt`.

vk@kaperna ~/clases/so/examenes/2021-2/E1

Archivo Editar Ver Buscar Terminal Ayuda

```
    1  #include <stdio.h>
    2  #include <stdlib.h>
    3  #include <unistd.h>
    4
    5  #define MAXLINE 80
    6
    7  int
    8  main(void)
    9  {
   10    int i, n;
   11    int fd[7][2];
   12    unsigned int h1, h2=0, n2=0;
   13    char line[MAXLINE];
   14
   15    for (i=0; i<7; i++) pipe(fd[i]);
   16
   17    if ( (h1=fork()) && (n2=1,h2=fork()) || (h1=fork()) ) {
   18      if ( h2 ) {
   19        for (i=1; i<6; i++) { close(fd[i][0]); close(fd[i][1]); }
   20        dup2(fd[0][1],STDOUT_FILENO); close(fd[0][0]);
   21        system("/bin/date +%a"); close(STDOUT_FILENO);
   22        waitpid(h2,(int *)NULL,0);
   23        dup2(fd[6][0],STDIN_FILENO); close(fd[6][1]);
:
   24        n = read(STDIN_FILENO, line, MAXLINE); close(STDIN_FILENO);
   25        line[n]=0; /* null terminate */
   26        fprintf(stderr, "%s", line);
   27        dup2(STDERR_FILENO,STDOUT_FILENO);
   28        execl("/bin/date","date", (char *)NULL);
   29      } else {
   30        if ( !n2 ) {
   31          for (i=2; i<7; i++) {
   32            if ( i != 4 && i != 5 ) { close(fd[i][0]); close(fd[i][1]); }
}
   33          dup2(fd[0][0],STDIN_FILENO); close(fd[0][1]);
   34          dup2(fd[1][1],STDOUT_FILENO); close(fd[1][0]);
   35          n = read(STDIN_FILENO, line, 3); close(STDIN_FILENO);
   36          line[n]=' '; line[n+1]= 0; /* null terminate */
   37          write(STDOUT_FILENO, line, 4);
   38          system("/bin/date +%b"); close(STDOUT_FILENO);
   39          sleep(3);
   40          dup2(fd[4][0],STDIN_FILENO); close(fd[4][1]);
   41          dup2(fd[5][1],STDOUT_FILENO); close(fd[5][0]);
   42          n = read(STDIN_FILENO, line, 19); close(STDIN_FILENO);
   43          line[n]=' '; line[n+1]= 0; /* null terminate */
   44          write(STDOUT_FILENO, line, 20);
   45          execl("/bin/date","date", "+%Z", (char *)NULL);
:
```



Según el manual de `date(1)`, este programa despliega el tiempo actual según el formato por defecto o indicado como su argumento: `date ... +FORMAT`, donde algunos `FORMAT` que se usan en este programa son los siguientes: `%a` (abbreviated weekday name), `%b` (abbreviated month name), `%e` (day of month), `%T` (time), `%Y` (year), `%Z` (alphabetic time zone abbreviation).

Presente el árbol de procesos creados indicando en este árbol con cuál fork fue creado cada hijo y qué valores de `h1`, `h2` y `n2` tiene cada proceso (los valores de pid se puede asignar arbitrariamente). También presente la conexión entre los procesos con las tuberías indicando solamente las tuberías que se usan por el proceso para lectura y/o escritura y la dirección de flujo en estas tuberías.

Preparado por VK
con LibreOffice Writer en Linux Mint 20.2 “Uma”

Profesor del curso: (0781) V. Khlebnikov

Lima, 20 de octubre de 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS
Examen 1
(Segundo semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 3 horas
Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**
Puntaje total: 20 puntos

Pregunta 3 (5 puntos – 30 min.)

El archivo de su respuesta debe estar en **INF239_0781_Ex1_P3_Buzón** del **Examen 1** en PAIDEIA **antes de las 10:15**. Por cada 3 minutos de retardo son -2 puntos.
El nombre de su archivo debe ser <*su_código_de_8_dígitos*>`_13.txt`. Por ejemplo, `20202912_13.txt`.

En tiempos actuales se pretende lograr la distancia social en los lugares públicos. Un local pretende permitir a entrar solo 6 personas al mismo tiempo según el siguiente algoritmo:

```
#define N 6
typedef int semaphore;
semaphore mutex = 1;
semaphore stop = 0;
int inside = 0;
int outside = 0;
bool full = false;

{
    down(&mutex);
    if (full) {
        outside++;
        up(&mutex);
        down(&stop);
        down(&mutex);    /* conseguir otra vez mutex puede ser un problema */
        outside--;
    }
    inside++;
    full = (inside == N);
    up(&mutex);

    /* inside activity */

    down(&mutex);
    inside--;
    if (inside == 0) {
        k = min(N,outside);
        for (i=0; i<k; i++)
            up(&stop);
        full = false;
    }
    up(&mutex);
}
```

¿Por qué conseguir otra vez mutex puede ser un problema? ¿Se garantiza que adentro van a estar no más de 6 personas? Explíquelo.

Preparado por VK
con LibreOffice Writer en Linux Mint 20.2 “Uma”

Profesor del curso: (0781) V. Khlebnikov

Lima, 20 de octubre de 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS
Examen 1
(Segundo semestre de 2021)

Horario 0781: prof. V. Khlebnikov

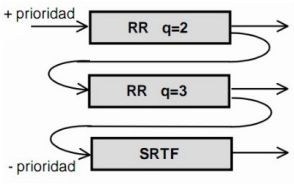
Duración: 3 horas
Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**
Puntaje total: 20 puntos

Pregunta 4 (5 puntos – 30 min.)

El archivo de su respuesta debe estar en **INF239_0781_Ex1_P4_Buzón** del **Examen 1** 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*>`_14.txt`. Por ejemplo, `20202912_14.txt`.

Se tienen los siguientes datos de un conjunto de procesos:

Proceso	Ráfaga de CPU	Tiempo de llegada
A	7	0
B	8	3
C	8	4
D	5	11
E	6	12
F	2	20



El sistema operativo utiliza un algoritmo de planificación con 3 colas multinivel con **prioridad apropiativa** con las características indicadas en el esquema de arriba.

Presente la ubicación de procesos en las colas y la secuencia temporal de ejecución de procesos en el formato

```
RR q=2:  A  BC      DE       F
RR q=3:
SRTF:
CPU:     AA
Tiempo:  0....5...10...15...20...
```

Indique los tiempos de terminación de ráfagas y los tiempos de espera de cada proceso.

Preparado por VK
con LibreOffice Writer en Linux Mint 20.2 “Uma”

Profesor del curso: (0781) V. Khlebnikov

Lima, 20 de octubre de 2021