

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

3ra práctica (tipo a)
(Segundo semestre de 2020)

Horario 0781: prof. V. Khlebnikov
 Horario 0782: prof. F. Solari A.

Duración: 1 h. 50 min.
 Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**
 Puntaje total: 20 puntos

Pregunta 1 (6 puntos – 30 min.) Su respuesta debe estar en la carpeta **Buzón 1** de la **Práctica 3** en PAIDEIA **antes de las 11:40**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_31.txt`. Por ejemplo, 20171903_31.txt.

En memoria particionada el sistema operativo ocupa siempre una zona de la memoria, mientras que el resto está disponible para los procesos que van llegando al sistema. La zona de memoria que está a disposición de los procesos se puede dividir en un número fijo o variable de particiones. Cuando el número de particiones es fijo, hablamos de un sistema de gestión de memoria con número fijo de particiones o MFT (Multiprogramación con número Fijo de Tareas) que está presentado en las diapositivas 43 y 44 de la clase 9 del nuestro curso.

En este caso la zona de memoria reservada para los procesos se divide en un número fijo de particiones; el tamaño de cada partición es invariable una vez que se ha establecido. El número de particiones y el tamaño de cada una se define en el momento de arrancar la máquina. El resultado es un sistema multiprogramado con número fijo de tareas. Las particiones pueden ser todas del mismo tamaño o de distinto tamaño.

El algoritmo de ubicación debe decidir en qué partición albergar al nuevo proceso. La ubicación de un proceso en memoria es trivial si todas las particiones tienen el mismo tamaño. Es indiferente en qué partición se ubique un proceso mientras haya particiones libres. Si no hay particiones libres, los procesos que deban cargarse en memoria tendrán que esperar a que se libere una partición.

Cuando las particiones de memoria son de distinto tamaño se pueden aplicar varios algoritmos de ubicación. Una política de ubicación es asignar la primera partición disponible en la que quepa el proceso. Otra política de ubicación consiste en asignar al proceso la partición libre más pequeña en la que quepa. En el primer caso, se utilizará una única cola. En el segundo caso, se empleará una cola para cada partición. Si existen varias particiones de igual tamaño, se asigna una cola a todas las particiones de igual tamaño.

El particionamiento de la memoria en zonas estáticas de igual tamaño plantea dos problemas: (1) se produce fragmentación interna, (2) un programa puede ser demasiado grande y no caber en ninguna de las particiones.

a) (3 puntos – 15 min.) Considérese un sistema de gestión de memoria con particiones estáticas o MFT. El sistema operativo ocupa la primera partición que tiene un tamaño de 10 K; el resto de la memoria está dividido en cuatro particiones:

Base de la partición	Tamaño de la partición
10 K	18 K
28 K	12 K
40 K	8 K
48 K	16 K

El algoritmo de ubicación asigna a cada proceso la partición más pequeña en la que quepa; por lo tanto se empleará una cola de planificación para cada partición. Los procesos llegaron juntos en el orden A, B, C, D y E. Cada proceso termina su ejecución en 3 minutos. Las cantidades de memoria que requiere cada proceso son las siguientes:

Proceso	Tamaño
A	8 K
B	14 K
C	18 K
D	6 K
E	14 K

Indique la siguiente información relacionada con cada proceso:

proceso: tiempo de inicio de ejecución, tamaño de partición, fragmentación interna.

b) (3 puntos – 15 min.) Considérese el mismo sistema que en el ejercicio anterior con la salvedad de que en este caso se emplea una única cola de procesos. Los procesos que llegan al sistema y sus características son los mismos que los del ejemplo anterior. El algoritmo de ubicación carga los procesos en la partición disponible (o libre) más pequeña en la que quepan. Se solicita la misma información relacionada con cada proceso.

Pregunta 2 (6 puntos – 30 min.) Su respuesta debe estar en la carpeta **Buzón 2** de la **Práctica 3** en PAIDEIA **antes de las 12:20**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_32.txt`. Por ejemplo, `20171903_32.txt`.

Considere el siguiente programa en C, para Linux x86_64 o amd64:

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int g_var_noi;
int g_var_i=321;
char *g_var_str="variable global string";
char *g_var_str_ptr;

int main(void) {
    int i,j=123;
    char *cad;

    cad=(char *)malloc(20);
    printf("pid %d g_var_noi \t dirección %p \t contenido %d\n",getpid(),&g_var_noi,g_var_noi);
    printf("pid %d g_var_i \t dirección %p \t contenido %d\n",getpid(),&g_var_i,g_var_i);
    printf("pid %d g_var_str \t dirección %p \t contenido %p \t cadena %s\n",\
        getpid(),&g_var_str,g_var_str,g_var_str);
    printf("pid %d g_var_str_ptr \t dirección %p \t contenido %p \t cadena %s\n",\
        getpid(),&g_var_str_ptr,g_var_str_ptr,g_var_str_ptr);
    printf("pid %d &main \t dirección %p\n",getpid(),main);
    printf("pid %d &malloc \t dirección %p\n",getpid(),malloc);
    printf("pid %d &printf \t dirección %p\n",getpid(),printf);
    printf("pid %d i \t dirección %p \t contenido %d\n",getpid(),&i,i);
    printf("pid %d j \t dirección %p \t contenido %d\n",getpid(),&j,j);
    printf("pid %d cad \t dirección %p \t contenido %p \t cadena %s\n",getpid(),&cad,cad,cad);
    getchar();
    printf("-----\n");
    return(0);
}
```

La compilación del programa:

```
$ gcc -o pa3_p2 -static pa3_p2.c # comentario: compilación estática, no librerías dinámicas
```

La ejecución del programa:

```
$ ./pa3_p2
pid 3019 g_var_noi      dirección 0x4c3300      contenido 0
pid 3019 g_var_i        dirección 0x4c00f0      contenido 321
pid 3019 g_var_str      dirección 0x4c00f8      contenido 0x495008      cadena variable global string
pid 3019 g_var_str_ptr  dirección 0x4c3308      contenido (nil)          cadena (null)
pid 3019 &main           dirección 0x401d95
pid 3019 &malloc         dirección 0x426550
pid 3019 &printf         dirección 0x410e10
pid 3019 i               dirección 0x7ffe4eab67f8 contenido 4195608
pid 3019 j               dirección 0x7ffe4eab67fc contenido 123
pid 3019 cad             dirección 0x7ffe4eab6800 contenido 0xc93b70      cadena
```

Y mientras ejecuta, como ya hemos visto, podemos revisar el mapa de memoria:

```
$ cat /proc/3019/maps
00400000-00401000 r--p 00000000 08:04 4989286 /home/fsolari/pa3/pa3_p2
00401000-00495000 r-xp 00001000 08:04 4989286 /home/fsolari/pa3/pa3_p2
00495000-004bc000 r--p 00009500 08:04 4989286 /home/fsolari/pa3/pa3_p2
004bd000-004c0000 r--p 0000bc00 08:04 4989286 /home/fsolari/pa3/pa3_p2
004c0000-004c3000 rw-p 0000bf00 08:04 4989286 /home/fsolari/pa3/pa3_p2
004c3000-004c4000 rw-p 00000000 00:00 0
00c92000-00cb5000 rw-p 00000000 00:00 0
7ffe4ea97000-7ffe4eab8000 rw-p 00000000 00:00 0 [heap]
7ffe4eafa000-7ffe4eafd000 r--p 00000000 00:00 0 [stack]
7ffe4eafd000-7ffe4eafe000 r-xp 00000000 00:00 0 [vvar]
7ffe4eafe000-7ffe4eafe000 r-xp 00000000 00:00 0 [vdso]
fffffffff600000-fffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

En el mapa de memoria, mostrado como tabla, la primera columna indica la dirección de inicio y fin (excluyente), luego los bits de permisos (rwxp), el offset en el file, el dispositivo mayor:menor del file, el i-nodo, y el filename del file.

a) (3 puntos) Ubique los elementos como variables y funciones, en el mapa de memoria, indicando que a espacio del proceso corresponde (código, datos inicializados, no inicializados, heap, stack...) sin considerar [vsyscall].

En su respuesta, puede copiar el mapa obtenido y (quitando lo no relevante) colocar lo solicitado apropiadamente.

b) (3 puntos) Si el tamaño de página es de 4 KiBytes, ¿cuántas páginas tiene cada una de las partes descritas, y cuántas páginas tiene el programa en total en memoria? Justifique sus cálculos.

Pregunta 3 (8 puntos – 30 min.) Su respuesta debe estar en la carpeta **Buzón 3** de la **Práctica 3** en PAIDEIA antes de las **13:00**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser <su_código_de_8_dígitos>_33.txt. Por ejemplo, 20171903_33.txt.

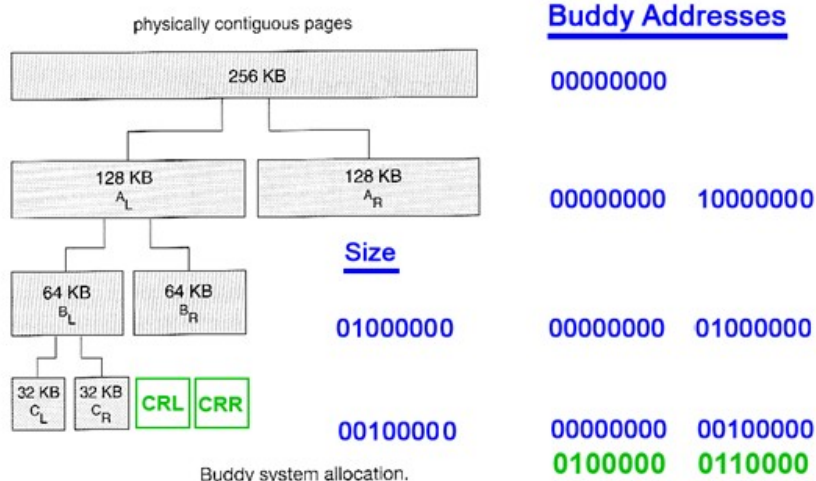
a) (4 puntos) La diapositiva 56 de la clase 9 del nuestro curso contiene la siguiente figura:

(1) ¿En qué unidades de medida se presentan las direcciones de *buddies*?

(2) Si el bloque AR se divide en sus dos *buddies*, BRL y BRR, ¿cuáles serán sus direcciones?

(3) Si el bloque BRR (creado de AR, el derecho) se divide en sus dos *buddies*, CRRL y CRRR, ¿cuáles serán sus direcciones?

(4) Si considerar los bloques **CRL** y **CRR** (presentados en la figura con el color verde) como los *buddies* del bloque BR, ¿cuáles serán sus direcciones?



b) (4 puntos) Tomando el programa y proceso de la Pregunta 2, tenemos el mapa de memoria:

```
$ cat /proc/3019/maps
00400000-00401000 r--p 00000000 08:04 4989286 /home/fsolari/pa3/pa3_p2
00401000-00495000 r-xp 00001000 08:04 4989286 /home/fsolari/pa3/pa3_p2
00495000-004bc000 r--p 00095000 08:04 4989286 /home/fsolari/pa3/pa3_p2
004bd000-004c0000 r--p 000bc000 08:04 4989286 /home/fsolari/pa3/pa3_p2
004c0000-004c3000 rw-p 000bf000 08:04 4989286 /home/fsolari/pa3/pa3_p2
004c3000-004c4000 rw-p 00000000 00:00 0
00c92000-00cb5000 rw-p 00000000 00:00 0 [heap]
7ffe4ea97000-7ffe4eab8000 rw-p 00000000 00:00 0 [stack]
7ffe4eafa000-7ffe4eafd000 r--p 00000000 00:00 0 [vvar]
7ffe4eafd000-7ffe4eafe000 r-xp 00000000 00:00 0 [vdso]
fffffffff6000000-fffffffff6010000 --xp 00000000 00:00 0 [vsyscall]
```

Ya sabemos que el tamaño de página es 4 KiBytes y hemos calculado el número de páginas, de cada espacio del proceso y el total.

Si consideramos ahora, que las tablas de páginas están divididas en 3 niveles, de igual tamaño cada nivel, y que cada entrada de tabla de páginas tiene 8 bytes, determine:

- i) ¿Cuántos bits se usan para cada nivel? **(1 punto)**
- ii) ¿Cuántas entradas tiene cada tabla de páginas? **(1 punto)**
- iii) ¿Cuántas tablas de páginas se requieren? (de todos los niveles) **(1 punto)**
- iv) ¿Qué tamaño en bytes tiene cada tabla de páginas y cuánto ocupan en total? **(1 punto)**



La práctica ha sido preparada por FS (2,3b) y VK (1,3a)
con LibreOffice Writer en Linux Mint 20 "Ulyana"

Profesores del curso: (0781) V. Khlebnikov
(0782) F. Solari A.

Lima, 27 de noviembre de 2020