

# FINAL 5/4/2021

## Ejercicio 1

---

1. Dado el siguiente código, tomado del código fuente de la aplicación **ffmpeg** (utilizada para [de]codificar/reproducir/manipular archivos multimedia), se pide:

- Indicar el tamaño en bytes de las 3 estructuras: `ogg_stream`, `ogg_state` y `ogg`.
- Escribir el código assembly x86\_64 correspondiente a la función `int ogg_find_stream (struct ogg * ogg, int32_t serial)`.
- En caso de haber *padding*, muestre claramente su tamaño y ubicación.

```
1: struct ogg_stream {
2:     uint8_t *buf;
3:     uint32_t bufsize;
4:     uint32_t bufpos;
5:     uint32_t pstart;
6:     uint32_t psize;
7:     uint32_t pflags;
8:     uint32_t pduration;
9:     uint32_t serial;
10:    uint64_t granule;
11:    uint64_t start_granule;
12:    int64_t lastpts;
13:    int64_t lastdts;
14:    int64_t sync_pos;
15:    int64_t page_pos;
16:    int32_t flags;
17:    const struct ogg_codec *codec;
18:    int32_t header;
19:    int32_t nsecs, segp;
20:    uint8_t segments[255];
21:    int32_t incomplete;
22:    int32_t page_end;
23:    int32_t keyframe_seek;
24:    int32_t got_start;
25:    int32_t got_data;
26:    int32_t nb_header;
27:    int32_t start_trimming;
28:    int32_t end_trimming;
29:    uint8_t *new_metadata;
30:    int32_t new_metadata_size;
31:    void *private;
32: };
33:
34: struct ogg_state {
35:     uint64_t pos;
36:     int32_t curidx;
37:     struct ogg_state *next;
38:     int32_t nstreams;
39:     struct ogg_stream streams[1];
40: };
41:
42: struct ogg {
43:     struct ogg_stream *streams;
44:     int32_t nstreams;
45:     int32_t headers;
46:     int32_t curidx;
47:     int64_t page_pos;
48:     struct ogg_state *state;
49: };
50:
51: int ogg_find_stream (struct ogg * ogg, int32_t serial)
52: {
53:     for (int i = 0; i < ogg->nstreams; i++)
54:         if (ogg->streams[i].serial == serial)
55:             return i;
56:     return -1;
57: }
```

## Ejercicio 2

---

1. ¿Qué se entiende por ejecución fuera de orden (de las instrucciones)?
2. ¿Qué desventaja presenta la emisión múltiple estática en comparación con su contraparte dinámica? ¿Qué ventaja presenta? ¿Por qué se dice estática?
3. Responda si son verdaderas o falsas las siguientes afirmaciones y **justifique** cada respuesta (las respuestas sin justificar serán consideradas incorrectas):
  - a. La optimización por bloqueo (*blocking*) puede ser utilizada en cualquier algoritmo.
  - b. En assembly x86\_64, al ingresar a un procedimiento, es necesario `push %rbp` al *stack*.
  - c. Cada función que se ejecuta en la computadora tiene su propio *stack*.
  - d. El tamaño de la tabla de paginación no puede ser mayor al tamaño de una página virtual.
  - e. La representación de enteros en complemento a dos es balanceada, es decir, tanto el máximo representable como el mínimo representable tienen el mismo valor absoluto.
  - f. Según el estándar IEEE-754 de punto flotante sólo hay una forma de redondeo: al par más cercano (o *round-to-even*).
  - g. Siempre que se usa paginación virtual es necesaria la TLB.
  - h. Al igual que sucede con las tablas de paginación, hay una cache por cada programa que se ejecuta en el procesador.
  - i. La relación entre cache *hits* y *misses* es una propiedad del diseño del procesador.
  - j. Al segmentar el *datapath* de un procesador se mejora tanto el *throughout* de instrucciones, como la latencia de las mismas y la frecuencia de operación del procesador.
  - k. Si la frecuencia de operación de una arquitectura secuencial es 400 MHz y la misma se divide en 5 etapas, la frecuencia de operación resultante es 2.00 GHz.

l. Considerando un sistema que implementa memoria virtual, cuando el conjunto de trabajo (*working set*) de un único programa es mayor que la memoria instalada, no se ve afectado el desempeño del sistema, esto sólo ocurre cuando todos los procesos tienen conjuntos de trabajo mayores a la memoria cache.

m. Una burbuja es la única forma de evitar riesgos en arquitecturas segmentadas.

n. Dada una representación de números similar a la estándar IEEE-754, con 3 bits de exponente y 4 de fracción, el número 5.6250 se puede representar en forma exacta.

o. Todo número representable en el formato del estándar IEEE-754 se puede representar también en una representación similar con una mayor cantidad de bits en su parte fraccionaria.

p. Todo número representable en el formato del estándar IEEE-754 se puede representar también en una representación similar con una mayor cantidad de bits en su exponente.

q. En caso de tener que elegir entre poner una memoria cache SRAM, o utilizar un sistema de paginación virtual, es preferible optar por la memoria cache SRAM.