

Correcciones

Consola

- En el leerArchivo (char* ruta) de la consola, cuando se hace el malloc del buffer es:
buffer = malloc(largo + 1);
memset(buffer, '\0', (largo + 1));

Swap

- Descomentar en el main del SWAP el: socket_umc = recibirConexion(socket_servidor);
- Comentar CASOS DE PRUEBA en el main
- En el inicializar se deberia usar un escribirArchivoBinarioEnPag(int numPagina, char* buffer), porque actualmente el escribirArchivoBinario(char* programa), con el "a" del fopen siempre agrega al final. Para probar el programa deje el escribirArchivoBinario y cambie el fopen a un "r+" (Pero porque sabia que corria un solo proceso, supongo que cuando corramos mas de uno se van a pisar)
- Falta enviar "OK", en el inicializar al umc despues de escribir el archivo binario, en caso de que haya paginas disponibles
- En el leerPagina() del swap.c es:
t_proceso* proceso = list_get(paginasDelProceso, i);
dentro del if(i == num_pagina). Osea, antes hacia i - 1.
- En el recibir del socketComms.c, despues del:
void* recibido = malloc(sizeof(int));
hay que hacer:
memset(recibido, '\0', sizeof(int)); // inicializarla antes de usarla
- En el leerArchivoBinarioEnPag hay que hacer:
char* buffer = malloc(pagina_size + 1); //Osea agregarle el + 1
memset(buffer, '\0', (pagina_size + 1)); //Inicializarlo. Va a tirar un implicit declaration no se porque, pero no importa.
- En el leerPagina de swap.c hay que hacer:
enviar(socket_umc, paginaLeida, (pagina_size + 1)); /Agregarle el + 1
free(paginaLeida);
Tambien hay que sacar el log_info que hace el else de abajo, sino lo hace cada vez que busca la pagina, que no significa que no exista, solo que la esta buscando

UMC

- En el cambio_proceso_activo de umc.c hace un flush sin que se compruebe (adentro o afuera de la funcion) si la TLB esta habilitada. Ademas Valgrind marca que se hace uso de un uninialized value dentro del flush cuando hace el if(tlb[i].idp == idp). El idp de parametro le llega bien asique me parece que el tlb[i].idp no se estaria inicializando bien, pero la verdad no le di importancia porque el programa funcionaba, pero es raro. (Cuando corri el facil.ansisop lo hice con la tlb habilitada con 2 o 3 entradas no me acuerdo :S). Igual ya te digo, no modifique nada.
- Valgrind dice lo mismo del uninizialided value en obtener_marco(int idp, int num_pagina), cuando hace un aumentar_uso_tlb(idp, num_pagina); Ni idea porque es.
- En el leer_pagina(int num_pagina, int offset, size_t t, void*cpu), despues del:
copiar_pagina_en_memoria(idp, num_pagina, contenido_pagina);
hay que hacer:
marco = obtener_marco(idp, num_pagina);

porque sino esa variable marco cuando se le pasa al leer_posicion_memoria queda con un -1 (si la pagina no estaba en MP)

Nucleo

- En el threadPlanificador(), donde hace el manejo de la cola ready, el:

```
elemento = queue_pop(colaPCBReady);
```

va antes del:

```
if(elemento → estado == EXEC){
```

Actualmente el queue_pop lo hace dentro del if y cuando compara no sabe que es elemento-estado

- En el interfazCPUConsola.c, en el serializarCargarPCB el:

```
serializarPCB(pcb, aux);
```

estaba serializando mal (no entendi muy bien, creo que porque se perdía la referencia a aux en manejo interno del serializarPCB, osea aux terminaba apuntando a cualquier cosa).

Lo que hice fue comentar el serializarPCB(pcb, aux) y cambie como lo llama. Masomenos quedo asi:

```
//serializarPCB(pcb, aux); (Comente lo que estaba)
```

```
void* puntero = NULL;
```

```
int tamano_pcb_serializado = serializarPCB(pcb, &puntero); //lo paso con &  
memcpy(aux, puntero, tamano_pcb_serializado); //Copio lo que se guardo en  
puntero a la porcion de memoria de aux
```

```
free(puntero);
```

La verdad no le di mas bola porque cuando lo cambie asi funciono.

- En el nucleo.c en el ThreadReceptroYEscuchaConsola hay que sacar el:

```
consola = malloc(sizeof(t_consola));
```

```
consola → socketConsola = conexionRecibida;
```

```
list_add(consolas, consola);
```

Porque sino, se crea y se agrega la consola 2 veces a la lista de consolas, ya que despues se llama a nuevoPrograma(), que hace exactamente lo mismo.

CPU

- Crear un pcb con todos los campos creados y asignarlo al pcb_actual, al inicializar, sino cuando llega el primer cargarPCB(), cuando hace pcb_destroy(pcb_actual) rompe todo.
- En la primitiva definirVariable hay que considerar si la lista del stack tiene size 0. En ese caso agregarlo.