

Entrega 3

Sistemas Operativos II

Mellino, Natalia

Farizano, Juan Ignacio

-
1. Se prefiere emular una CPU porque de esta forma se simplifica mucho el desarrollo de sistemas operativos al reducir el ciclo compilar-ejecutar-depurar y permite el uso de depuradores ya disponibles.
 2. La memoria se define como una constante en el archivo `/machine/mmu.hh` como `MEMORY_SIZE = NUM_PHYS_PAGES x PAGE_SIZE`. Como tenemos 32 páginas y cada una tiene un tamaño de 128 bytes el tamaño de la memoria es $32 \times 128 = 4096$ bytes.
 3. Modificaría la cantidad de páginas. El tamaño de cada página se deja fijo en 128 bytes para que coincida con el tamaño del sector del disco y proveernos así más simplicidad.
 4. El disco cuenta con una única superficie dividida en pistas y a su vez, cada pista está dividida en sectores. Por defecto, el disco de la máquina de Nachos cuenta con 32 pistas y 32 sectores por cada pista. Luego el tamaño de cada sector, como fue mencionado en el punto anterior, es de 128 bytes. Como resultado contamos con $32 \times 32 \times 128 = 131.072$ bytes.
 5. En el archivo `/machine/encoding.hh` podemos encontrar todas las instrucciones enumeradas, las instrucciones `OP_UNIMP` y `OP_RES` no son simuladas, por lo tanto contamos con un total de 61 instrucciones MIPS que simula la máquina virtual.
 6. Utilizando el comando `grep` podemos ver que la función `main` está definida en los siguientes archivos:

- `/bin/fuse/nachosfuse.c`
- `/bin/out.c`
- `/bin/coff2noff.c`
- `/bin/disasm.c`
- `/bin/readnoff.c`
- `/bin/main.c`
- `/bin/coff2flat.c`
- `/userland/echo.c`
- `/userland/filetest.c`
- `/userland/tiny_shell.c`
- `/userland/matmult.c`
- `/userland/touch.c`
- `/userland/halt.c`
- `/userland/sort.c`
- `/userland/shell.c`
- `/threads/main.c`

Al correr `make`, en el archivo `Makefile.depends`, podemos ver que la función `main` se encuentra en el archivo `/threads/main.cc`

7. • `Initialize` en `/threads/system.cc`
 - `ASSERT` en `/lib/assert.hh`

- ParseDebugOpts en /threads/system.cc
- RandomInit en /machine/system_dep.cc
- SetFlags en /lib/debug.cc
- SetOpts en /lib/debug.cc
- TimerInterruptHandler en /threads/system.cc
- SetStatus en /threads/threads.cc
- Enable en /machine/interrupt.cc
- CallOnUserAbort en /machine/system_dep.cc
- Cleanup en /threads/system.cc
- SetUp en /threads/preemptive.cc
- SetExceptionHandler en /userprog/exception.cc
- DEBUG en /lib/utility.hh
 - Print en /lib/debug.cc
- SysInfo en /threads/sys_info.cc
- PrintVersion en /threads/main.cc
- ThreadTest en /threads/thread_test.cc
 - DEBUG en /lib/utily.hh
 - Choose en /threads/thread_test.cc
 - Run en /threads/thread_test.cc
- Halt en /machine/interrupt.cc
 - Print en /machine/statistics.cc
 - Cleanup en /threads/system.cc
- ASSERT en /lib/assert.hh
 - Assert en /lib/assert.cc
- StartProcess en /userprog/prog_test.cc
 - ASSERT en /lib/assert.hh
 - Open en /filesys/file_system.cc
 - InitRegisters en /userprog/address_space.cc
 - RestoreState en /userprog/address_space.cc
 - Run en /userprog/debugger_command_manager.cc
- ConsoleTest en /userprog/prog_test.cc
- – P en /threads/semaphore.cc
 - GetChar en /machine/console.cc
 - PutChar en /machine/console.cc
- Copy en /filesys/fs_test.cc
 - ASSERT en /lib/assert.hh
 - DEBUG en /lib/utility.hh
 - Create en /filesys/file_system.cc
 - Open en /filesys/file_system.cc
 - Write en /filesys/open_file.cc
- Print en /threads/scheduler.cc
 - Apply en /lib/list.hh
 - ThreadPrint en /threads/scheduler.cc
- Remove en /lib/list.hh
- List en /filesys/directory.cc
- Check en /filesys/file_system.cc

- DEBUG en /lib/utility.hh
 - Mark en /lib/bitmap.cc
 - GetRaw en /filesys/file_header.cc
 - FetchFrom en /filesys/file_header.cc
 - CheckForError en /filesys/file_system.cc
 - CheckFileHeader en /filesys/file_system.cc
 - CheckBitmaps en /filesys/file_system.cc
 - PerformanceTest en /filesys/fs_test.cc
 - Print en /filesys/fs_test.cc
 - FileWrite en /filesys/fs_test.cc
 - FileRead en /filesys/fs_test.cc
 - Remove en /lib/list.hh
 - Delay en /machine/system_dep.cc
 - MailTest en /network/net_test.cc
 - Send en /network/post.cc
 - Receive en /network/post.cc
 - Halt en /machine/interrupt.cc
 - Finish en /threads/thread.cc
 - SetLevel en /machine/interrupt.cc
 - ASSERT en /lib/utility.hh
 - DEBUG en /lib/utility.hh
 - GetName en /threads/thread.cc
 - Sleep en /threads/thread.cc
8.
 - ASSERT: aborta el program en caso de que una condición no se cumpla.
 - DEBUG: muestra por pantalla mensajes de depuración brindando al usuario flags de depuración y permitiendole definir las suyas propias.
 9.
 - + - encender todas las banderas de depuración
 - t - mensajes sobre sistemas de threads
 - s - mensajes sobre semaforos, locks y condiciones
 - i - mensajes acerca de emulación de interrupciones
 - m - mensajes sobre emulación de la máquina
 - d - mensajes sobre emulación del disco
 - f - mensajes sobre el sistema de archivos
 - a - mensajes sobre el espacio de direcciones
 - e - mensajes sobre manejo de excepciones
 - n - mensajes sobre emulación de red
 10.
 - USER_PROGRAM - definida en los makefile de userprog, vmem, network y filesystem
 - FILESYS_NEEDED - definida en los makefile de filesystem, network, userprog y vmem
 - FILESYS_STUB - definida en userprog y vmem
 - NETWORK - definida en network
 11.
 - Opciones generales
 - -d – hace que ciertos mensajes de depuración se muestren
 - -do – habilita opciones que modifican el comportamiento cuando se imprimen mensajes de depuración
 - -p – habilita multitarea preemptiva para hilos del kernel

- **-rs** – hace que ocurra 'yield' en lugares aleatorios
- **-z** – imprime información sobre la versión y copyright y sale
- Opciones de hilos
 - **-tt** – prueba el subsistema de threads, se le pide al usuario elegir un test a correr de entre una colección de tests disponibles
- Opciones de **USER_PROGRAM**
 - **-s** – causa que el programa se ejecute en modo de un solo paso
 - **-x** – corre un programa de usuario
 - **-tc** – prueba la consola
- Opciones del sistema de archivos
 - **-f** – causa que el disco físico sea formateado
 - **-cp** – copia un archivo de UNIX a nachos
 - **-pr** – imprime un archivo de nachos por la salida standard
 - **rm** – elimina un archivo nachos del sistema de archivos
 - **ls** – lista los contenidos del directorio de nachos
 - **-D** – imprime contenidos del sistema de archivos entero
 - **-c** – verifica la integridad del sistema de archivos
 - **-tf** – testea la performance del sistema de archivos de nachos
- Opciones de red
 - **-n** – setea la confiabilidad de la red
 - **-id** – setea el host id de esta maquina (necesario para la red)
 - **-tn** – corre un simple test del software de red de nachos

13. La primera es una lista enlazada simple donde el acceso de los hilos no tiene restricciones, en cambio en la clase `SynchList`, si la lista está vacía y un hilo quiere remover un elemento de ella, deberá esperar a que contenga un elemento. Además, sólo un thread a la vez puede acceder a las estructuras de datos de la lista.
17. De forma análoga al problema de la entrega 1, se soluciona realizando el yield recién luego de incrementar el contador.