

## Práctica Mach i GNU Hurd

### RESPUESTAS:

**5. Un cop arreglat el problema de la pregunta anterior, comproveu que el programa memorymanagement.c funciona correctament. Aquest programa usa host\_processors i vm\_map de forma intercalada, per demanar memòria 8 cops. L'ús de processor\_info per demanar memòria queda fora del seu ús habitual, però funciona correctament. Responen:**

**5.1. Quanta memòria assigna al procés cada crida a host\_processors?**

Cada llamada de host\_processors, asigna 0x1000.

**5.2. Quanta memòria assigna al procés cada crida a vm\_map?**

Cada llamada de vm\_map, asigna 0x2000.

**5.3. Quines adreces ens dóna el sistema en cada crida (host\_processors i vm\_map)?**

Para host\_processors: 0x1033000, 0x12e5000, 0x12e8000, 0x12eb000 y para vm\_map: 0x12e3000, 0x12e6000, 0x12e9000, 0x12ec000.

**5.4. Són pàgines consecutives? (pista: us ajudarà, incrementar el número d'iteracions que fa el programa... per veure la seqüència d'adreces més clara)**

Se puede observar que son consecutivas y se van entrelazando, menos la primera que no sabemos porque lo hace diferente. Hemos aumentado a 32 iteraciones para verlo un poco mejor.

**5.5. Quines proteccions podem demanar a l'assignar memòria a un procés Mach? (pista: veieu el fitxer <mach/vm\_prot.h)**

Protecciones:

```
#define VM_PROT_NONE    ((vm_prot_t) 0x00)
#define VM_PROT_READ    ((vm_prot_t) 0x01)    /* read permission */
#define VM_PROT_WRITE   ((vm_prot_t) 0x02)    /* write permission */
#define VM_PROT_EXECUTE ((vm_prot_t) 0x04)
#define VM_PROT_DEFAULT (VM_PROT_READ|VM_PROT_WRITE)
#define VM_PROT_ALL (VM_PROT_READ|VM_PROT_WRITE|VM_PROT_EXECUTE)
#define VM_PROT_NO_CHANGE ((vm_prot_t) 0x08)
#define VM_PROT_COPY    ((vm_prot_t) 0x10)
```

**5.6. Canvieu el programa per a que la memòria demanada sigui de només lectura. Quin error us dona el sistema quan executeu aquesta nova versió del programa?**

Nos sale un Segmentation Fault. Para ponerlo como solo lectura, hemos modificado la función `vm_map`, poniendo "VM\_PROT\_READ".

**5.7. Després, afegiu una crida a `vm_protect` (...) per tal de desprotegir la memòria per escriptura i que el programa torni a permetre les escriptures en la memòria assignada. Proveu la nova versió i comproveu que ara torna a funcionar correctament.**

Para solucionarlo con la función `vm_protect()`, hemos añadido lo siguiente al código:

```
res = vm_protect (mach_task_self(), (vm_address_t) p, 8192, 0, VM_PROT_DEFAULT);
```

**8. Feu un programa que creï un flux (`thread_create`) i li canviï l'estat (`uesp`, `eip`) amb les crides `thread_get_state` i `thread_set_state`, per engegar-lo posteriorment (`thread_resume`).**

Código en `ex8.c`. No aparece ningún error de compilación, pero al ejecutar nos sale un Segmentation Fault. Creemos que se debe a que no el hijo no tiene permisos para ejecutar el `printf`.

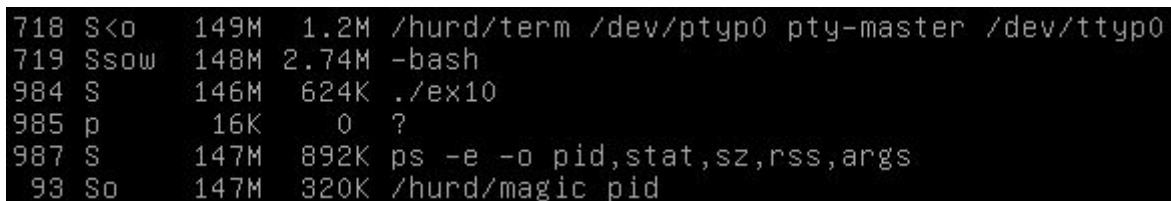
**9. Observar que en el fitxer tenim dues definicions de funcions interessants per resoldre el problema de la pregunta anterior: ... però cap de les dues soluciona el problema...**

La función `mach_setup_thread()` le da al thread una pila y la configura para que se ejecute con el PC cuando se reanude.

Mientras que, `mach_setup_tls()` le da al thread una zona TLS (thread-local storage)

**10. Feu un programa que creï una task (`task_create` / `task_terminate`), i li doni memòria (`vm_allocate`), per després copiar-li una pàgina de dades (`vm_write`).**

Código en `ex10.c`.



```
718 S<0 149M 1.2M /hurd/term /dev/ptyp0 pty-master /dev/ttyp0
719 Ssow 148M 2.74M -bash
984 S 146M 624K ./ex10
985 p 16K 0 ?
987 S 147M 892K ps -e -o pid,stat,sz,rss,args
93 So 147M 320K /hurd/magic pid
```

Imagen del `ps` mientras se ejecuta `ex10`

**12. [opcional] Feu un programa que creï un procés amb `fork()` i faci que pare i fill es comuniquin amb un missatge de Mach, usant `mach_msg_send()` i `mach_msg_receive()`.**

Código en `ex12.c`. Se han creado dos procesos: El hijo envia el proceso y el padre lo recibe. No se ha conseguido que se comuniquen y el proceso padre no llega a recibir el mensaje. Se ha probado hardcodeando el port en ambos pero no se ha tenido éxito.