

Un novato en un proyecto legacy

Historia del proyecto Hurd SMP

Almudena García

EsLibre 2022

26/06/2022

Hurd SMP

- Hurd SMP busca desarrollar soporte multiprocesador en el sistema operativo GNU/Hurd
 - Para ello, se modifica el código de GNU Mach, añadiendo las rutinas para encontrar, enumerar, arrancar y configurar los procesadores; para finalmente añadirlos al sistema
- El proyecto empezó su desarrollo en Junio de 2018, y sigue en la actualidad
- **En esta charla contaremos su historia y anécdotas**

Comienzos

- El proyecto surge de forma bastante anodina
 - En un inicio, ni siquiera pensaba en desarrollar el SMP: lo creíamos implementado
 - En 2015, haciendo pruebas, encontramos que Hurd solo hace uso de un procesador
 - Los desarrolladores nos dicen “no está implementado”
 - ¡¡Pero en el FAQ no dicen eso!!

GNU Hurd/ [faq/](#) Does GNU/Hurd support SMP/Multicore?

[Recent Changes](#)

[Preferences](#)

This page:

[Edit](#)

[History](#)

[Source](#)

[?Discussion](#)

The Hurd servers themselves are multithreaded, so they should be able to take benefit of the parallelism brought by SMP/Multicore boxes. This has however never been tested yet because of the following.

[Mach](#) used to be running on SMP boxes like the [Intel iPSC/860](#), so principally has the required infrastructure. It has however not yet been enhanced to support nowadays' SMP standards like ACPI, etc. Also, [GNU Mach's](#) Linux device driver glue code likely isn't SMP-safe. As this glue code layer is not used in the [Xen](#) port of GNU Mach, the plan is to try it in this environment first.

That is why for now GNU/Hurd will only use one logical processor (i.e. one core or one thread, depending on the socket type).

Once this issue is solved, there are follow-up issues about [multiprocessing](#) and [multithreading](#).

[Project idea](#).

FAQ de GNU Hurd hablando sobre el soporte multiprocesador
<https://www.gnu.org/software/hurd/faq/smp.html>

Consultando sobre el SMP

- El FAQ nos indica que hay un cierto soporte multiprocesador, aunque obsoleto
- Se encuentra desactivado debido a que hay un código proveniente de Linux que no es paralelizable
- **Entonces, será simplemente cuestión de probarlo. Pero... ¿cómo se activa?**

Preguntando a los desarrolladores

- Escribimos en la lista de correo, a ver que nos dicen...

Enable SMP support

From: Almudena Garcia

Subject: Enable SMP support

Date: Thu, 7 Jun 2018 23:47:55 +0200

Hi all:

Reading this post in Hurd FAQ, I read that Mach has SMP support, but It was disabled because the Linux device drivers glue code isn't thread-safe.

<https://www.gnu.org/software/hurd/faq/smp.html>

Then, I ask . Are there any form to enable this SMP support in GNU Mach? (At my own risk).

I would like to test it.

P.D.: It's only a curiosity, not an urgency

Pregunta inicial en la lista de correo “bug-hurd”

<https://lists.gnu.org/archive/html/bug-hurd/2018-06/msg00048.html>

Preguntando a los desarrolladores

Re: Enable SMP support

From: James Clarke

Subject: Re: Enable SMP support

Date: Fri, 8 Jun 2018 20:27:19 +0100

From gnumach's configfrag.ac:

```
> # Multiprocessor support is still broken.  
> AH_TEMPLATE([MULTIPROCESSOR], [set things up for a uniprocessor])  
> mach_ncpus=1  
> AC_DEFINE_UNQUOTED([NCPUS], [$mach_ncpus], [number of CPUs])  
> [if [ $mach_ncpus -gt 1 ]; then  
>   AC_DEFINE([MULTIPROCESSOR], [1], [set things up for a multiprocessor])  
> [fi]
```

So enabling it is just a case of tweaking that constant or making it a configurable option.

James

Nos indican que hay que modificar un fichero de configuración

Primeros descubrimientos

- La línea *mach_ncpus*, en *configfrag.ac*, controla el número de procesadores a utilizar
 - Poniéndola a un valor superior a 1, Mach usará SMP
- **Ese fichero de configuración ¿dónde está?**
 - ¡¡En el código fuente de gnumach!!
- **¡¡El número de procesadores se indica en tiempo de compilación!! :-O**

Primeras pruebas

- En fin, vayamos a probarlo...
- Hay que descargar el código fuente de gnumach, buscar ese fichero, y poner *mach_ncpus = 2*
- Ponemos el código a compilar...
 - ¡¡Y falla la compilación!!
- <https://www.youtube.com/watch?v=8CbmTAQNmXc>

Preguntando

- Supongo que será fallo mío... **vamos a preguntar**

Re: Enable SMP support

From: Almudena Garcia

Subject: Re: Enable SMP support

Date: Thu, 14 Jun 2018 17:58:42 +0200

Ok, I just found the file in the gnumach sources directory. Now I'm trying to compile mach, using this guide

<https://www.gnu.org/software/hurd/microkernel/mach/gnumach/building.html>

But, during the compilation process with dpkg-buildpackage, It shows an error, and stop.

You can see the error in this log

<https://pastebin.com/wZnXQ61y>

Feels I syntax error, but I don't sure

Algunos errores al escribir en inglés...

Preguntando...

Re: Enable SMP support

From: Samuel Thibault

Subject: Re: Enable SMP support

Date: Thu, 14 Jun 2018 18:36:03 +0200

User-agent: NeoMutt/20170113 (1.7.2)

Almudena Garcia, le jeu. 14 juin 2018 17:58:42 +0200, a ecrit:

> *You can see the error in this log*

>

> [2]<https://pastebin.com/wZnX061y>

>

> *Feels I syntax error, but I don't sure*

Look at the very first warning, the rest can be just a consequence of it. It seems to be saying that it's simply missing the `cpu_number()` implementation.

Samuel

Hay una función sin implementar

Preguntando...

Re: Enable SMP support

From: Almudena Garcia

Subject: Re: Enable SMP support

Date: Thu, 14 Jun 2018 20:21:05 +0200

Ok, I found the warning what you refers

```
../kern/thread.h:401:42: warning: implicit declaration of function 'cpu_number'; did you mean 'cpu_idle'? [-Wimplicit-function-declaration]
```

After check the cpu_number.h file, feels that It don't define the cpu_number function in multi-processor mode

<https://pastebin.com/rnpVCTXp>

But I don't found the source file to this library, to check the implementation.

Mas sorpresas

```
#ifndef _KERN_CPU_NUMBER_H_
#define _KERN_CPU_NUMBER_H_

/*
 *   Definitions for cpu identification in multi-processors.
 */

int    master_cpu;    /* 'master' processor - keeps time */

#if    (NCPUS == 1)
    /* cpu number is always 0 on a single processor system */
#define cpu_number()    (0)
#define CPU_L1_SIZE (1 << CPU_L1_SHIFT)

#endif /* NCPUS == 1 */
#endif /* _KERN_CPU_NUMBER_H_ */
```

No sabía que se pudieran reemplazar funciones completas por valores fijos...

Aplicando parches

- Con ayuda de varias personas, vamos corrigiendo errores de código
 - No solo falta `cpu_number()`, hay muchos mas errores
 - <http://git.savannah.gnu.org/cgiit/hurd/gnumach.git/log/?ofs=300>
- *“Supongo que será cuestión de corregir estos errores y probarlo”*
 - Mis conocimientos de C y de sistemas se quedan cortos (lo cual sería habitual a lo largo del proyecto)

Aplicando parches

git index : hurd/gnumach.git master switch

summary refs **log** tree commit diff log msg search

Age	Commit message (Expand)	Author	Files	Lines
2018-06-27	Add missing include	Samuel Thibault	1	-0/+1
2018-06-27	Fix pmap name	Samuel Thibault	1	-2/+2
2018-06-27	Add splvm prototype	Samuel Thibault	1	-0/+1
2018-06-27	Add cause_ast_check prototype	Samuel Thibault	1	-0/+5
2018-06-27	Define CPU_L1_SIZE even when NCPUS != 1	Samuel Thibault	1	-1/+2
2018-06-27	Disable position-independant compilation	Samuel Thibault	1	-0/+4
2018-06-17	Add missing prototype	Samuel Thibault	1	-0/+1
2018-06-17	Re-fix typo	Samuel Thibault	1	-1/+1
2018-06-17	Fix typo	Samuel Thibault	1	-1/+1
2018-06-13	Keep DMA enabled on QEMU disks	Samuel Thibault	1	-0/+8
2018-04-22	vm_map: Fix bugs on huge masks parameters	Samuel Thibault	2	-3/+5
2018-03-03	Avoid old type	Samuel Thibault	1	-4/+2
2018-01-28	Add const MIG types	Samuel Thibault	2	-0/+2
2018-01-28	Fix warning	Samuel Thibault	1	-1/+1
2018-01-28	Fix warning	Samuel Thibault	1	-1/+1
2018-01-28	Fix warning	Samuel Thibault	1	-1/+1
2018-01-28	Fix warning	Samuel Thibault	1	-1/+1
2017-11-12	Fix halt message	Samuel Thibault	1	-1/+2
2017-11-01	linux block: fix outbound access to non-directmap user data	Samuel Thibault	1	-4/+6
2017-10-31	linux-block: Fix calling vm_map_copy_discard on uninitialized variable	Samuel Thibault	1	-2/+1
2017-10-27	Fix commit 10ebf9565f69760e46a8f271e22c5367d54e10ff.	Justus Winter	1	-13/+0
2017-10-26	kern: Fix new task notifications.	Justus Winter	1	-1/+3
2017-10-26	kern: Fix crash.	Justus Winter	3	-2/+32
2017-10-26	linux: Fix warnings.	Justus Winter	1	-2/+1

*Muchos, muchos errores por corregir...
Algunos bastante divertidos ;)*

Probando el microkernel en modo SMP

- Tras 3 semanas de trabajo, por fín logramos compilar el código
- ¡¡Vamos a probarlo!!
- <https://www.youtube.com/watch?v=78JYeXMNeVM>

Probando el microkernel en modo SMP

```
needed.  
lance.c:v1.14 2/3/1998 dplatt@3do.com, becker@cesdis.gsfc.nasa.gov  
Partition check (DOS partitions):  
Linux PCMCIA Card Services 3.2.8  
  kernel build: 2.0.36  
  options: [pci]  
PCI routing table version 1.0 at 0xff2a0  
Intel ISA/PCI/CardBus PCIC probe:  
  no bridges found.  
ds: no socket drivers loaded!  
  
xirc2ps_cs.c 1.31 1998/12/09 19:32:55 (dd9jn+kvh)  
kernel page fault at 802811ed:  
Dump of i386_saved_state c1226034:  
EAX 00000000 EBX 00010000 ECX 80280201 EDX 178bfbfb  
ESI c114c230 EDI c114c230 EBP 00000000 ESP c1021792  
CS 0008 SS 9c70 DS 0010 ES 0010 FS 0000 GS 0000  
v86:      DS 0000 ES 1300 FS 071c GS 071c  
EIP c1040656 EFLAGS 00010013  
trapno 14: Page fault, error 00000000  
panic (cpu0) ../i386/i386/trap.c:225: kernel_trap: kernel thread accessed user s  
pace!  
  
-
```

Kernel panic. ¿cuál será el error?

Revisando el código

- El soporte multiprocesador está incompleto
 - + **Colaborador: No hay ningún código que inicialice los procesadores.**
 - **Yo: ¿a qué se refiere con “inicializar los procesadores”? ¿no están simplemente allí?**
- *“Hay soporte multihilo, pero no multiprocesador. Hay soporte por software, pero no por hardware”*
 - No termino de entender a qué se refiere con *“soporte por hardware”*... ¿cómo se arregla eso?

Evaluando la situación

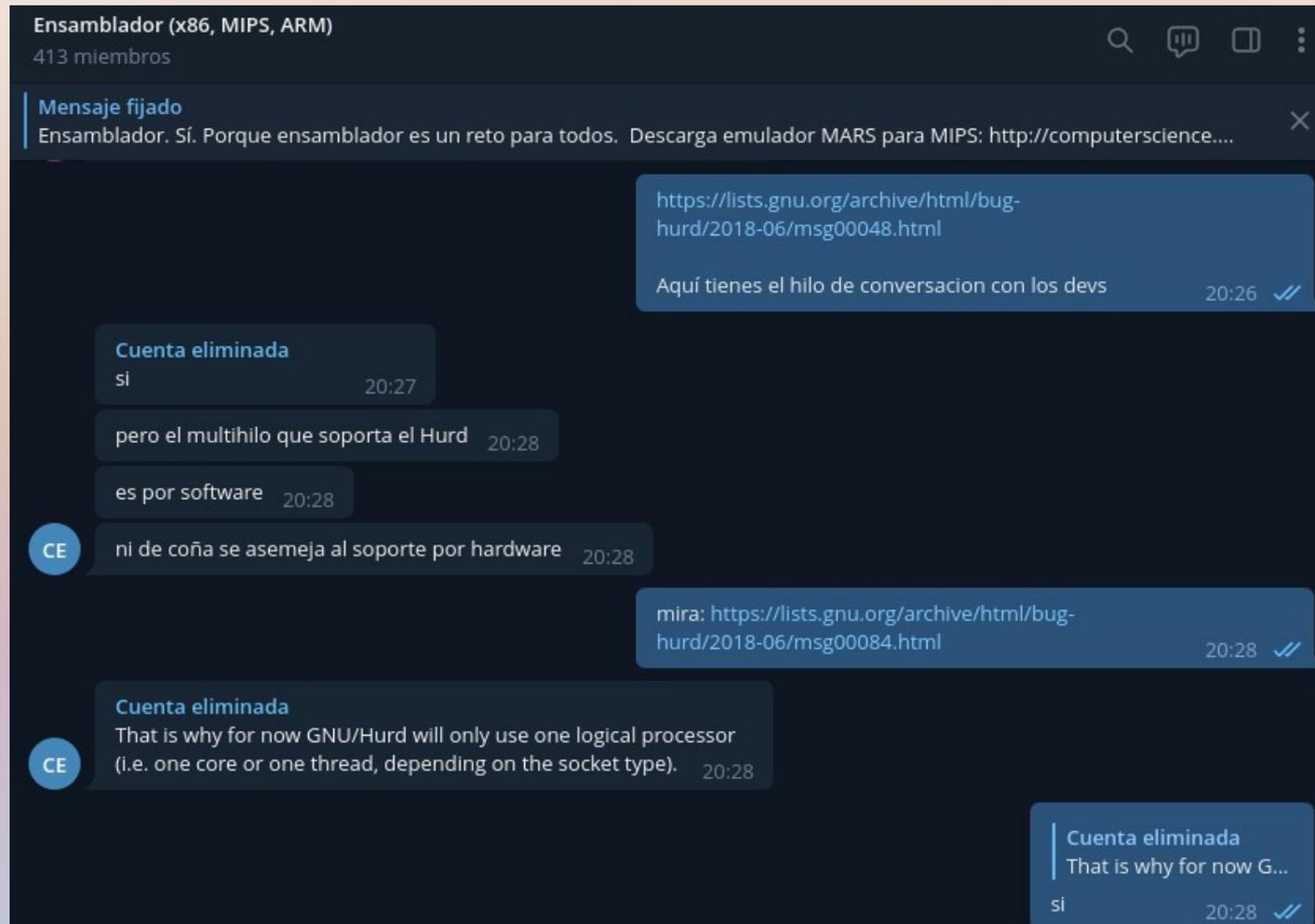
The screenshot shows a chat interface for a group named "Ensamblador (x86, MIPS, ARM)" with 413 members. At the top, there is a pinned message: "Mensaje fijado: Ensamblador. Sí. Porque ensamblador es un reto para todos. Descarga emulador MARS para MIPS: <http://computerscience...>".

The chat history includes the following messages:

- A greyed-out message from a deleted account: "Cuenta eliminada no, a ver" (20:30).
- A greyed-out message from a deleted account: "soporte SMP tiene por defecto" (20:30).
- A greyed-out message from a deleted account: "pero es soporte SW" (20:30).
- A message from user "CE": "no tiene nada que ver con hardware" (20:30).
- A message from user "si": "si" (20:30) with two checkmarks.
- A greyed-out message from a deleted account: "Cuenta eliminada y leyendo esto" (20:30).
- A message from user "si": "si vas siguiendo el hilo, ves las indicaciones y comentarios que me van dando" (20:31) with two checkmarks.
- A greyed-out message from a deleted account: "Cuenta eliminada ya estoy convencido de que ni siquiera han implementado la funcionalidad para dar soporte por Hardware" (20:31).
- A greyed-out message from a deleted account: "de hecho, estoy empezando a ver que esta ha sido toda la fuente de errores" (20:31).
- A message from user "CE": "es que no es facil" (20:31).

¿a qué se refiere con “dar soporte por hardware”?

Evaluando la situación



Multihilo pero por software...

Evaluando la situación

Ensamblador (x86, MIPS, ARM)
413 miembros

Mensaje fijado
Ensamblador. Sí. Porque ensamblador es un reto para todos. Descarga emulador MARS para MIPS: [http://computerscience....](http://computerscience...)

Cuenta eliminada
es que al driver 17:00

no debería de accederse desde un hilo 17:00

dependiendo del hardware que sea, vas a tener que cambiar con total seguridad los anillos de acceso al modo privilegiado 17:01

para posteriormente, acceder a la memoria mapeada del dispositivo 17:01

y eso, no se debería hacer desde un hilo 17:01

eso es contexto de kernel 17:01

imaginate si tienes que acceder a la memoria mapeada de un dispositivo PCI 17:02

como lo hagas desde un hilo, te puedes reír si luego otro dispositivo 17:02

hace la misma petición al mismo dispositivo 17:02

eso se podría hacer única y exclusivamente desde hilos, pero con un buen sistema de interbloqueo 17:03

y mach 17:03

CE no lo tiene 100% seguro 17:03

Me faltan conocimientos para entender esto

Evaluando la situación

Ensamblador (x86, MIPS, ARM)
413 miembros

Mensaje fijado
Ensamblador. Sí. Porque ensamblador es un reto para todos. Descarga emulador MARS para MIPS: <http://computerscience....>

Cuenta eliminada
no 17:30

cambiar lo del SMP 17:31

es decir, que Mach sea configurable para funcionar en modo SMP mediante parametro 17:31

o opcion de compilacion 17:31

y que sea el propio Mach quien detecte el numero de procesadores o cores que tiene la CPU 17:31

CE y sea capaz de configurarlos 17:31

se me ocurre que, ya que tenemos el mach_cpus, usar eso como flag 17:32 ✓

que, en vez de definir el numero de CPUs a fuego en compilación, eso sea un true/false editado 17:32 ✓

no se si me explico 17:32 ✓

CE Cuenta eliminada si 17:32

Vamos entendiendo mejor... Y se unen nuevos contribuidores ;)

Preguntando de nuevo a los desarrolladores

- Entiendo que falta algo para “inicializar los procesadores”
 - Pero ¿en qué consiste esa inicialización?
- Preguntamos de nuevo a los desarrolladores
 - Aprovechamos una consulta de un entusiasta
 - <https://lists.gnu.org/archive/html/bug-hurd/2018-08/msg00048.html>

Preguntando de nuevo a los desarrolladores

Re: How do I learn the Hurd stuff?

From: Almudena Garcia

Subject: Re: How do I learn the Hurd stuff?

Date: Sun, 26 Aug 2018 20:32:14 +0200

Hi all:

I'm reading about you're interested in contributing with Hurd and practice with C.

I have an interesting project, about implement SMP support in Hurd. Currently, Mach has a little support to multiprocessing, but this is old and very buggy.

A friend has said me that It's possible to implement SMP support via hardware, instead use software techniques, and I would try this project.

If any person can tutorize me and contribute, I will be grateful.

Preguntando de nuevo a los devs

Re: How do I learn the Hurd stuff?

From: Joshua Branson

Subject: Re: How do I learn the Hurd stuff?

Date: Wed, 29 Aug 2018 17:08:07 -0400

User-agent: Gnus/5.13 (Gnus v5.13) Emacs/26.1 (gnu/linux)

Almudena Garcia <liberamensol0000@gmail.com> writes:

> Hi all:
>
> I'm reading about you're interested in contributing with Hurd and practice
> with C.
>
> I have an interesting project, about implement SMP support in Hurd.
> Currently, Mach has a little support to multiprocessing, but this is old and
> very buggy.
>
> A friend has said me that It's possible to implement SMP support via
> hardware, instead use software techniques, and I would try this project.
>
> If any person can tutorize me and contribute, I will be grateful.
>

I can probably encourage you for this project, but I'm not sure if I'd be able to help very well. I'm still on the struggle bus with the Hurd development. I have a ever growing Hurd cheatsheet, but I am no CS major.

Also how does one implement SMP via hardware and NOT software? Is this a portable approach to do SMP? Will this approach allow SMP on Intel, AMD, ARM, POWER, and RISC chips?

Honestly, I could probably just spend some time trying to update the Hurd wiki. A lot of places just need some documentation updates. Like a lot. hahaha.

Preguntando de nuevo a los devs

Re: How do I learn the Hurd stuff?

From: Richard Braun

Subject: Re: How do I learn the Hurd stuff?

Date: Thu, 30 Aug 2018 17:01:16 +0200

User-agent: Mutt/1.5.23 (2014-03-12)

On Thu, Aug 30, 2018 at 04:52:28PM +0200, Almudena Garcia wrote:

> > *But this solution would be specific to Intel chips? ie: this method of*

> > *SMP wouldn't work on AMD or POWER right*

>

> *All x86 architecture, I think*

What on Earth are you talking about ?

--

Richard Braun

Respuestas sorprendentes, XD

Intentando extraer información

Re: How do I learn the Hurd stuff?

From: Richard Braun

Subject: Re: How do I learn the Hurd stuff?

Date: Thu, 30 Aug 2018 17:35:54 +0200

User-agent: Mutt/1.5.23 (2014-03-12)

On Thu, Aug 30, 2018 at 05:32:40PM +0200, Almudena Garcia wrote:

> Excuse me, It's a little offtopic.

>

> I was talking about implement SMP via hardware in Hurd (Mach really). In
> this implementation, Mach configures the processor during the boot,
> enabling multicore support. (All Pentium 4 or modern x86 processor supports
> this)

>

> In the current SMP support, this multicore support feels been build from
> software, generating the threads and assigning It to the processor from
> software. Then, my idea is to start a project to change this SMP software
> to a newer implementation via hardware, as I previously said. As this
> form, we don't need to know how many cores has the processor and set the
> core number in Mach in compilation time. Instead, the same processor will
> detect the cores number and configure SMP automatically.

>

> https://c9x.me/x86/html/file_module_x86_id_45.html

>

> <https://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-software-developer-vol-3a-part-1-manual.html>

What is "a newer implementation via hardware" ? Can you be very specific
about what you're referring to ?

--

Richard Braun

Intentando extraer información

Re: How do I learn the Hurd stuff?

From: Richard Braun

Subject: Re: How do I learn the Hurd stuff?

Date: Thu, 30 Aug 2018 18:24:34 +0200

User-agent: Mutt/1.5.23 (2014-03-12)

On Thu, Aug 30, 2018 at 06:15:28PM +0200, Almudena Garcia wrote:
> *It's not only a test. Obviously, It must start as a test, but I want to add*
> *this to Hurd if runs.*
>
>

That's not what I meant. Please improve your English.

You're heavily misusing words here. You're not talking about a "hardware SMP implementation", just asking the system how many processors it has. That's a very tiny detail among all that's required to write a decent scalable SMP kernel.

--
Richard Braun

Vale, pero ¿cómo se hace lo que dices?

Intentando extraer información

Re: How do I learn the Hurd stuff?

From: Samuel Thibault

Subject: Re: How do I learn the Hurd stuff?

Date: Thu, 30 Aug 2018 19:01:16 +0200

User-agent: NeoMutt/20170113 (1.7.2)

Hello,

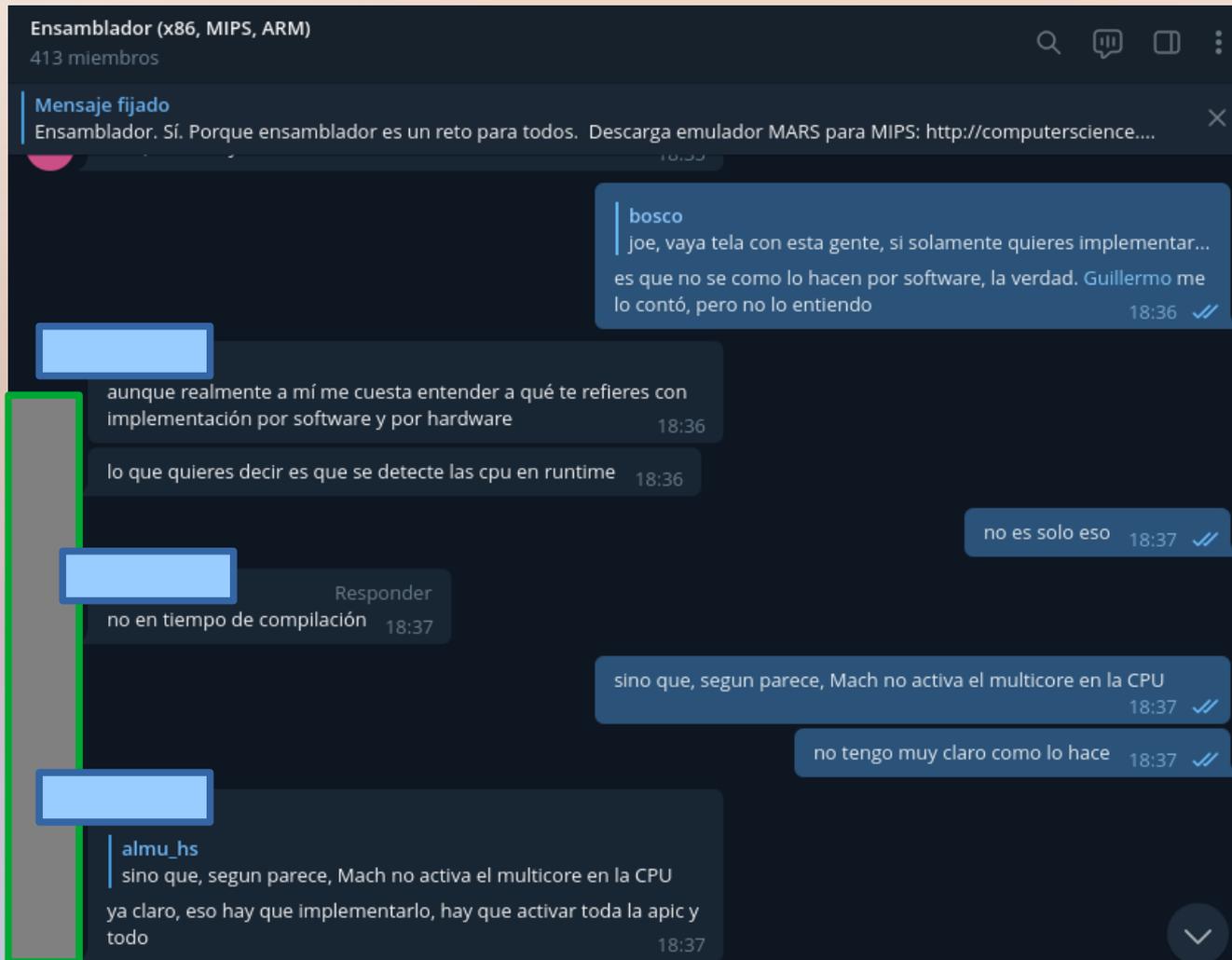
Almudena Garcia, le jeu. 30 août 2018 18:58:19 +0200, a écrit:
> *Instead, It feels to use a strange implementation to do*
> *multithreading (not real multiprocessor) via software.*

It's not "strange", it's what all OSes do. Even with multiple processors, OSes do that, to schedule threads on the available processors via software.

Samuel

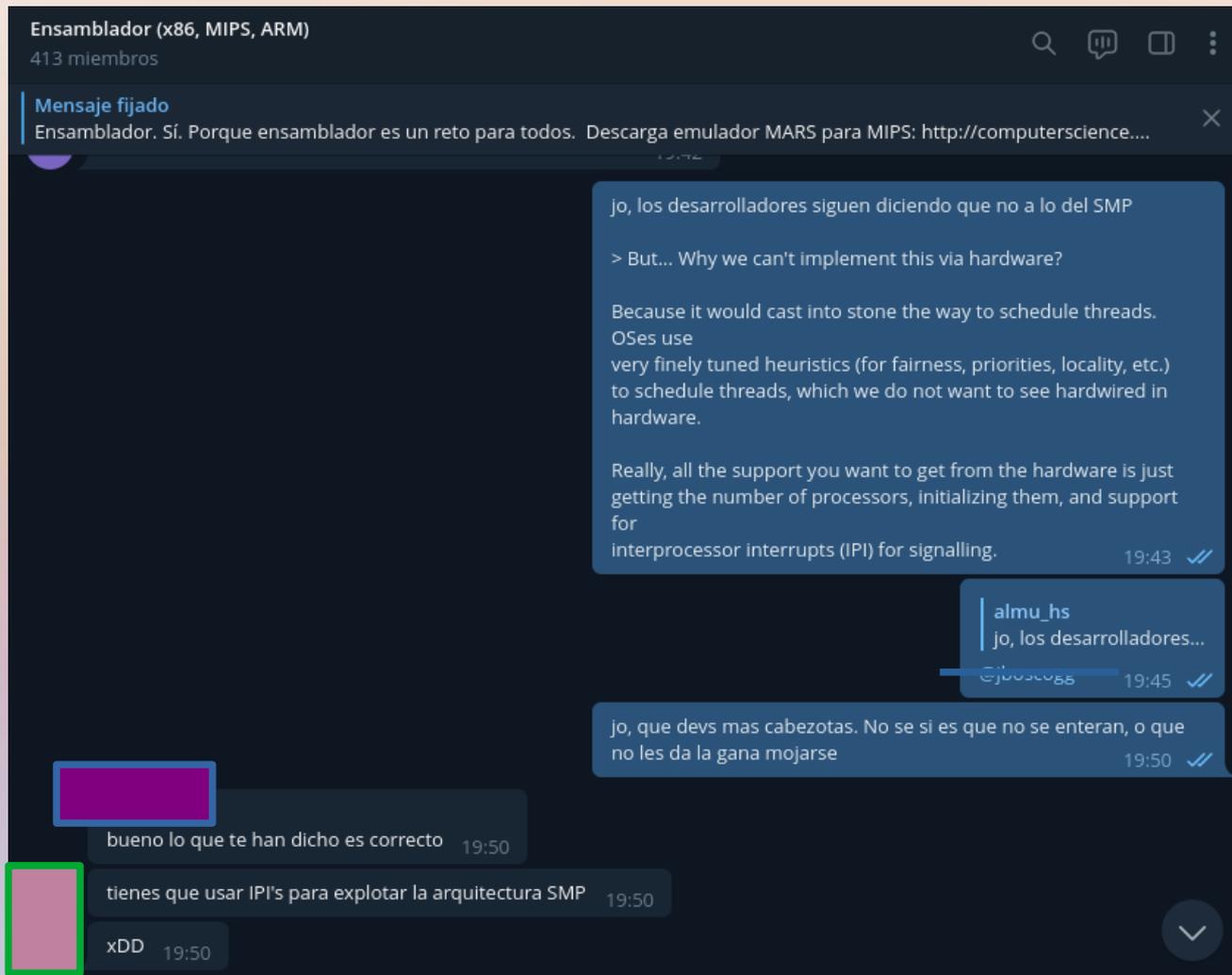
Entonces ¿qué es lo que falta?

Volviendo a consultar en Telegram



Se van uniendo nuevos colaboradores

Volviendo a consultar en Telegram



Gracias por avisar ;)

Respuesta correcta

Re: How do I learn the Hurd stuff?

From: Samuel Thibault

Subject: Re: How do I learn the Hurd stuff?

Date: Thu, 30 Aug 2018 19:42:35 +0200

User-agent: NeoMutt/20170113 (1.7.2)

Almudena Garcia, le jeu. 30 août 2018 19:35:23 +0200, a écrit:
> *But... Why we can't implement this via hardware?*

Because it would cast into stone the way to schedule threads. OSes use very finely tuned heuristics (for fairness, priorities, locality, etc.) to schedule threads, which we do not want to see hardwired in hardware.

Really, all the support you want to get from the hardware is just getting the number of processors, initializing them, and support for interprocessor interrupts (IPI) for signalling.

Samuel

*Por fín tenemos una idea de lo que hay que hacer.
Pero... ¿qué es una IPI?*

Evaluando la situación

- Situación: no hay soporte multiprocesador
 - El sistema arranca con un solo procesador
- Hay que enumerarlos, activarlos, configurarlos y meterlos al kernel
 - Para ello, tenemos que usar el controlador APIC y las IPI

Redactando borrador del proyecto

- Con ayuda de un nuevo colaborador, enumeramos las tareas y redactamos el borrador
 - Ya de paso, voy resolviendo dudas sobre tecnicismos que van surgiendo ;)
- <https://gitlab.com/-/snippets/1756024>

Modelo de pruebas: MinSMP

- Nuestro colaborador elabora un pequeño sistema llamado MinSMP
 - Dedicado específicamente a enumerar y arrancar los procesadores
 - Por fín entiendo cómo se implementan ciertas cosas...
 - Aunque no me gusta el estilo de código, lo mejoro y seguimos trabajando en paralelo

MinSMP: enumerando CPUs

```
almu@debian: ~/MinSMP-005
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
MinSMP
ACPI:
  rsdp = 000f5880; rsdp->rsdt_addr = 1ffe15c9
  rsdt = 1ffe15c9; rsdt->length = 00000030 (n = 00000003)
    00000000: FACP (1ffe149d)
    00000001: APIC (1ffe1511)
    00000002: HPET (1ffe1591)
LAPIC
  lapic_addr = fee00000
CPU:
  cpu 00000000: apic_id = 00000000
  cpu 00000001: apic_id = 00000001
IOAPIC:
  ioapic: apic_id= 00000000 addr = fec00000, base = 00000000
```

MinSMP: Ideas locas

```
typedef struct icr_low{
    uint8 vector:8;
    icr_type type:3;
    icr_dest_mode dest_mode:1;
    icr_deliv_status deliv_status:1;
    uint8 reserved1:1;
    icr_level level:1;
    icr_trig_mode trigger_mode:1;
    uint8 reserved2:2;
    icr_dest_sh dest_shorthand:2;
    uint16 reserved3:12;
} icrl;

typedef struct icr_high{
    uint32 reserved:24;
    uint8 dest:8;
} icrh;

typedef struct ApicICRHReg
{
    icrh r; /* the actual register */
    unsigned p[3]; /* pad to the next 128-bit boundary */
} ApicICRHReg;

typedef struct ApicICRLReg
{
    icrl r; /* the actual register */
    unsigned p[3]; /* pad to the next 128-bit boundary */
} ApicICRLReg;
```

```
typedef enum level_enum{
    De_assert = 0b0,
    Assert = 0b1
} icr_level;

typedef enum trig_mode_enum{
    Edge = 0b0,
    Level = 0b1
} icr_trig_mode;

typedef enum dest_sh_enum{
    NoShortHand = 0b00,
    Self = 0b01,
    AllIncludingSelf = 0b10,
    AllExcludingSelf = 0b11
} icr_dest_sh;
```

Bitfield sobre enums ¿qué podría fallar?

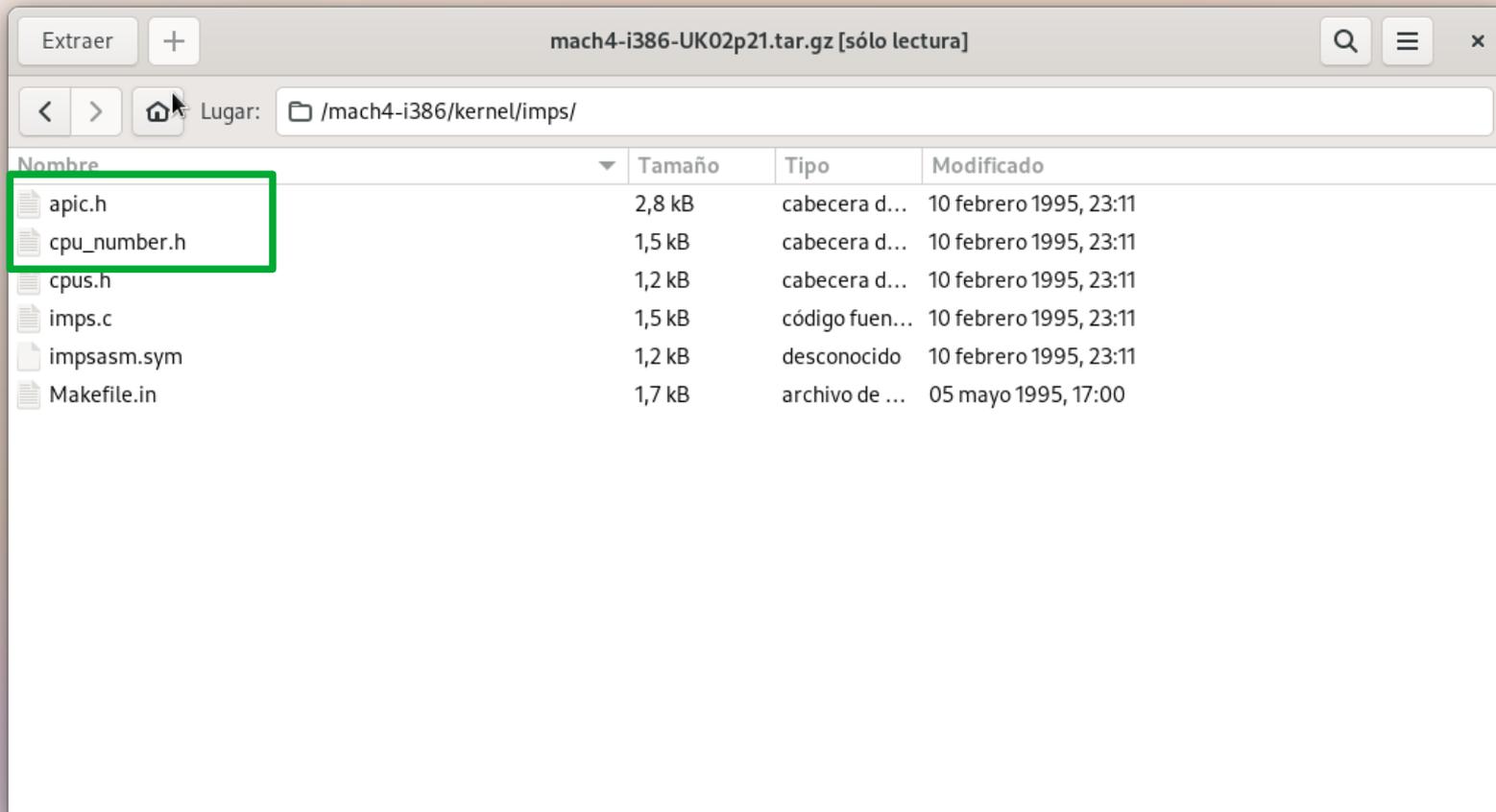
Recopilando documentación

- **Gran parte de la documentación sobre Hurd y gnumach viene de palabra de los propios devs**
 - Algunas incluso de conversaciones de IRC!!
- **La documentación de Intel sobre SMP en x86 es demasiado densa y poco clara**
 - Debemos acudir a foros y cursos online para ir aprendiendo
 - En muchas ocasiones tenemos el “qué” pero no el “cómo”
 - Conocemos el algoritmo de la StartUp IPI pero...
¿cómo se lanza una IPI?

Haciendo arqueología informática

- Buscando por internet, recuperamos el código fuente original de Mach 4
 - ¡¡Incluye los ficheros del multiprocesador!!
 - Habían sido eliminados de gnumach hace años
 - Recuperamos algunos de esos ficheros para nuestro trabajo
 - De momento, los incorporamos a MinSMP para probarlos

Revisando código fuente de Mach 4



Imps... ¿Intel MultiProcessor Specification?

Revisando código de Mach 4

```
apic.h
~/cache/fr-qhRMx5/mach4-i386/kernel/imps

Abrir + Guardar ☰ x

22 */
23 #ifndef _IMPS_APIC_
24 #define _IMPS_APIC_
25
26 typedef struct ApicReg
27 {
28     unsigned r; /* the actual register */
29     unsigned p[3]; /* pad to the next 128-bit boundary */
30 } ApicReg;
31
32 typedef struct ApicIoUnit
33 {
34     ApicReg select;
35     ApicReg window;
36 } ApicIoUnit;
37 #define APIC_IO_UNIT_ID      0x00
38 #define APIC_IO_VERSION     0x01
39 #define APIC_IO_REDIR_LOW(int_pin) (0x10+(int_pin)*2)
40 #define APIC_IO_REDIR_HIGH(int_pin) (0x11+(int_pin)*2)
41
42 typedef struct ApicLocalUnit
43 {
44     ApicReg reserved0;
45     ApicReg reserved1;
46     ApicReg unit_id;
47     ApicReg version;
48     ApicReg reserved4;
49     ApicReg reserved5;
50     ApicReg reserved6;
51     ApicReg reserved7;
52     ApicReg task_pri;
53     ApicReg reservedb;
54     ApicReg reservedc;
55     ApicReg eoi;
56     ApicReg remote;
57     ApicReg logical_dest;
58     ApicReg dest_format;
```

Cabecera C/ObjC ▾ Anchura del tabulador: 4 ▾ Ln 1, Col 1 ▾ INS

¡¡Una cabecera con las estructuras de APIC!!

Revisando código de Mach 4

```
Abrir [+] cpu_number.h Guardar [≡] x
~/cache/fr-A2oTFq/mach4-i386/kernel/imps
14 * THE UNIVERSITY OF UTAH AND CSL ALLOW FREE USE OF THIS SOFTWARE IN ITS "AS
15 * IS" CONDITION. THE UNIVERSITY OF UTAH AND CSL DISCLAIM ANY LIABILITY OF
16 * ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.
17 *
18 * CSL requests users of this software to return to csl-dist@cs.utah.edu any
19 * improvements that they make and grant CSL redistribution rights.
20 *
21 * Author: Bryan Ford, University of Utah CSL
22 */
23 #ifndef _IMPS_CPU_NUMBER_
24 #define _IMPS_CPU_NUMBER_
25
26
27 #ifndef ASSEMBLER
28
29 #include "apic.h"
30
31 static inline int
32 cpu_number()
33 {
34     return apic_local_unit.unit_id.r >> 24;
35 }
36
37 #else ASSEMBLER
38
39 #include "impsasm.h"
40
41 #define CPU_NUMBER(reg) \
42     movzbl APIC_LOCAL_VA+APIC_LOCAL_UNIT_ID+3,reg
43
44 #endif ASSEMBLER
45
46
47 #include "i386/cpu_number.h"
48
49
50 #endif _IMPS_CPU_NUMBER_

Cabecera C/ObjC Anchura del tabulador: 4 Ln 1, Col 1 INS
```

¡¡Una implementación de `cpu_number()`!! Pero incompleta

Revisando código de Mach 4

```
Abrir [icon] imps.c ~/.cache/fr-2zTDRk/mach4-i386/kernel/imps Guardar [icon] x
15 * IS" CONDITION. THE UNIVERSITY OF UTAH AND CSL DISCLAIM ANY LIABILITY OF
16 * ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.
17 *
18 * CSL requests users of this software to return to csl-dist@cs.utah.edu any
19 * improvements that they make and grant CSL redistribution rights.
20 *
21 * Author: Bryan Ford, University of Utah CSL
22 */
23
24 #include <mach/kern_return.h>
25
26 void
27 interrupt_processor(int which_cpu)
28 {
29     panic("interrupt_processor");
30 }
31
32 void
33 start_other_cpus()
34 {
35     printf("start other CPUs please!!!\n");
36 }
37
38 kern_return_t
39 cpu_control(int cpu, int *info, int count)
40 {
41     printf("cpu_control %d\n", cpu);
42     return KERN_FAILURE;
43 }
44
45 kern_return_t
46 cpu_start(int cpu)
47 {
48     printf("cpu_start %d\n", cpu);
49     return KERN_FAILURE;
50 }
51
```

C Anchura del tabulador: 4 Ln 1, Col 1 INS

Estas funciones me suenan... pero están sin implementar

Comienzan las prisas

- Quedan 2 semanas para las evaluaciones del Concurso Universitario de Software Libre
 - ¡¡Y aún sigo trabajando sobre el MinSMP!!
 - Al CUSL se presenta Hurd SMP, no el MinSMP
- Hay que empezar la migración a gnumach lo antes posible
 - Problemas: el entorno de gnumach es muchísimo mas complicado que el del MinSMP
 - Queda MUY poco tiempo...

Portando de MinSMP a gnumach

- Empezamos portando la enumeración de los procesadores
 - **Problema: Hay que recorrer las tablas ACPI, que están enlazadas mediante direcciones físicas**
 - En gnumach, la conversión a dirección lógica no es tan sencilla como en el MinSMP
 - En paginación, hay que registrar cada dirección física en la tabla de páginas, lo cual es tedioso
 - **¡¡Pongámoslo antes de la paginación!!**
 - En modo segmentación, solo hay que sumar la dirección base. Nos ahorramos tener que mapear
 - **Problema: ¡¡La dirección de la Local APIC está en una posición mas alta que la dirección base!!**
 - Bueno, nos apuntamos la dirección física y ya la mapearemos cuando tengamos paginación...

Portando de MinSMP a gnumach

- Continuamos con el arranque de los procesadores
 - El envío de la StartUp IPI y la rutina ensamblador para pasar a modo protegido da algunos problemas
 - + *Yo: Parece que la rutina para pasar a modo protegido se ejecuta, pero algo está provocando un kernel panic*
 - *Colaborador: ¿Has cargado la GDT de gnumach?*
 - + *Yo: ¿qué es una GDT?*
 - **Spoiler: la GDT es la tabla de segmentos... los procesadores arrancan en segmentación**
 - **¡¡La segmentación y paginación están implementados en hardware!!**
 - **Primera vez que me entero de eso**

Continuando el desarrollo

- MinSMP solo activaba los procesadores, pero en gnumach hay que hacer mas cosas
- **Hay que configurar la paginación, pero ¿¿cómo se hace eso??**
 - “*It’s just register initialization!!*”
 - <https://pastebin.com/j4eSnGp2>
 - Cargamos la tabla de páginas del kernel en los registros indicados, pero algo falla
 - **Esto demoraría tiempo en arreglarse... varios meses**
 - ¡¡Hay que hacer un mapeado temporal!!
 - Por supuesto, esta información la obtenemos de palabra de los propios devs, después de un par de meses de bloqueo
 - <https://pastebin.com/bjDzNwyV>

Consultando sobre la paginación

<AlmuHS> hi

<AlmuHS> **how can I initialize the pagging in a new processor? In gnumach**

<AlmuHS> configure the virtual memory in the processor, exactly

<youpi> **you can just use the same page table as the main processor**

<youpi> i.e. like the set_cr3 lines in i386at_init()

<youpi> (and cr4 for enabling PAE)

<AlmuHS> but, how can I assign this page table to the new processor?

<youpi> I don't know how that works, but that is probably documented along the rest of how to boot additional processors

<youpi> **it's just register initialization**

<AlmuHS> i found pmap_init() function, but It feels to be pagging general initialization

<youpi> yes, you don't need to delve into it

<youpi> it's already initialized

<youpi> you just need to pick its result

<youpi> by just setting the cr3 registers

<youpi> just like it is done by i386at_init for the main cpu

<AlmuHS> pmap_bootstrap() ?

<youpi> no

<youpi> really no

<youpi> **it's *JUST* register initialization**

<youpi> nothing more

<AlmuHS> cr3 related functions then

Consultando sobre paginación

<AlmuHS> hi. I continue with my pagging problem. Some days ago, we find the line where the boot stops, but I don't know the reason

- **<AlmuHS> the line is this: https://github.com/AlmuHS/GNUMach_SMP/blob/wip/i386/i386/cpuboot.S#L131**
- **<youpi> AlmuHS: earlier today, jrtc27 pointed out: « I'm guessing the other CPUs might need the temporary mapping in place too »**
- **<youpi> https://github.com/AlmuHS/GNUMach_SMP/blob/smp/i386/i386at/model_dep.c#L520-L542**
- **<AlmuHS> oh, I didn't read this, but I can try It**
- **<AlmuHS> Do I need to translate this code in my assembly snippet?**
- **<AlmuHS> feels a bit complex**
- **<youpi> not this one (which gets rid of the temporary mapping)**
- **<youpi> but probably you'd need to make gnumach keep the page table with the temporary mapping somewhere**
- **<youpi> for your assembly code to use it as a trampoline page table**
- **<youpi> between linear adressing and full page table + segmentation**
- **<AlmuHS> yes, It's possible**
- **<AlmuHS> I have to research a few, because I'm not clear about how to do this**

Completando el desarrollo

- En modo paginación, las macros para sumar la dirección base no funcionan.
 - Hay que modificar algunas partes del código
 - Por suerte, a algunos datos se puede acceder directamente :)
- Poco a poco el código se va llenando de extern y variables globales
 - Tocaré arreglarlo después...

Integrando los procesadores al sistema

- Implementamos `cpu_number()`
- Ya podemos invocar a las funciones para integrar los procesadores al sistema
- Probamos el microkernel
 - ¡¡Y arranca!!

Probando el SMP

- Arrancamos el microkernel, pero con problemas
 - No hay teclado :(
 - El controlador PIC utilizado por gnumach no soporta multiprocesador
 - Se necesita el IOAPIC, pero su configuración es compleja
 - No hay soporte de red
 - El gnumach “upstream” no incluía soporte a NetDDE, necesario para los drivers de red
 - No tenemos acceso al sistema
 - No podemos saber si ha funcionado bien :(

Accediendo al sistema

- Otro desarrollador de Hurd sale al rescate
 - Aplica los parches de Debian GNU/Hurd para añadir los drivers de red a *upstream*
- ¡¡Por fin podemos acceder al sistema!!
- Todos los procesos aparecen ejecutándose en la CPU 0...
 - ¿el scheduler no está usando los nuevos procesadores?

Mas parches

- Las estructuras del scheduler de gnumach no almacenan la información del último procesador en ejecutar el proceso
 - La tabla *stat* tiene ese valor “hardcodeado” a 0
- Hay que añadir ese campo a las estructuras
 - Para ello hay que parchear tanto gnumach como dos servidores de Hurd
 - **De nuevo, recibo ayuda de varios devs de Hurd**

Parcheando el scheduler en gnumach

- Hay que añadir un nuevo campo a un struct que almacene el último procesador usado por el hilo
 - Esa información ya la tiene el scheduler, por lo que es fácil rellenarlo
 - La estructura se modifica en el header
- Pero hay que mantener la retrocompatibilidad
 - Puede haber programas que aún usen el header antiguo sin ese campo en el struct
 - Hay que comprobar si ese campo existe o no en el header que use el programa
- <http://git.savannah.gnu.org/cgiit/hurd/gnumach.git/commit/?id=75267dd103637d38fa95ecdee0eedb16ba0f662c>

Parcheando la tabla stat en Hurd

- La estructura enviada por el scheduler la recibe el servidor *proc* de Hurd, el cual la envía a *procfs* para rellenar la tabla *stat*.
- Hay que modificar ambos servidores para recoger la nueva información y rellenar la tabla *stat* correctamente
 - Pero respetando la retrocompatibilidad

Enviando parches

- Tras comprobar la retrocompatibilidad, envío el parche a través de IRC
 - ¡¡Mi primer parche!!
 - Aviso de que no está completamente testeado
- **El parche es rechazado en varias ocasiones**
 - No sigue la guía de estilo de GNU
 - El indentado no sigue el estilo del resto del fichero
 - ... **Pero nadie se para a probar el parche**

Enviando parches

- **Finalmente es aceptado**
 - **Hacen el commit a master sin probarlo**
 - <http://git.savannah.gnu.org/cgiit/hurd/hurd.git/commit/?id=d8671bc2a0fead7655b9e80736db33d84f14025c>
 - **¡¡Hay un error de compilación!!**
 - Ya había avisado de que no estaba probado...
 - **Corrijo el parche y lo vuelvo a enviar**
 - **Pero ya el anterior está en el repositorio oficial**
 - El nuevo se publica como un nuevo commit
 - http://git.savannah.gnu.org/cgiit/hurd/hurd.git/log/?qt=grep&q=last_processor

Comprobando las CPU en uso

- Ya con la tabla *stat* corregida, podemos comprobar qué procesador usa cada hilo
- Todos van a la CPU 0 :(
 - Los otros procesadores parecen quedarse bloqueados en alguna parte
 - No sabemos la causa

Recomenzando

- Por si acaso, recomendamos el proyecto para refactorizar todo
 - Eliminamos variables globales, mejoramos los direccionamientos...
 - Y a ver si hay suerte ;)
- Este trabajo sigue a día de hoy
 - En Noviembre de 2020 se logró incorporar la primera fase (enumeración de los procesadores) al “upstream” de gnumach
 - Pero esta es otra historia...