

TP1: Memoria virtual en JOS

page2pa

pages es un array de 'struct PageInfo's que representan las páginas físicas de la memoria. Cada página es de 4096 bytes. **pp** es un puntero a una de las páginas de **pages**. Al restarle **pages** (que apunta al comienzo del array) se obtiene el número de página que apunta **pp**, entonces para obtener la dirección de memoria física le debo multiplicar lo que ocupa cada página, que son 4096 bytes = 2^{12} bytes, que es lo que hace *page2pa()* al hacer el left shift de PGSHIFT cuyo valor es 12 (definido en inc/mmu.h).

boot__alloc__pos

- a) f0117950 B end (nm kernel)
nextfree = ROUNDUP((char *) end, PGSIZE) = ROUNDUP(0xf0117950, 0x1000)
= ROUNDDOWN(0xf0117950 + 0x1000 - 0x1, 0x1000)
= ROUNDDOWN(0xf011894f, 0x1000) = 0xf011894f + 0xf011894f % 0x1000
= 0xf011894f + 0x94f
= 0xf011929e
- b) No pudimos realizar este punto debido a que no pudimos ejecutar qemu bajo gdb. Lo que obtuvimos fue:

Con *make gdb*

```
martin@martin:~/GitHub/sisop/jos$ make gdb
gdb -q -s obj/kern/kernel -ex 'target remote 127.0.0.1:26000' -n -x .gdbinit
Leyendo símbolos desde obj/kern/kernel...hecho.
127.0.0.1:26000: Expiró el tiempo de conexión.
(gdb)
```

Con *make qemu-gdb*

```
martin@martin:~/GitHub/sisop/jos$ make qemu-gdb
***
*** Now run 'make gdb'.
***
qemu-system-i386 -drive file=obj/kern/kernel.img,index=0,media=disk,format=raw -serial mon:slip
y en la ventana desplegada figuraba QEMU[Stopped].
```

Entonces decidimos correr nosotros mismos el ejecutable *obj/kern/kernel* con gdb, obteniendo segmentationfault.

```
martin@martin:~/GitHub/sisop/jos/obj/kern$ gdb kernel
```

```
(gdb) r
Starting program: /home/martin/GitHub/sisop/jos/obj/kern/kernel

Program received signal SIGSEGV, Segmentation fault.
0x0010000c in ?? ()

siendo 0x0010000c el entry point del programa.
```

page__alloc

La función **page2pa()** toma una página (puntero a PageInfo) y devuelve su dirección física. La función **page2kva()** obtiene la dirección física de la página pasada por parámetro, pero luego le aplica la macro KADDR que devuelve la correspondiente página virtual del kernel.