

La salida del comando de pmap para un proceso en una máquina es la siguiente.

*> Solo muestra memoria virtual*

```
antonio@abyecto:~$ pmap -x 3709
3709:  ./shell
Address   Kbytes    RSS   Dirty Mode  Mapping
4k ( 08048000         4        4      0 r----- shell
      08049000       636       572      0 r-x--- shell
      080e8000       236       128      0 r----- shell
      08124000         8         8      4 r----- shell
      08126000         8         8      8 rw----- shell
      08128000        76        12     12 rw----  [ anon ]
      091c9000       136         8      8 rw----  [ anon ]
      b7f11000        16         0      0 r-----  [ anon ]
      b7f15000         8         4      0 r-x---  [ anon ]
      bfd4f000       132        16     16 rw----  [ stack ]
-----
total kB   1260       760       48
```

*cuanto está en memoria (Resident Set Size)*

*Compilado estáticamente, porque no aparecen las librerías ni el linkador*

antonio@abyecto:~\$

1.-¿Se trata de una máquina de 32 o de 64 bits?

**Es una máquina de 32 bit. Las direcciones son de 8 cifras hexadecimales**

2.-¿Cual es el tamaño del espacio de direcciones del proceso?

**1260 kbytes**

3.-¿Cuanta memoria física tiene asignada el proceso?

**En memoria hay 760 kbytes (Resident Set Size)**

4.-¿Cuál es el tamaño de página para esta arquitectura?

**Todas las regiones del proceso comienzan en una dirección cuyas tres últimas cifras hexadecimales son 000. Además el tamaño de la zona mas pequeña es 4k. Por tanto podemos asumir que el tamaño de página es 4K**

5.-¿Cuál es el tamaño de cada entrada de la tabla de páginas, 1, 2, 4, 8 o 16 bytes?

**No se dice nada en contrario, asumimos que el espacio físico es tambien de 32 bit y dado que necesita 20 bits para la dirección del marco físico, además de los bits de presencia, referencia, privilegio, etc. dos bytes no sería suficiente. cuatro bytes por entrada es lo razonable. Si el espacio físico fuese de 28 o 36 bits, 4 bytes por entrada tambien sería lo razonable.**

6.-Suponiendo que el sistema utiliza paginación en dos niveles. ¿Cuánto ocupa la tabla de páginas de dicho proceso?

**La manera mas sencilla de contestar esta pregunta se ve al final, en el apartado otras consideraciones, no obstante por claridad haremos primero la solución completa.**

**Vamos a reescribir las zonas del proceso poniendo la direccion de comienzo y fin y el tamaño en páginas. El número de páginas lo obtenemos dividiendo el tamaño en Kbytes entre 4, y la direccion de fin sumándole a la dirección del principio el tamaño en hexadecimal (número de páginas en hexadecimal con tres ceros, ejemplo 2 páginas es 2000, 59 páginas es 3B000)**

Principio	Fin	Kbytes	Número páginas	Número páginas (Hex)	Tamano (Hex)	perm	
08048000	08049000	4	1	1	1000	r--	shell
08049000	080E8000	636	159	9F	9F000	r-x	shell
080e8000	08123000	236	59	3B	3B000	r--	shell
08124000	08126000	8	2	2	2000	r--	shell
08126000	08128000	8	2	2	2000	rw-	shell
08128000	0813B000	76	19	13	13000	rw-	[ anon ]
091c9000	091EB000	136	34	22	22000	rw-	[ anon ]
b7f11000	B7F15000	16	4	4	4000	r--	[ anon ]
b7f15000	B7F17000	8	2	2	2000	r-x	[ anon ]
bfd4f000	BFD70000	132	33	21	21000	rw-	[ stack ]

**Si la paginación es en dos niveles, y dado que tenemos direcciones de 32 bits y 12 bits para el desplazamiento dentro de la página, hay 20 bits para el número de página, 10 para cada nivel. Con 10 bits hay 1024 (0x400) entradas y como las entradas son de 4 bytes, cada entrada de la tabla de páginas raíz aúnta a una tabla de segundo nivel que ocupa exactamente una página.**

Dirección	Numero página (bin)	Entrada tabla páginas nivel 1		Entrada inicial tabla páginas nivel 2		Págs
08048000	0000 1000 0000 0100 1000	0000100000	0x20	00 0100 1000	0x48	0x01
08049000	0000 1000 0000 0100 1001	0000100000	0x20	00 0100 1001	0x49	0x9F
080e8000	0000 1000 0000 1110 1000	0000100000	0x20	00 1110 1000	0xE8	0x3B
08124000	0000 1000 0001 0010 0100	0000100000	0x20	01 0010 0100	0x124	0x02
08126000	0000 1000 0001 0010 0110	0000100000	0x20	01 0010 0110	0x126	0x02
08128000	0000 1000 0001 0010 1000	0000100000	0x20	01 0010 1000	0x128	0x13

Dirección	Numero página (bin)	Entrada tabla páginas nivel 1		Entrada inicial tabla páginas nivel 2		Págs
091C9000	0000 1001 0001 1100 1001	0000100100	0x24	01 1100 1001	0x1C9	0x22
B7F11000	1011 0111 1111 0001 0001	1011011111	0x2DF	11 0001 0001	0x311	0x04
B7F15000	1011 0111 1111 0001 0101	1011011111	0x2DF	11 0001 0101	0x315	0x02
BFD4F000	1011 1111 1101 0100 1111	1011111111	0x2ff	01 0100 1111	0x14F	0x21

**Si ahora ponemos para cada zona qué entradas en cada tabla de páginas usa**

Dirección	Entrada tabla páginas nivel 1	Entrada inicial tabla páginas nivel 2	Núm Págs	Entradas usadas en tabla de páginas de nivel 2
08048000	0x20	0x48	0x01	0x48
08049000	0x20	0x49	0x9F	0x49->0xE7
080e8000	0x20	0xE8	0x3B	0xE8->0x122
08124000	0x20	0x124	0x02	0x124->0x125
08126000	0x20	0x126	0x02	0x126->0x127
08128000	0x20	0x128	0x13	0x128->0x13A
091C9000	0x24	0x1C9	0x22	0x1C9->0x1EA
B7F11000	0x2DF	0x311	0x04	0x311->0x314
B7F15000	0x2DF	0x315	0x02	0x315->0x316
BFD4F000	0x2ff	0x14F	0x21	0x14f->0x16F

**En la tabla anterior se muestra cada dirección de inicio, qué entrada en la tabla de páginas de primer nivel usa, y que entradas en la tabla de páginas de segundo nivel usa cada zona.**

**Por tanto para el proceso se usa**

- La tabla de páginas raíz o de primer nivel
- La página de segundo nivel de la tabla de páginas apuntada por la entrada 0x20 de la tabla de páginas raíz, de la que usamos las entradas 0x48 hasta 0x122, y 0x124 hasta 0x13A.
- La página de segundo nivel de la tabla de páginas apuntada por la entrada 0x24 de la tabla de páginas raíz, de la que usamos las entradas 0x1C9 hasta 0x1EA.
- La página de segundo nivel de la tabla de páginas apuntada por la entrada 0x2DF de la tabla de páginas raíz, de la que usamos las entradas 0x311 hasta 0x316
- La página de segundo nivel de la tabla de páginas apuntada por la entrada 0xDFF de la tabla de páginas raíz de la que usamos las entradas 0x14F hasta 0x16F

**Total 20 Kbytes ocupa la tabla de páginas del proceso (la tabla de páginas raíz + cuatro páginas de segundo nivel).**

7.- ¿Cuanto ocuparía la tabla de páginas si fuese en un nivel?.

**Con 20 bits tendríamos  $2^{20}$  entradas y cada entrada 4 bytes, en total 4Mb, aunque la mayor parte de las entradas (todas excepto 315) no se usarían**

8.-¿En que direcciones de memoria física está la pila del proceso?  
**No podemos saberlo, se muestran solo direcciones virtuales**

9.-¿Está compilado estática o dinámicamente?

**Estaticamente: no aparecen en el espacio de direcciones ni las librerías ni el linkador**

## OTRAS CONSIDERACIONES

El ejemplo corresponde a una máquina intel de 32 bits, con lo que las suposiciones que hemos hecho de tamaño de página, del tamaño de cada entrada de la tabla de páginas son correctas.

El ejercicio podría haberse simplificado, si nos fijamos que en la primera tabla muchas zonas son contiguas, podemos considerar las zonas contiguas como una sola zona (prescindiendo de lo que contienen y de sus permisos). Así tendríamos

- A partir de la dirección 08048000 hay 219 (0xDB) páginas (aunque con distintos permisos)
- A partir de la dirección 08124000 hay 23 (0x17) páginas (aunque con distintos permisos)
- A partir de la dirección 091C9000 hay 24 (0x22) páginas
- A partir de la dirección B7F11000 hay 6 páginas (aunque con distintos permisos)
- A partir de la dirección BFD4F000 hay 33 (0x22) páginas

Y así tendríamos lo mismo a lo que llegamos anteriormente.

Dirección	Numero página	Entrada tabla páginas nivel 1		Entrada inicial tabla páginas nivel 2		última
08048000	0000 0100 0000 0100 1000	0000010000	0x10	00 0010 1000	0x48	0x122
08124000	0000 1000 0001 0010 0100	0000100000	0x10	01 0010 0100	0x124	0x13A
091C9000	0000 1001 0001 1100 1001	0000100100	0x24	01 1100 1001	0x1c9	0x1EA
B7F11000	1011 0111 1111 0001 0001	1011011111	0x2DF	11 0001 0001	0x311	0x316
BFD4F000	1011 1111 1101 0100 1111	1011111111	0x2ff	01 0100 1111	0x14f	0x16F

Todavía se podría hacer de manera mas sencilla si miramos que entrada en la tabla de páginas raíz o de primer nivel de las direcciones de inicio y fin de cada zona, así puede determinarse cuantas páginas de segundo nivel se han asignado.

<b>Direccion Inicio</b>	<b>Entrada TP nivel 1 dir inicio</b>	<b>Dirección fin</b>	<b>Entrada TP nivel 1 dirrección fin</b>
08048000	0x20	08049000	0x20
08049000	0x20	080E8000	0x20
080e8000	0x20	08123000	0x20
08124000	0x20	08126000	0x20
08126000	0x20	08128000	0x20
08128000	0x20	0813B000	0x20
091C9000	0x24	091EB000	0x24
B7F11000	0x2DF	B7F15000	0x2DF
B7F15000	0x2DF	B7F17000	0x2DF
BFD4F000	0x2ff	BFD70000	0x2ff

Vemos que solo se utilizan 4 entradas distintas de la tabla de páginas raíz o de primer nivel (0x20, 0x24, 0x2DF y 0x2FF) por lo que hay asignadas 4 páginas de segundo nivel. además de la raíz Por tanto 5 páginas de tabla de páginas (de 4K cada una): 20K ocupa la tabla de páginas.