

1. Esquema de traducció d'adreces. Doneu totes les mides (en bits i Bytes) de les estructures que hi aparegui i poseu-hi els multiplicadors que calgui. (fins a 2 punts)

La mida dels segments és 8KBytes, la mida de les pàgines és 2KBytes i la mida de les paraules és 1Byte. Això significa que hi haurà $81024=8192$ paraules per segment i $21024=2048$ paraules per pàgina. A més, hi haurà 4 segments per procés i un màxim de $8192/4=2048$ pàgines per procés.

2. Doneu un exemple de la taula de segments i de les taules de pàgines del procés P1. Supposeu que les poseu a la cel·la 5 i que ocupen el mínim espai possible. Poseu les adreces correctament.(fins a 2 punts)

La taula de segments del procés P1 es trobaria a la cel·la 5 i ocuparia un espai mínim de $48=32$ paraules. Això significa que la taula de segments ocuparia $32/8=4$ pàgines. Així, la taula de segments del procés P1 es trobaria a les cel·les 5, 6, 7 i 8. La taula de pàgines del procés P1 es trobaria a la cel·la 9 i ocuparia un espai mínim de $20484=8192$ paraules. Això significa que la taula de pàgines ocuparia $8192/8=1024$ pàgines. Així, la taula de pàgines del procés P1 es trobaria a les cel·les 9 a 1032.

0		11	DP1	22	DP1	33	DP1	44	DP1	55	DP1	66	DP1	77	DP1	88	DP1
1		12	DP1	23	DP1	34	DP1	45	DP1	56	DP1	67	DP1	78	DP1	89	DP1
2		13	DP1	24	DP1	35	DP1	46	DP1	57	DP1	68	DP1	79	DP1	90	DP1
3		14	DP1	25	DP1	36	DP1	47	DP1	58	DP1	69	DP1	80	DP1	91	DP1
4		15	DP1	26	DP1	37	DP1	48	DP1	59	DP1	70	DP1	81	DP1	92	DP1
5	CP1	16	DP1	27	DP1	38	DP1	49	DP1	60	DP1	71	DP1	82	DP1	93	DP1
6	CP1	17	DP1	28	DP1	39	DP1	50	DP1	61	DP1	72	DP1	83	DP1	94	DP1
7	CP1	18	DP1	29	DP1	40	DP1	51	DP1	62	DP1	73	DP1	84	DP1	95	DP1
8	CP1	19	DP1	30	DP1	41	DP1	52	DP1	63	DP1	74	DP1	85	DP1	96	DP1
9	DP1	20	DP1	31	DP1	42	DP1	53	DP1	64	DP1	75	DP1	86	DP1	...	
10	DP1	21	DP1	32	DP1	43	DP1	54	DP1	65	DP1	76	DP1	87	DP1	1032	DP1

3. Rang d'adreçament lògic i físic del procés P1. (fins a 2 punts)

El rang d'adreçament lògic del procés P1 seria de 0 a $81924-1=32767$ paraules. El rang d'adreçament físic seria de 0 a $20481024-1=2097151$ paraules.

Paginació Segmentada

Dibuixa l'esquema de traducció d'adreces de la paginació segmentada. Raona un escenari on creguis que podria ser útil.

Ricard Bosch Perianes i Joan Bonell Ruiz

L'esquema de traducció d'adreces de la paginació segmentada es pot representar de la següent manera:

Adreça lògica del procés -> Índex del segment -> Adreça del segment a la taula de segments -> Índex de la pàgina -> Adreça de la pàgina a la taula de pàgines -> Desplaçament a la pàgina -> Adreça física de la paraula

Un escenari on aquesta esquema de traducció d'adreces pot ser útil és quan es vol utilitzar un sistema operatiu per a gestionar la memòria d'un ordinador amb diferents processos que comparteixen memòria. La paginació segmentada permet separar la memòria física en segments i pàgines de manera que cada procés tingui una adreça lògica i una adreça física diferent. Això facilita la gestió de la memòria ja que es pot assignar i desallotjar espai de memòria de manera més eficient i segura.

Expliqueu que fa aquest programa de forma detalla. Podeu afegir comentaris a les línies. (fins a 1 punts).

Aquest programa llegeix un fitxer (en aquest cas "ex3.c") i el mapa a memòria. A continuació, recorre totes les línies del fitxer i les imprimeix per pantalla.

Les línies incloent els fitxers de capçalera (fcntl.h, unistd.h, sys/mman.h, sys/stat.h, stdio.h i stdlib.h) indiquen quines funcionalitats del sistema s'estan utilitzant en el programa.

La funció stat() es fa servir per obtenir informació del fitxer "ex3.c". La funció open() obre el fitxer "ex3.c" en mode de lectura només (O_RDONLY). Si open() o fstat() tenen errors, es mostra un missatge d'error i es retorna 1.

La funció mmap() mapa el fitxer "ex3.c" a memòria. PROT_NONE indica que no es permet l'accés de lectura ni escriptura a la memòria mapejada, i MAP_PRIVATE indica que la memòria mapejada és privada i no es comparteix amb altres processos. Si mmap() té error, es mostra un missatge d'error i es retorna 1.

La funció munmap() desmapa la memòria mapejada. La funció close() tanca el fitxer "ex3.c". Finalment, es retorna 0.

Si executeu el programa, observareu que hi ha un segmentation fault. Corregiu-lo.

Per arreglar el segmentation fault, cal assegurar que el programa tingui accés de lectura a la memòria mapejada del fitxer "ex3.c". Això es pot fer modificant la crida a la funció mmap() de la següent manera:

```
//buf = mmap(NULL, size, PROT_NONE, MAP_PRIVATE, fd, 0);  
buf = mmap(NULL, size, PROT_READ, MAP_PRIVATE, fd, 0);
```

PROT_READ indica que es permet l'accés de lectura a la memòria mapejada, i MAP_PRIVATE indica que la memòria mapejada és privada i no es comparteix amb altres processos.

Així, quan es fa la crida a la funció printf() per imprimir cada línia del fitxer "ex3.c", el programa tindrà accés de lectura a la memòria mapejada i no es produirà el segmentation fault.