

5. MEMORIAREN KUDEAKETA

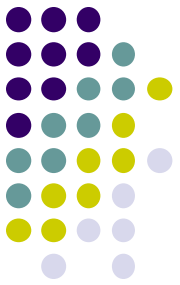
5.9 MEMORIAN PROIEKTATUTAKO FITXATEGIAK

- Fitxategi bat, eskaera hori sortu duen prozesuaren memorian proiektatzeko sistema – deia:

*void *mmap(void *start, size_t length, int prot, int flags, int fd, off_t offset);*

- Memoriako zein helbidetan proiektatu den adierazten duen erakuslea itzultzen du
 - `start` zein helbidetan proiektatuko den agindu.
 - Baldin `NULL`, SE-ak aukeratzen du
 - `length` proiektatu nahi den fitxategiaren byte-kopurua
 - `prot` atzipen/babes-mota (RWX)
 - Fitxategiaren irekitzean eman diren baimenekin bat etorri behar da
 - `flags` proiektzio-motaren informazioa (partekatu/pribatu, etab)
 - `fd` proiektatu beharreko fitxategiaren deskribatzailea `fd`
 - Aurretiaz fitxategia ireki behar da `fd`-en balioa izateko
 - `offset` fitxategiaren zein kokapenetik aurrera proiektatu
- Desproiektatu prozesu baten helbide – espazio zati bat `offset` helbidetik hasita `offset+length` helbideraino:

*void munmap(void *offset, size_t length);*



5. MEMORIAREN KUDEAKETA

5.9 MEMORIAN PROIEKTATUTAKO FITXATEGIAK

- Jatorrizko fitxategi baten kopia egin:

```
void main(int argc, char **argv)
{
    int i, fdj, fdh;
    char *jatorri, *helburu, *p, *q;
    struct stat bstat;

    if (argc!=3) {
        fprintf (stderr, "Erabilera zuzena: %s jatorritik helburura\n", argv[0]);
        exit(1);
    }

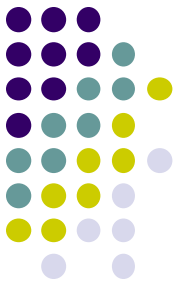
    /* Jatorria ireki irakurtzeko */
    if ((fdj=open(argv[1], O_RDONLY))<0) {
        perror("Ezin da ireki jatorrizko fitxategia");
        exit(1);
    }

    /* Sortu helburua, 640 baimenaz (rw-r----- */
    if ((fdh=open(argv[2], O_CREAT|O_TRUNC|O_RDWR, 0640))<0) {
        perror("Ezin da sortu helburu fitxategia");
        close(fdj); exit(1);
    }
}
```

Baimenak

5. MEMORIAREN KUDEAKETA

5.9 MEMORIAN PROIEKTATUTAKO FITXATEGIAK



```
if (fstat(fdj, &bstat)<0) { /* Jatorrizko fitxategiaren luzera lortu */
    perror("Ezin da irakurri jatorrizko fitxategiaren egoera");
    close(fdj); close(fdh); unlink (argv[2]); exit(1);
}

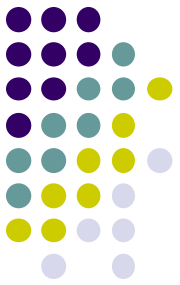
if (ftruncate(fdh, bstat.st_size)<0) { /* Helburuaren luzera jatorrizkoaren bera*/
    perror("Errorea luzera finkatzerakoan");
    close(fdj); close(fdh); unlink (argv[2]); exit(1);
}

/* Jatorrizko fitxategia memorian proiektatu */
if ((jatorri=mmap((caddr_t) 0, bstat.st_size, PROT_READ, MAP_SHARED, fdj, 0)) == MAP_FAILED) {
    perror("Errorea jatorriaren proiektzioan");
    close(fdj); close(fdh); unlink (argv[2]); exit(1);
}

/* Helburu fitxategia memorian proiektatu */
if ((helburu=mmap((caddr_t) 0, bstat.st_size, PROT_WRITE, MAP_SHARED, fdh, 0)) == MAP_FAILED) {
    perror("Errorea helburuaren proiektzioan");
    close(fdj); close(fdh); unlink (argv[2]); exit(1);
}
```

5. MEMORIAREN KUDEAKETA

5.9 MEMORIAN PROIEKTATUTAKO FITXATEGIAK



```
/* Orain ez dira behar fitxategien deskribatzaileak: itxi */
close(fdj);
close(fdh);

/* Kopiaren begizta: memoria primarioan hitzez hitz kopiatu, S/I eragiketarik egin gabe*/
p=jatorri;
q=helburu;
for (i=0; i<bstat.st_size; i++)
    *q++= *p++;

/* Memoria primarioan burutu den kopia, sekundariora irauli eta proiektzioaren memoria askatu */
munmap(jatorri, bstat.st_size);
munmap(helburu, bstat.st_size);
}
```