

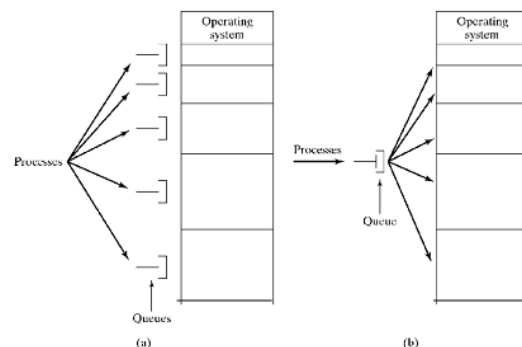
6. Lageradministration - Fysisk lager

RAM lageret er et fladt array. **Celler** med adresser 0, 1, n-1 (n celler), som hver indeholder et antal bits. Adresserummets størrelse bestemmes af adressenbussens bredde – med 32bit kan 4G celler adresseres.

Problemer når programmet fylder mere end main memory, eller i multiprogramming systemer.

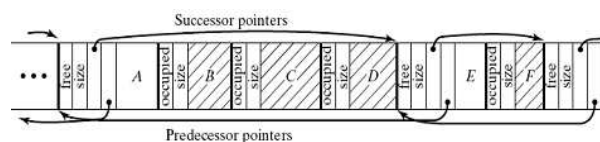
1. Statisk opdeling af lageret (fx OS og user programmer + data)

- Deles yderligere op ved multiprogramming
- Separate proces køer:** Proces til mindste partition, der passer (best-fit). Måske ubrugte partitioner.
- Én kø til alle partitioner:** OS bestemmer, hvilke processer, der skal hvorhen
 - Nemt at impl.
 - Begrænsning af processtørrelser. Intern fragmentering (ubrugt plads i hver partition).



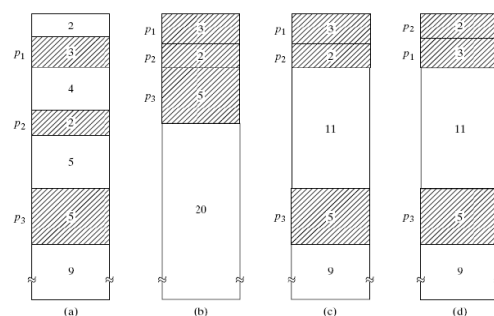
2. Dynamisk opdeling af lageret: Man får den plads man skal bruge.

- Når processer terminerer, kommer der et hul → ekstern fragmentering (hul udenfor partitionen).
- Coalescing:** Nabohuller vokser sammen (OS skal sørge for dette)
- Kædet liste af huller:** Tomme huller indeholder pointers til næste tomme hul. Lidt ineffektivt.
- Bitmap impl.:** En streng af 0'er (fri) og 1'er (taget). Bitvis AND operation for at frigøre. Bitvis OR for at allokere. Som array.
 - Fx 1024 bits: B[128] á 8 bit. B[0] & '11100111' 0 frigøres – 1 er som før. B[0] | '11000000' allokerer de første to huller, og lader resten være som før.
 - Ingen behov for coalescing, da 0'erne jo står ved siden af hinanden...
- Placeringsalgoritmer:** Memory manager skal finde hul, hvor vores n bytes kan være.
 - First-fit (bedst), next-fit (fortsætter hvor søgningen slap), best-fit (dårligst), worst-fit



Compaction

- Ekstern fragmentering kan minimeres ved at compacte. Den brugte plads flyttes til en af enderne i rammen.
- Tidskrævende. Kun for dynamisk opdeling!
- Fuld, partiel eller minimal data flytning.



Swapping

- Hvis rammen er fuld, udnytter vi sekundær lager.
- OS bestemmer hvor på disken vi swapper hen (swap space, for at undgå for meget overhead).
- Mere effektiv, da der kan swappes så mange processer ud, der er brug for.