

2° LIVELLO DEL SISTEMA OPERATIVO

GESTORE DELLA MEMORIA CENTRALE

GESTORE DELLA MEMORIA CENTRALE

Per poter eseguire più processi contemporaneamente e indipendentemente dal numero di processori, **NON BASTA LA SOLA VIRTUALIZZAZIONE DELLA CPU**, è necessario che il sistema operativo si occupi anche della **VIRTUALIZZAZIONE DELLA MEMORIA CENTRALE**.

Ciò è necessario, poiché i processi che utilizzano processori virtuali per poter essere eseguiti **DEVONO RISIEDERE IN MEMORIA CENTRALE**.

PROBLEMA: l'insieme di N processi generalmente occupa molto più spazio di quello disponibile nella memoria centrale.

VIRTUALIZZAZIONE MEMORIA CENTRALE

OBIETTIVO: Suddividere la memoria centrale tra gli N processi dando ad ogni processo l'impressione di poter disporre dell'intera memoria centrale fisica (**TRASPARENZA**).

VIRTUALIZZAZIONE MEMORIA CENTRALE

OBIETTIVO: Suddividere la memoria centrale tra gli N processi dando ad ogni processo l'impressione di poter disporre dell'intera memoria centrale fisica (**TRASPARENZA**).

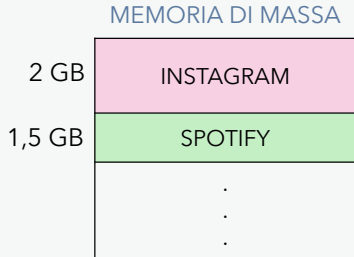
COME?!

VIRTUALIZZAZIONE MEMORIA CENTRALE

SEGMENTAZIONE: Ogni processo viene suddiviso e caricato in memoria centrale a **SEGMENTI**, ovvero viene caricata in una **PORZIONE CONTINUA** di spazio di memoria solo una porzione del programma di dimensione non necessariamente prefissata.

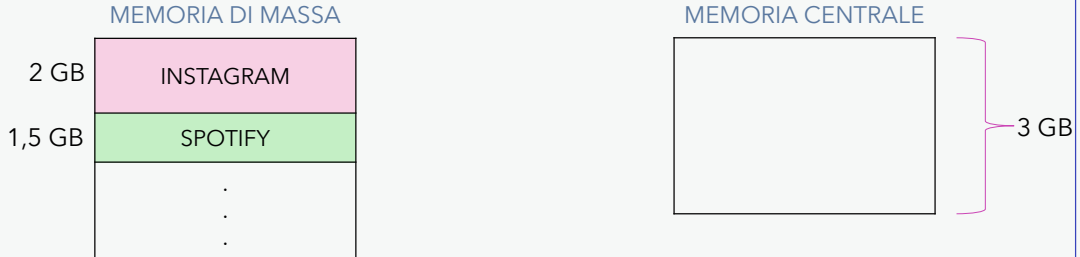
VIRTUALIZZAZIONE MEMORIA CENTRALE

SEGMENTAZIONE: Ogni processo viene suddiviso e caricato in memoria centrale a **SEGMENTI**, ovvero viene caricata in una **PORZIONE CONTINUA** di spazio di memoria solo una porzione del programma di dimensione non necessariamente prefissata.



VIRTUALIZZAZIONE MEMORIA CENTRALE

SEGMENTAZIONE: Ogni processo viene suddiviso e caricato in memoria centrale a **SEGMENTI**, ovvero viene caricata in una **PORZIONE CONTINUA** di spazio di memoria solo una porzione del programma di dimensione non necessariamente prefissata.



VIRTUALIZZAZIONE MEMORIA CENTRALE

SEGMENTAZIONE: Ogni processo viene suddiviso e caricato in memoria centrale a **SEGMENTI**, ovvero viene caricata in una **PORZIONE CONTINUA** di spazio di memoria solo una porzione del programma di dimensione non necessariamente prefissata.



VIRTUALIZZAZIONE MEMORIA CENTRALE

Un processo durante la sua esecuzione generalmente non utilizza tutte le istruzioni e i dati presenti nel programma che sta eseguendo, ma solo **UNA PORZIONE DI TALI DATI E ISTRUZIONI**. Pertanto non è necessario caricare in memoria centrale l'intero programma per poter essere eseguito, ma solamente le parti necessarie all'esecuzione degli specifici compiti richiesti.

I processi non sono consapevoli della propria segmentazione in memoria, essa viene infatti eseguita dal S.O. che **VIRTUALIZZA LA MEMORIA CENTRALE** dando l'impressione ai processi di poterla utilizzare completamente e liberamente.

VIRTUALIZZAZIONE MEMORIA CENTRALE

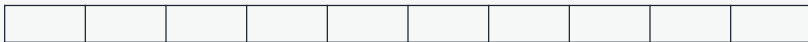
Il Sistema Operativo ogni volta che il processo richiede di utilizzare segmenti nuovi per proseguire procedere allo **SWAPPING**, ovvero al caricamento/scaricamento della memoria centrale per consentire al processo di continuare la sua esecuzione senza che esso riesca a rendersi conto di non star sfruttando tutta la memoria ma solo parte di essa.

La **SEGMENTAZIONE** è una tecnica che ha come danno collaterale il **PROBLEMA DELLA FRAMMENTAZIONE**.

PROBLEMA DELLA FRAMMENTAZIONE

ESEMPIO: Viene utilizzata una memoria centrale di 10 unità e su di essa avvengono le seguenti operazioni:

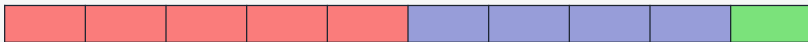
- 1) CARICAMENTO di un processo A nelle prime 5 unità
- 2) CARICAMENTO di un segmento B nelle successive 4 unità
- 3) CARICAMENTO di un segmento C nell'ultima unità



PROBLEMA DELLA FRAMMENTAZIONE

ESEMPIO: Viene utilizzata una memoria centrale di 10 unità e su di essa avvengono le seguenti operazioni:

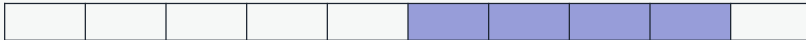
- 1) CARICAMENTO di un processo **A** nelle prime **5** unità
- 2) CARICAMENTO di un segmento **B** nelle successive **4** unità
- 3) CARICAMENTO di un segmento **C** nell'**ultima** unità



PROBLEMA DELLA FRAMMENTAZIONE

ESEMPIO: Viene utilizzata una memoria centrale di 10 unità e su di essa avvengono le seguenti operazioni:

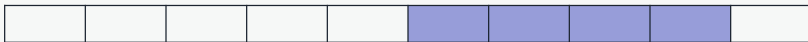
- 1) SCARICAMENTO del processo **A** (libero **5** unità di memoria)
- 2) SCARICAMENTO del segmento **C** (libero **una** unità di memoria)



PROBLEMA DELLA FRAMMENTAZIONE

ESEMPIO: Viene utilizzata una memoria centrale di 10 unità e su di essa avvengono le seguenti operazioni:

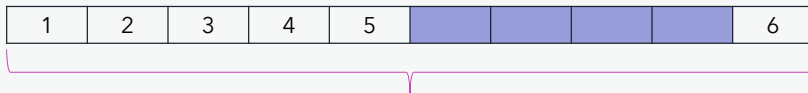
- 1) SCARICAMENTO del processo **A** (libero **5** unità di memoria)
- 2) SCARICAMENTO del segmento **C** (libero **una** unità di memoria)
- 3) **CARICAMENTO del segmento D che richiede 6 unità di memoria!**



PROBLEMA DELLA FRAMMENTAZIONE

ESEMPIO: Viene utilizzata una memoria centrale di 10 unità e su di essa avvengono le seguenti operazioni:

- 1) SCARICAMENTO del processo **A** (libero **5** unità di memoria)
- 2) SCARICAMENTO del segmento **C** (libero **una** unità di memoria)
- 3) CARICAMENTO del segmento D che richiede 6 unità di memoria!**



SPAZIO FRAMMENTATO (non contiguo) NON UTILIZZABILE DA D!



PROBLEMA DELLA FRAMMENTAZIONE

Un modo per risolvere questo problema è utilizzare la **DEFRAMMENTAZIONE**, che sposta i segmenti in modo da lasciare lo spazio inutilizzato tutto all'inizio o alla fine della memoria. Purtroppo però è un'operazione molto costosa in termini di tempo e consumo delle risorse.

Alla segmentazione si affianca quindi un'altra tecnica: la **PAGINAZIONE**, la quale suddivide la memoria logica e fisica in pagine di **DIMENSIONE FISSA** (non più variabile). Così facendo la frammentazione diminuisce ottenendo prestazioni migliori.

