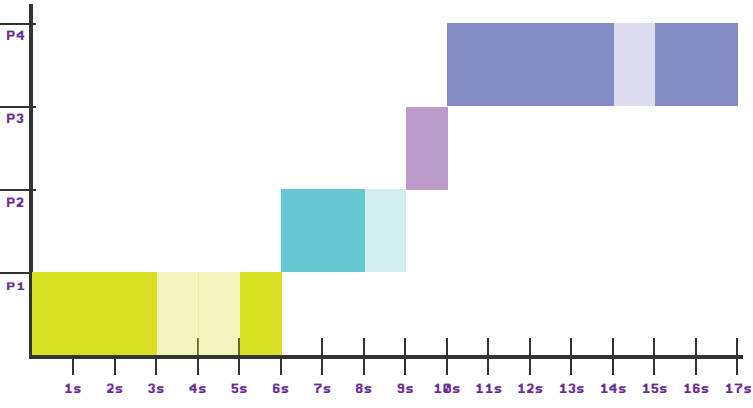
CONSIDERANDO 4 PROCESSI IN ORDINE, CON I TEMPI DI ESECUZIONE I/O DATI IN TABELLA - INDIVIDUARE IL MODO + EFFICACE PER LA GESTIONE E L'ESECUZIONE DEI PROCESSI

SCHEDULING CPU

SCHEDULING SPAZI MEMORIA CPU:
Le istruzioni dell'esecuzione del processo P1 da parte della **CPU** vengono caricate nelle celle **RAM** (***memoria primaria*** o centrale). Il processo può essere caricato da una ***memoria secondaria***, o memoria di massa, che è un tipo di memoria non volatile (**USB, HDD, SSD**).
Nel gestore time-sharing - evoluzione del multi-tasking - per ottimizzare l'allocazione della memoria ai vari task si utilizza ad esempio l'allocazione lineare, che però lascia parecchi spazi vuoti di memoria inutilizzata tra un task concluso e l'esecuzione di un altro, a causa delle dimensioni variabili degli spazi di memoria di cui necessitano i vari programmi in esecuzione -> fenomeno di **FRAMMENTAZIONE**: impatto negativo sulla performance dei PC. Passiamo quindi alla soluzione -> **PAGING**: la memoria viene divisa in blocchi di uguale misura e per ogni processo viene allocata un porzione di memoria pari al paging o ad un suo multiplo intero. Si può aggiungere anche il concetto di memoria virtuale con aree si swap per processi messi in attesa di esecuzione.

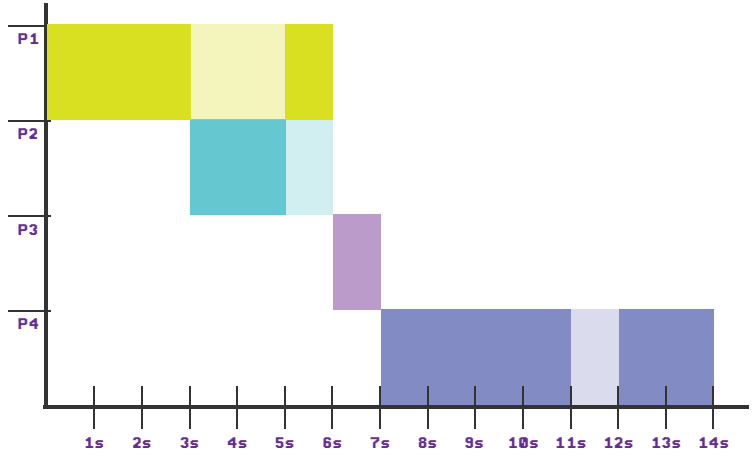
PROCESSO 1: ESECUZIONE 3s - ATTESA 2s - ESECUZIONE POST ATTESA 1s
PROCESSO 2: ESECUZIONE 2s - ATTESA 1s - ESECUZIONE POST ATTESA 0s
PROCESSO 3: ESECUZIONE 1s - ATTESA 0s - ESECUZIONE POST ATTESA 0s
PROCESSO 4: ESECUZIONE 4s - ATTESA 1s - ESECUZIONE POST ATTESA 2s

#MONOTASKING:
Nel sistema mono-tasking, che non supporta l'esecuzione parallela di più processi, l'intera esecuzione dei 4 processi, uno dopo l'altro, coi relativi tempi di attesa, impiegherà in totale 17sec, visto che non è possibile sospenderne uno.



#MULTI-TASKING:

Nel sistema multi-tasking, è possibile l'esecuzione di più processi contemporaneamente, i processi possono essere interrotti per spostare l'attenzione del processore su un altro processo. Si utilizza la pianificazione con **PRELAZIONE** (***preemptive multitasking***) che fa in modo che quando un processo è in attesa di eventi esterni, la CPU possa essere impiegata per altro. Così, quando il processo A passerà dallo stato di esecuzione allo stato di attesa, la CPU potrà essere impiegata per eseguire le istruzioni del processo B. L'esecuzione dei 4 processi avverrà in 14 secondi.



#TIME-SHARING:

Evoluzione dei sistemi multi-tasking. Ogni processo viene eseguito in maniera ciclica per piccole porzioni di tempo che prendono il nome di **QUANTI**. In presenza di una CPU con velocità sufficientemente elevata, il sistema time-sharing darà l'impressione di un'evoluzione parallela dei processi. Valutando 1 QUANTO - come 1 secondo, l'esecuzione dei 4 processi avverrà in 13 secondi.