*Introduzione*

Si definisce *sistema operativo* un insieme di utilità progettate per:

1. Offrire all’utente un’astrazione più semplice e potente della macchina assembler, dunque una macchina virtuale, definita come ambiente virtuale dove eseguire applicazioni e originariamente per sistemi multiutente. Essa deve essere semplice da usare (senza bisogno di conoscenze di microprogrammazione) e potente (es., usando la memoria secondaria per realizzare una più ampia memoria principale virtuale).
2. Gestire in maniera ottimale le risorse fisiche e logiche dell’elaboratore (definendo l’ottimalità come minimizzazione dei tempi di attesa e la massimizzazione dei lavori svolti per unità di tempo).

Un *processo* è un programma in esecuzione e corrisponde a:

1. L’insieme ordinato di stati assunti dal programma nel corso dell’esecuzione (sulla sua macchina virtuale)
2. L’insieme ordinato delle azioni effettuate dal programma nel corso dell’esecuzione (sulla sua macchina virtuale)

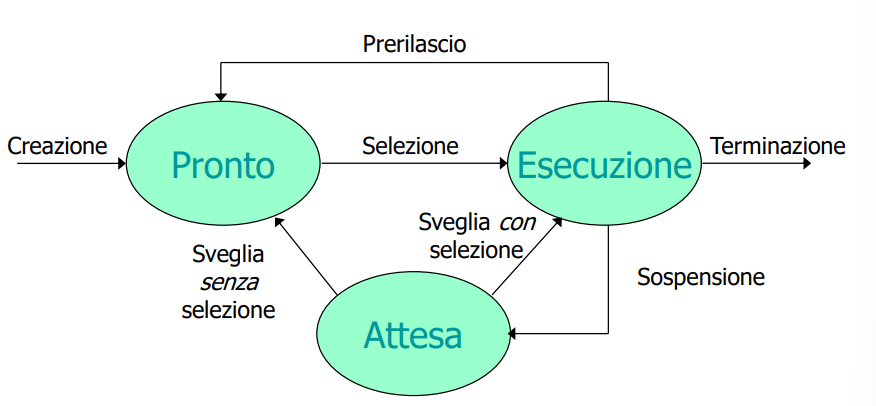
Esso ha un suo spazio di indirizzamento logico, in quanto possiede una parte di memoria della macchina virtuale che può leggere e scrivere, organizzata ad esempio in pagine o segmenti e rappresentano i dati del programma e le sue aree di lavoro.

In un sistema coesistono processi utente e di S/O e possono cooperare tra loro ma hanno privilegi diversi. Avanzano concorrentemente, dato che il S/O assegna loro le risorse necessarie secondo diverse politiche di ordinamento (es. a divisione di tempo/priorità). Spesso hanno bisogno di dover comunicare tra loro e il S/O deve essere in grado di fornire i meccanismi e i servizi necessari.

Un processo può creare processi “figli” (ad esempio, un processo interprete di comandi (shell) lancia un processo figlio per eseguire un comando di utente) ed essi vengono:

* creati per eseguire un lavoro
* sospesi per consentire l’esecuzione di altri processi
* terminati al compimento del lavoro assegnato (un processo figlio che sopravvive alla terminazione del processo padre è detto “orfano” e può essere molto dannoso)

Riporta questi stati di avanzamento:



Il *gestore dei processi* costituisce il cuore o nucleo del S/O (kernel) e gestisce ed assicura l’avanzamento dei processi (ognuno col suo stato di avanzamento come si vede sopra) e la scelta del processo da eseguire ad un dato istante si chiama ordinamento (scheduling).

Il gestore decide il cambio di stato dei processi, in particolare:

* per divisione di tempo
* per trattamento di eventi (es. risorsa libera/occupata

Esso inoltre ha il compito di gestire il nucleo del S/O e di:

* gestire l’avanzamento dei processi, registrando ogni transizione nel loro stato di attivazione
* gestire le interruzioni esterne (all’esecuzione corrente) causate da eventi di I/O o situazioni anomale rilevate da altri processi o componenti del S/O
* consentire ai processi di accedere a risorse di sistema e di attendere eventi

La politica di ordinamento deve essere equa (fair → fairness), perché i processi pronti per eseguire devono avere l'opportunità di farlo e processi in attesa di risorse devono avere l'opportunità di accederle.

I meccanismi e servizi di comunicazione e sincronizzazione devono essere efficaci, in quanto il dato (o segnale) inviato da un processo mittente deve raggiungere il destinatario in un tempo breve e in modo sicuro.

Viene definita *risorsa* un qualsiasi elemento fisico (hardware) o logico (realizzato a software) necessario alla creazione, esecuzione e avanzamento di processi.

Le risorse possono essere

– Durevoli (es., CPU)

– Consumabili (es., memoria fisica)

– Ad accesso divisibile o indivisibile

• Divisibile se tollera alternanza con accessi di altri processi

• Indivisibili se non tollera alternanza durante l’uso

– Ad accesso individuale o molteplice

• Molteplicità fisica o logica (virtualizzata)