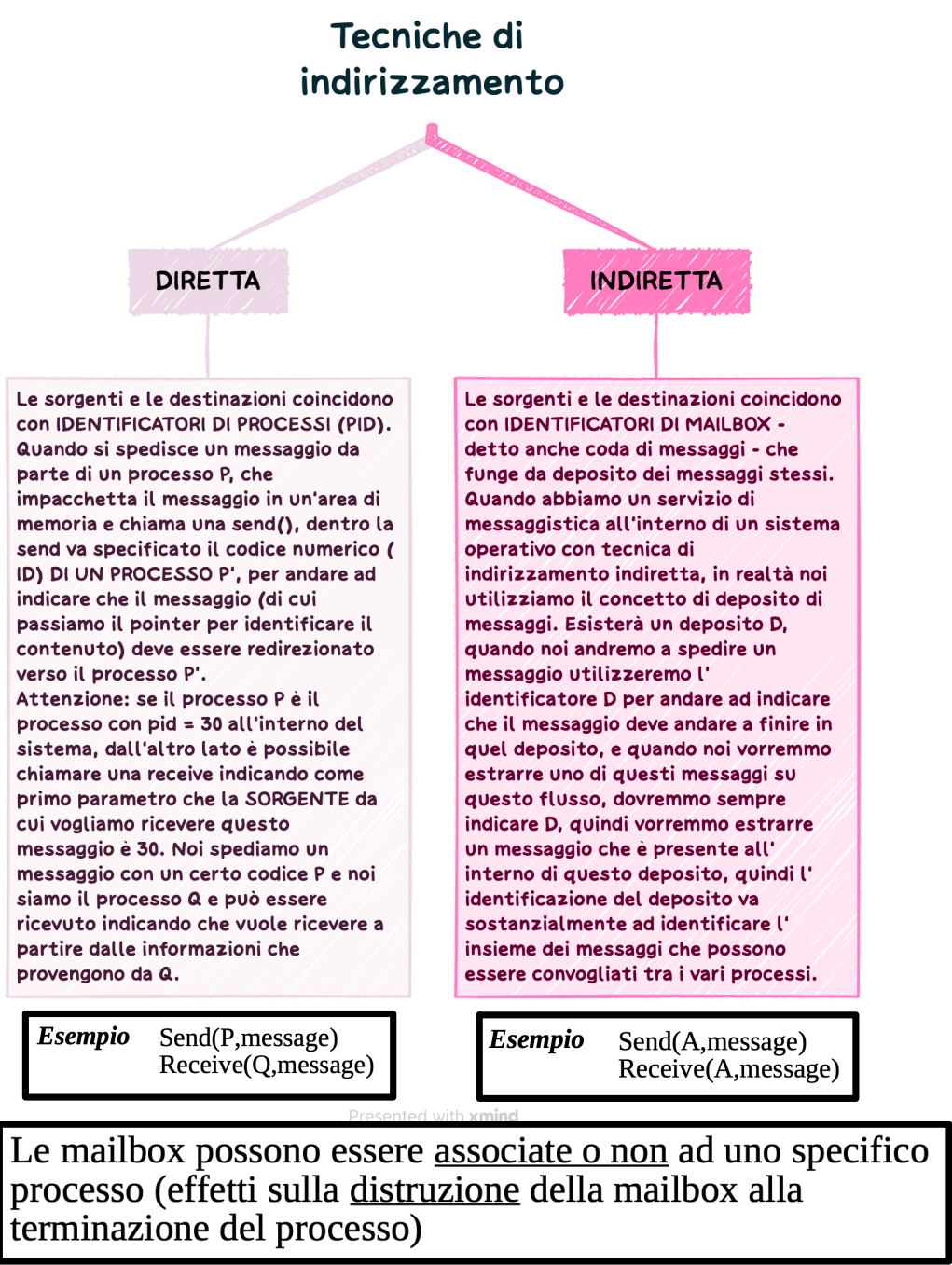


Tecniche di indirizzamento

Abbiamo sin ora parlato nel dettaglio di qual è il messaggio che dobbiamo spedire o qual è l'area dove dobbiamo ricevere il messaggio, in particolare andiamo ad utilizzare un pointer che va a specificare l'area in cui abbiamo impacchettato i dati da spedire o vogliamo che i dati ci vengano consegnati su una ricezione. Sia sulla destinazione, sia sulla sorgente, siamo rimasti vaghi.

Quali sono i modi in cui su un sistema operativo possiamo specificare una destinazione per un messaggio o una sorgente da cui vogliamo prelevare un messaggio? Ci sono due modi classici: Uno si chiama "TECNICA DIRETTA DI INDIRIZZAMENTO" e l'altro "TECNICA INDIRETTA DI INDIRIZZAMENTO".



Il deposito viene ad essere rimosso quando il processo creatore termina la sua esecuzione. Dato un deposito di messaggio D questo è associato ad uno specifico processo P creatore, e quando questo processo P termina, questo deposito viene ad essere eliminato.

Ma possiamo anche non avere questa situazione, magari è possibile che il deposito venga eliminato quando tutti i processi che possono arrivare su di esso - quindi che hanno un riferimento valido - vengono a terminare, quindi in questo caso il deposito viene ad essere rimosso, e quindi i messaggi depositati al suo interno vengono ad essere persi.

Un deposito di messaggio, indipendentemente se il suo processo creatore viene a terminare, può continuare ad esistere, e questa esistente può essere indipendente dalle attività di qualsiasi altro processo all'interno del sistema.

Quando un indirizzamento è indiretto questo ci permette di realizzare una relazione "UNO A MOLTI" E "MOLTI A MOLTI". Cosa che non possiamo fare utilizzando le tecniche dirette. In particolare quando noi utilizziamo una tecnica diretta spediamo delle informazioni per un'unica destinazione, mentre invece quando utilizziamo un deposito le informazioni che andiamo a spedire possono essere poi estratte eventualmente da "MOLTI", IN PARTICOLARE TRA UNO DI MOLTI PROCESSI O THREAD. Quindi in qualche modo andiamo a consegnare dei dati in un deposito, chiunque può accedere a quel deposito può essere il destinatario. Il concetto di deposito ci permette di realizzare una relazione "molti a molti", possiamo avere un thread M che deposita un messaggio, un altro thread M' che deposita un altro messaggio, e dall'altro lato possiamo avere un thread che estrae M ed uno che estrae M'.

