

Messaggi nei sistemi operativi

Un messaggio è una serie di informazioni che noi possiamo depositare ovunque, in particolare su un file, pipe, named pipe e così via...! Però attenzione perché i messaggi all'interno dei sistemi operativi sono implementati anche secondo dei servizi di scambio dei messaggi, per cui quello che noi abbiamo realmente è esattamente la seguente cosa:

Noi possiamo avere un thread che chiama un servizio che ci permette di andare ad inoltrare un messaggio, attenzione, non andare a registrare quel messaggio su una named pipe o su una pipe o su un file, utilizzando un servizio S di SCAMBIO MESSAGGI. Quindi un servizio particolare per andare a poter immettere messaggi che possono essere estratti da un altro thread che chiama un servizio del kernel S' per andare a farsi consegnare questo messaggio.

Questi servizi S e S' rappresentano un sottosistema di scambio dei messaggi che i kernel dei sistemi operativo ovviamente offrono.

Un messaggio non è altro che un'unità di dati che può essere scambiata tra processi distinti. La spedizione di un messaggio avviene in maniera "ATOMICA": Il messaggio non è altro che un'unità informativa.

Se il messaggio è formato da N byte, o né depositiamo N, oppure non né depositiamo nessuno!

Mentre invece su uno stream noi sappiamo che ci possono essere dei residui, noi possiamo scrivere alcune informazioni su uno stream e queste informazioni vengono scritte solo parzialmente - chiamiamo ad esempio una write unix e questa scriverà una parte delle informazioni che noi volevamo emettere in output - ma nei servizi per lo scambio dei messaggi noi lavoriamo in modo atomico, quindi o il servizio prende tutto il messaggio oppure nessuno.

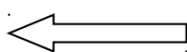
Quindi anche la ricezione (estrazione) di un messaggio avviene in maniera atomica.

- sono unità di dati che possono essere scambiate tra due o più processi tramite servizi di I/O
- il deposito (spedizione) di un messaggio avviene in modo "atomico"
- l'estrazione (ricezione) di un messaggio avviene anche essa in modo atomico
- due o più processi non possono quindi scambiarsi frazioni di un messaggio
- non necessariamente i messaggi hanno la stessa taglia

Primitive di spedizione

Andiamo a vedere qual è il servizio classico che noi possiamo utilizzare su un sistema operativo per andare a spedire messaggi verso una certa destinazione: il servizio è il "Send" e abbiamo due parametri interessanti da considerare: il primo è la specifica della destinazione e il secondo è l'informazione che indica il contenuto del messaggio, tipicamente un pointer che va a specificare un'area di memoria dove noi abbiamo assemblato questo contenuto del messaggio che deve essere inoltrato.

Send(destinazione, messaggio)



Modello generale

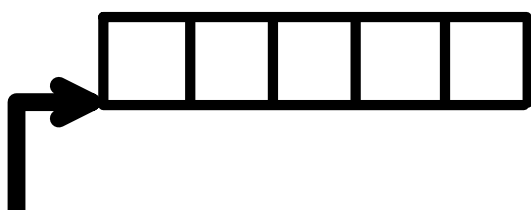
Ci sono delle modalità differenti con cui, tipicamente a livello sistema, questa send può essere eseguita.

Quindi noi chiamiamo una "send", passiamo il controllo al software del sistema, e poi il sistema ovviamente svolgerà questa attività a seconda di varie modalità possibili.

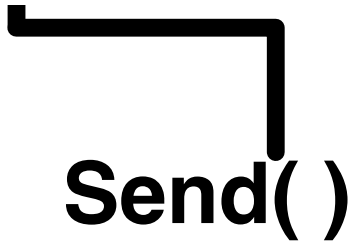


Sincrona: Noi chiamiamo questa spedizione, quindi chiamiamo questo servizio di sistema, il controllo non ci viene ritornato fino a che il buffer che contiene il messaggio NON può essere sovrascritto senza arrecare danni al messaggio stesso.

Abbiamo un area di memoria dove abbiamo scritto le informazioni che rappresenteranno il messaggio, chiamiamo una send sincrona, la send va a specificare quest'area di memoria, quando la send ritorna noi possiamo andare a sovrascrivere quest'area di memoria senza aver danneggiato il messaggio che abbiamo spedito.



Quando la send ritorna noi possiamo andare a



sovrascrivere quell'area di memoria.

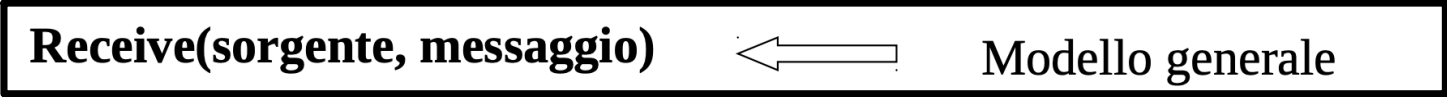
Il che implica dire che questo messaggio è già "SAFE": sicuro verso la destinazione. Una send è sincrona quando nel momento in cui riprendiamo il controllo siamo tranquilli che, la gestione del contenuto che noi avevamo definito come contenuto del messaggio da inoltrare, è già stata fatta. Quindi possiamo riscrivere le nostre aree di memoria dove originariamente avevamo impaccato quel contenuto.

Sincrona Rendez-Vous: Questa spedizione non ritorna il controllo fino a che il messaggio non è stato ricevuto. A livello sistema vengono usati meccanismi di acknowledgment. Il thread che chiama la spedizione chiama questo servizio del kernel del S.O, viene messo in stato di blocco, e verrà ad essere rimesso in stato ready per riprendere il controllo della CPU quando verrà schedulato in CPU, ovviamente per effetto di un arrivo di un acknowledgment, qualche altro thread ha chiamato il kernel - in particolare il servizio di ricezione di messaggio - per andare a ricevere il messaggio M che era stato prodotto per effetto della spedizione. La spedizione ha prodotto il messaggio e ha messo il thread in attesa, in attesa dell'arrivo di un altro thread che logicamente va a proporre un acknowledgment ad indicare che quel messaggio *è stato effettivamente ricevuto*.

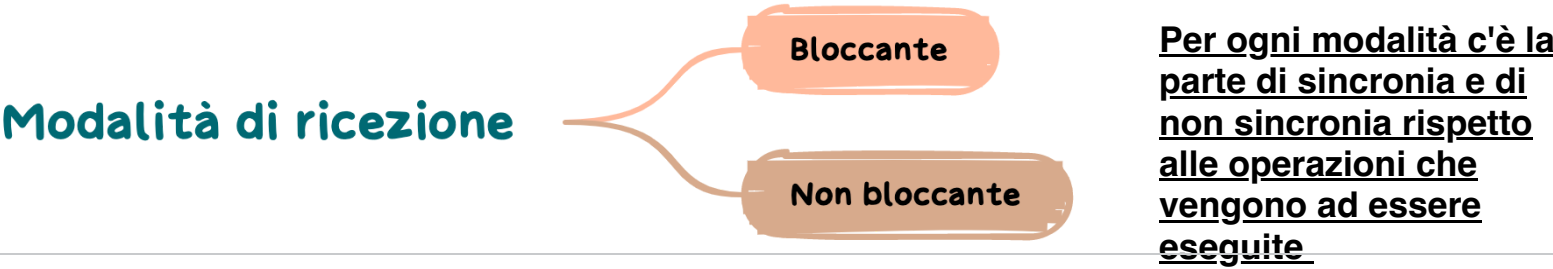
Asincrona: Quando abbiamo una send asincrona, noi chiamiamo una send e andiamo ad indicare che il messaggio è presente all'interno di una certa area di memoria, quindi stiamo parlando di ciò che avevamo introdotto prima. Avevamo introdotto i dati nell'area di memoria, chiamiamo una send e tra i vari parametri indichiamo che l'area che contiene il messaggio è esattamente quella, quando il software del sistema prende il controllo saprà dove è presente il messaggio, MA IL CONTROLLO AL CHIAMANTE DELLA "SEND" VIENE RESTITUITO SENZA AVERE CHE IL CONTENUTO DEL MESSAGGIO È STATO REALMENTE GIÀ PRESO IN CARICO DAL SISTEMA. Il sistema ha preso in carico la posizione del messaggio all'interno della nostra area di memoria, sa dov'è il messaggio, ma non ha preso il contenuto del messaggio. Ha preso soltanto l'indicazione che gli va ad indicare dove il messaggio è presente. Quindi idealmente noi potremmo andare a sovrascrivere il messaggio senza che la versione originale del messaggio sia stata realmente già spedita verso la destinazione. Questa si chiama ASINCRONIA: io vado avanti per fare le mie attività rispetto a quello che eventualmente il sistema farà quando deciderà di farlo. Una send sincrona o sincrona rendez-vous può essere bloccante o non bloccante. Bloccante significa che chiamiamo il servizio di sistema (send) ed è possibile che il thread chiamante viene messo in stato di blocco.

Perché vengo messo in uno stato di blocco su una spedizione sincrona? Se io devo poter riprendere il controllo e quando riprendo il controllo posso essere tranquillo che l'area di memoria in cui avevo inserito il mio messaggio può essere sovrascritta senza che questo messaggio sia danneggiato, evidentemente di questo messaggio ne è stata fatta una COPIA da parte del software del sistema, ma per fare una copia dobbiamo avere memoria a disposizione, se questa memoria non c'è ovviamente vengo messo correntemente nello stato di blocco. Il che implica dire che il thread chiamante di questa spedizione viene messo in stato di blocco fino a che eventualmente il messaggio che lui voleva spedire non è stato realmente copiato, e quindi l'area di memoria può essere tranquillamente riutilizzata. Una spedizione non bloccante e nel caso in cui non ci sia memoria otterremo tipicamente un errore da parte di questa API. L'API ci andrà ad indicare che il controllo viene ritornato senza aver avuto blocco di questo thread che ha chiamato questo servizio di sistema, ma ovviamente il servizio di sistema può ritornare errore, la stessa cosa può capitare per una send sincrona "rendez-vous". Sulla send asincrona abbiamo l'operatività non bloccante: il thread chiamante chiama il sistema per spedire un messaggio comunicando un indirizzo che è l'indirizzo di dove è presente il contenuto del messaggio nell'address space, e poi il controllo TORNA SUBITO a quel thread, e quindi segue una regola non bloccante. La spedizione verrà fatta più avanti, quindi il sistema ha solo registrato dove è presente quel messaggio.

Primitive di ricezione



Questo è il servizio che ci permette di poter ricevere un messaggio. I parametri sono tipicamente la SORGENTE da cui vogliamo acquisire il messaggio, e ovviamente dobbiamo andare ad indicare qual è l'area di memoria dove noi vogliamo ricevere questo messaggio, quindi forniamo informazioni a riguardo sul dove consegnarlo. Andiamo a vedere le modalità che abbiamo su una receive.



Quando una receive è **BLOCCANTE** il processo o thread che sta chiamando questa operazione di receive (thread se siamo in un ambiente multithreading) va in stato di wait - quindi in uno stato di blocco all'interno del diagramma a stati - fino a che non arriva un messaggio dalla sorgente specificata, quindi da quella specifica sorgente. Una receive bloccante ci dice che quel thread rimarrà in attesa effettiva che quel dato che sta cercando di ricevere sia disponibile.

Ovviamente una receive NON BLOCCANTE è esattamente il duale. Il processo o thread non va in stato di wait neanche nel caso in cui il messaggio da quella sorgente non sia disponibile.

Il che implica dire che quando noi chiamiamo una receive non bloccante essa ci può ritornare il controllo senza averci portato in stato di blocco, indicandoci eventualmente un errore di *non presenza del messaggio* che stiamo cercando da quella specifica sorgente.

Ovviamente poi possiamo avere SINCRONIA E NON in entrambi i casi: Sincronia significa che l'operazione di receive viene ad essere eseguita dal sistema in maniera "sincrona" rispetto alle attività del chiamante, quindi se una receive è sincrona, quando noi riprendiamo il controllo, essa è già avvenuta e quindi all'interno dell'area di memoria "messaggio" di cui noi passiamo il pointer abbiamo già il contenuto del messaggio che il sistema ci ha consegnato sincronamente rispetto alla nostra attività.

Mentre invece l'elaborazione NON SINCRONA implica dire che noi chiamiamo una receive, e quindi comunichiamo al sistema anche di dove vogliamo ricevere il messaggio, ossia qual è l'area di memoria dove il messaggio ci deve essere riconsegnato, E RIPRENDIAMO IL CONTROLLO, senza che il messaggio sia consegnato già quando noi riprendiamo il controllo.

Il che implica dire che la consegna avverrà in maniera asincrona rispetto alle attività che noi stiamo facendo nel tempo.

Una receive sincrona è una receive in cui tipicamente possiamo lavorare secondo uno schema bloccante anche in relazione all'assenza di dati da quella specifica sorgente.

Mentre tipicamente possiamo avere uno scenario in cui siamo non bloccanti e possiamo lavorare in maniera sincrona e il che implica dire che io chiedo di ricevere un messaggio in modo non bloccante, se non c'è non voglio aspettare, ma se c'è eventualmente mi va consegnato prima che io riprendo il controllo. Quindi l'operazione di consegna di questo messaggio deve essere eseguita dal software del sistema secondo uno schema sincrónico rispetto a quello che eventualmente io sto facendo.

Send(destinazione, messaggio) ← Modello generale	Receive(sorgente, messaggio) ← Modello generale
Modalità 1. <u>Sincrona</u> : non ritorna il controllo fino a che il buffer contenente il messaggio puo’ essere sovrascritto senza arrecare danni al messaggio appena spedito 2. <u>Sincrona rendez-vous</u> : non ritorna il controllo fino a che il messaggio non viene ricevuto (uso di acknowledgment) 3. <u>Asincrona</u> : possibilita’ di riutilizzo immediato del buffer contenente il messaggio (possibilità di <u>sovrascrittura</u> del messaggio in corso di spedizione)	Modalità 1. <u>Bloccante</u> : processo in wait fino a che non arriva almeno un messaggio dalla sorgente specificata 3. <u>Non-bloccante</u> : il processo non va in wait neanche in caso un messaggio dalla sorgente specificata non sia disponibile
<div><div></div><div></div><div></div></div> <div>bloccante e non</div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <	