

## 4N – TIPSIT MAZZOLA - D'ORSI - 25\_26

### TECNOLOGIE E PROGETTAZIONE di SISTEMI INFORMATICI e di TELECOMUNICAZIONI

1. MODELLO A PROCESSI: DEFINIZIONI E STRUMENTI DI BASE
  - a. stati di un processo
  - b. interrupt
  - c. comandi per i processi:
    - i. creazione
    - ii. sospensione
    - iii. terminazione
  - d. PCB
2. RISORSE E CONDIVISIONE
  - a. definizioni generali
  - b. classificazioni:
    - i. delle richieste
    - ii. dell'assegnazione
    - iii. delle risorse
  - c. grafo di Holt
3. LABORATORIO: implementazione in C della programmazione concorrente
  - a. macchina astratta e macchina parallela
  - b. *fork()*: la creazione di un processo
  - c. *exit()*: terminazione di un processo
  - d. *getpid()* e *getppid()*: identificativo processo corrente e processo padre
  - e. *sleep()*: attesa di un processo – per evitare processi orfani
  - f. Più Fork in sequenza e più Fork annidate
  - g. gestione di una esecuzione deterministica
    - i. *wait()* e *waitpid()*
  - h. funzioni predefinite per rilevare lo stato di un processo
4. I THREAD
  - a. definizioni di base
  - b. tipologie di sistemi multithreading
  - c. stati ed utilizzi dei thread
  - d. thread POSIX
5. LABORATORIO: implementazione in C della programmazione concorrente
  - a. *pthread\_create()*: creazione di un thread
  - b. *&codice\_thread*: per inserire il codice della funzione thread
  - c. *pthread\_exit()*: terminazione di un thread
  - d. *pthread\_join()*: attesa di un thread
  - e. esercizi esemplificativi per passare parametri ad un thread
    - i. passaggio diretto

1. dal 'create' al 'thread' e viceversa

ii. passaggio indiretto mediante condivisione di memoria

## 6. ELABORAZIONE CONCORRENTE E SUE DEFINIZIONI

- a. grafo delle precedenze nei processi non sequenziali
- b. scomposizione dei processi non sequenziali
- c. linguaggi di programmazione concorrente di alto livello (non sequenziali)

## 7. COMUNICAZIONE E SINCRONIZZAZIONE

- a. la comunicazione tra processi
  - i. modello a memoria comune e a scambio di messaggi
  - ii. problemi di:
    - 1. allocazione delle risorse
    - 2. competizione/cooperazione/interferenza
- b. la sincronizzazione
  - i. overlapping e interliving
  - ii. condizioni di Bernstein
  - iii. sezione critica e mutua esclusione
  - iv. starvation e deadlock
  - v. safety/liveness
- c. i semafori e loro applicazione
  - i. Spinlock – di basso livello
    - 1. Allocazione/rilascio/indivisibilità
  - ii. Dijkstra – a conteggio
    - 1.  $P(S) / V(S)$
  - iii. Applicazioni:
    - 1. Mutua esclusione tra due o più processi
    - 2. Come vincoli di precedenza
    - 3. Rendez-vous
  - iv. problemi tipici:
    - 1. produttori/consumatori
    - 2. lettori/scrittori
    - 3. deadlock

d. i monitor

e. lo scambio di messaggi

## 8. I REQUISITI SOFTWARE

- a. la specifica dei requisiti
- b. raccolta e analisi dei requisiti
- c. attori, casi d'uso e scenari
- d. la documentazione dei requisiti

## 9. DOCUMENTAZIONE DEL SOFTWARE

- a. La documentazione del progetto