Strutture fondamentali di un Sistema Operativo

Process Control Block (PCB) - pseudocodice in java

PCB : DESCRITTORE DI PROCESSO

E' una struttura dati che contiene tutte le informazioni che servono per descrivere un processo

```
class ProcessControlBlock {
    // shared by all instances
    private int pid;                    // IDentificatore del processo
    private int parentsPid;             // Il pid del padre di questo processo
    private int status;                 //  ACTIVE, ZOMBIE, or FREE
    prinate int  Registers[];           // i registri della CPU: PC,SP,etc..
    private Thread myThreads[];         // i thread del processo: new Thread[MAX_THREAD_PER_PROC]
    private int exitStatus;             // il valore ritornato da Sys_Exit
    private int addrSpace[];            // spazio di indirizzi logico: new int[Memory_Size]
    private OpenFile fileDescriptor[];  // file aperti: new OpenFile[MAX_FILES_PER_PROCESS]

    // costruttore
    ProcessControlBlock() {
       pid = ++sharedUniqueId;          // ricava un ID unico
    }
}
```

I registri vengono salvati quando si interrompe l'esecuzione e vengono ripristinati quando viene ripresa

Process Manager - pseudocodice in java

Strutture dati e Metodi per la gestione dei processi

```
class ProcessManager{
     ProcessControlBlock processTable[];
                     //Tabella dei processi: ProcessControlBlock[MAX_NUMBER_OF_PROCESSES]
     Vector<ProcessControlBlock> freeList=new Vector<ProcessControlBlock>();
                      //lista  di PCB liberi

     private ProcessControlBlock aProcessDied;
                      //condizione di processo terminato
     private ProcessControlBlock aPCBbecomesfree;
                      //condizione di un PCB che diventa libero
     ProcessManager () //Costruttore
     GetANewProcess () //ritorna un nuovo ProcessControlBlock
}
```

Ciclo fondamentale di un SO

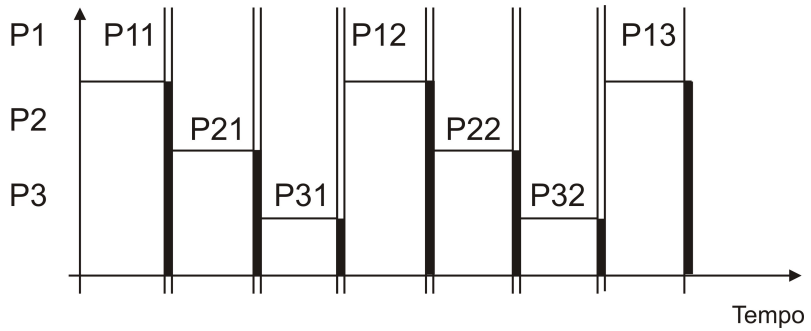
```
while(true){
    scheduler();
    dispatcher();
    ExecutePCB();
    waitforinterrupt();
}
```

I singoli componenti:

- scheduler(): Lo scheduler seleziona il prossimo PCB in base alla data politica di schedulazione. Lo scheduler preleva dal PCB selezionato l'indirizzo della prossima istruzione che deve essere eseguita. Questo indirizzo viene caricato nel PC del processore.
- dispatcher(): Il Dispatcher assegna un processo per la CPU. Il dispatcher estrae i parametri del PCB prima che la CPU esegua il PCB.
- ExecutePCB(): inizia l'esecuzione del PCB da parte della CPU
- waitforinterrupt(): aspetta l'interrupt da timer

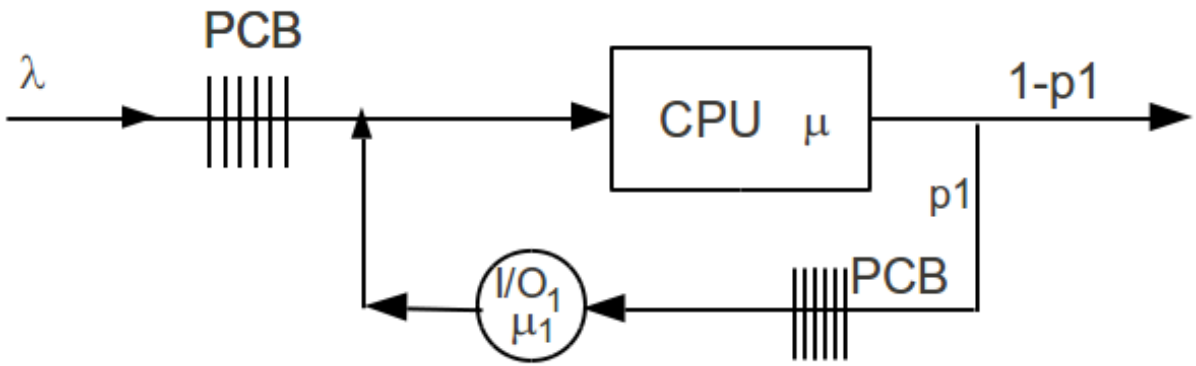
Il Contesto dei processi

- Contesto di un processo = spazio di indirizzamento + contenuto dei registri hardware + strutture dati del kernel che riguardano il processo.
- Context Switch: cambiando processo bisogna cambiare contesto
- Context switch nella schedulazione time sharing



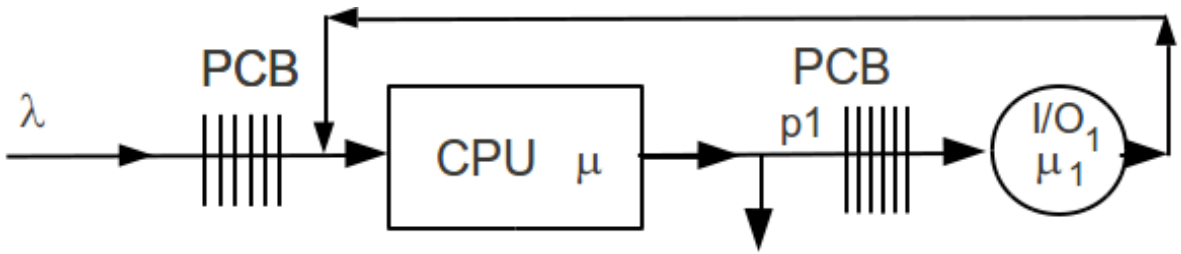
Topologie del SO come reti di code d'attesa

1 CPU, 1 I/O



Topologie dei SO come reti di code d'attesa

Configurazione alternativa (Tandem)



Topologie dei SO come reti di code d'attesa

1 CPU, 2 I/O (configuazione Tandem)