Planificació de Processos

1. Estats d’un Procés

Són 3. Un procés pot canviar d’estat preparat a executant milers de vegades. El SO decideix quan un procés es deixa d’executar i es passa a preparat, i a l’inrevés.

* 1. Preparat: si està disponible per ser executat.
  2. Executant
  3. Bloquejat: espera algun esdeveniment. Quan succeeix, passa a preparat. Es posa a bloquejat quan s’està executant i es realitza una acció com:
     1. Funció de lectura, com llegir dades de xarxa teclat o disc.
     2. S’adorm (sleep) voluntàriament o wait() (Esperar a que el fill acabi).
     3. Canvi de context (executant a preparat o bloquejat) (Gràcies al mode dual)
        1. Per deixar d’executar el SO emmagatzema tota la info del procés al seu BCP (Bloc de control de processos). La info és:
           1. ID del procés
           2. Estat del procés
           3. Privilegis
           4. Quina part de la memòria física resideix, fitxer executable associat.
        2. Restaurar: els valors del seu BCP.
     4. Planificador: algoritme del SO per triar el següent procés a executar.
     5. Va així:
        1. Interrupció de temporitzador
        2. Maquinari interromp l’execució del procés actual. Emmagatzema valors de la CPU a una pos. De memòria.
        3. Es crida a una determinada funció del SO
        4. SO decideix canviar de procés. Es torna la interrupció, maquinari carrega els valors i ale
     6. Quan es produeix un canvi de context?
        1. Interrupció de temporitzador
        2. Altre tipus d’interrupció: arribada de dades de xarxa, haver-se llegit dades de disc...
        3. Al bloquejar-se un procés
  4. Planificador (Algorismes que funcionen bé en un entorn de càrrega variada, és a dir, tant de càrrega de computació alta com processos amb càrrega alta d’entrada-sortida)
  5. Tasca o treball a realitzar
     1. Un procés pot executar varies tasques mentre s’executa, com desar dades a dics, dibuixar a pantalla, etc. Cada fil d’execució d’un procés pot executar diverses tasques.
  6. Round Robin (FIFO Circular)
     1. El planificador disposa d’una llista de tasques i les executa de manera ordenada
     2. Cada tasca s’executa en un temps limitat (llesca de temps). Quan finalitza, el planificador fa un canvi de context cap una altra tasca i posa l’anterior a la última pos.
     3. Llesca de temps: si es massa petita, hi haurà sobrecàrrega de planificació, és a dir, es perdrà molt de temps fen canvis de context i les tasques no podran fer la seva feina. Si és molt alta, les tasques hauran d’esperar massa a obtenir el seu torn. Entre 10 i 100 ms.
     4. No és adequat en un entorn de càrrega variada. Tasques amb càrrega alta en termes d’entrada – sortida necessiten poc temps de CPU per emetre la següent operació d’entrada – sortida.
     5. Va bé quan totes les tasques són CPU Bond, és adir, alta carrega computacional
  7. Max-min fairness
     1. Assignar en cada instant de temps la tasca a la CPU que ha rebut menys temps de CPU
     2. Recursos assignat en ordre creixent de demanda
     3. Si són del same tipus, es converteix en Round Robin, cada tasca obté la mateixa assignació de processador
     4. Com funciona idealment?
        1. Cada cop que el planificador pot escollir una tasca a executar, escull aquella que té fins aquell moment el menor temps acumulat al processador
        2. Cua de prioritat: alt cost computacional.
  8. Solució: Multi-level feedback Queue (Win, Mac, Linux)
     1. Max-min fairness amb extensió de round robin
     2. Múltiples cues round robin: cada una amb prioritat i un temps determinat
     3. Tasques es mouen pel planificador entre les cues. Si una consumeix la seva llesca de temps, es mou a una cua de planificació més baixa, sempre i quan no hagi accedit a disc
  9. Planificació multiprocessador: cada core amb la seva mem. Cache
     1. Cada core la seva pròpia estructura MFQ
     2. Cada tasca s’executa al mateix processador
     3. Si no tens res, pots ROBAR
  10. Planificació d’estalvi energètic: sistemes amb bateria
      1. Diseny: un core ràpid significa circuits complexes i gastar mes energia. Un de lent és mes senzill i pot reduir consum energètic. Sistemes que inclouen tots dos on es va triant
      2. Ús del processador: és possible apagar un processador o reduir el clock
      3. Dispositius entrada-sortida: apagar pantalles, wifi, etc
  11. Planificació a temps real: assegurarse que diferents tasques es facin en terminis determinats:
      1. Control del vol en un avió, bloqueig de frens, reproducció de vídeo...
      2. Diferents mètodes:
         1. Sobre-aprovisionament: agafar operacions del processador i ferles servir sempre per la tasca, així sexecutaran bé
         2. Priorització per termini (earliest deadline first): tasques de temps real son prioritzades segons el seu termini, sempre i quan siguin computacionals i no d’entrada-sortida
      3. Sobrecàrrega:
         1. Rebutjar peticions de connexió que poden aportar-ne mes: s’asegura un bon servei a les connexions existents: streaming vídeo
         2. Reduir temps de resposta