

Lucrarea de laborator nr 4

Tema lucrării: Elaborarea unui mecanism de planificare a activității sistemului de operare.

Partea teoretică:

Planificare în sisteme prelucrarea datelor în bloc (batch)

Unii algoritmi sunt utilizați atât în sistemele de planificare a datelor în bloc, cât și în sistemele interactive.

Algoritmul primul venit, primul realizat - este cel mai simplu dintre algoritmi de planificare. Sunt furnizate procese care accesează procesorul în ordinea în care îl solicită. Mai des se formează o singură coadă de procese de așteptare. De îndată ce apare primul proces, pornește imediat și rulează atât timp cât este necesar. Procesele rămase sunt plasate la sfârșitul cozii. Când procesul curent este blocat, următorul în linie începe, iar când este eliberat din blocare, procesul ajunge la capătul cozii. Principalul avantaj al acestui algoritm este că este ușor de înțeles și la fel de ușor de programat. În acest algoritm toate procesele aflate în starea de așteptare sunt controlate de o listă înlănțuită. Pentru a selecta un proces de rulat, trebuie doar să luați primul element din listă și apoi ștergeți-l. Apariția unui nou proces face ca acesta să fie plasat la sfârșitul listei.

Algoritmul cea mai scurtă sarcină este prima - este algoritmul care alege din lista de așteptare procesul cu cel mai scurt timp de realizare. Acest algoritm presupune că perioadele de timp de realizare a proceselor sunt cunoscute din timp. Dacă există mai multe procese la fel de importante în coadă, planificatorul selectează mai întâi sarcina cea mai scurtă.

Algoritmul cel mai scurt timp de realizare rămas este algoritmul care selectează de fiecare dată procesul cu cel mai scurt timp de execuție rămas. În acest caz, este necesar să se cunoască în prealabil timpul de execuție a procesului. Când sosește un nou proces, timpul total de execuție al acestuia este comparat cu timpul de execuție rămas al procesului activ. Dacă timpul de execuție al noului proces este mai scurt, procesul curent este suspendat și controlul este transferat noului proces. Această schemă vă permite să deserviți rapid cererile scurte.

Planificare în sisteme interactive

Algoritmul de planificare în ciclu (round-robin). Fiecare proces este furnizat un anumit interval de timp al procesorului, așa-numitul cuant de timp. Dacă procesul încă rulează la sfârșitul cuantului de timp, acesta este încheiat și controlul este transferat către următorul proces. Dacă procesul se blochează sau se termină mai devreme, în acest moment are loc o tranziție de control. Implementarea algoritmului round-robin este simplă. Planificatorul trebuie doar să mențină o listă de procese într-o stare pregătită în formă de ciclu. Când un proces și-a atins limita de timp, este trimis la sfârșitul listei trecând indicatorul la următorul proces.

Algoritmul cu planificare prioritară. Ideea de bază este simplă: fiecărui proces i se atribuie o prioritate, iar controlul este transferat procesului gata cu cea mai mare prioritate. Chiar și pe un computer personal cu un singur utilizator pot avea loc mai multe procese, dintre care unele sunt mai importante decât altele. Demonul responsabil pentru trimiterea e-mailurilor în fundal are o prioritate mai mică decât procesul de afișare a videoclipului live pe ecran.

Pentru a preveni rularea nelimitată a proceselor cu prioritate înaltă, planificatorul poate scădea prioritatea procesului cu fiecare bifă de ceas (adică cu fiecare întrerupere a taimerului). Dacă în rezultat prioritatea curentului proces va fi mai mică decât prioritatea procesului următor, va avea loc o comutare. Este posibil de acordat fiecărui proces un timp maxim de rulare. De îndată ce timpul a expirat, controlul este transferat la următorul proces prioritar. Prioritățile procesului pot fi atribuite static sau dinamic.

Algoritmul cu mai multe rânduri (clase) – este algoritmul cu clase prioritare. Proceselor din clasa cu cea mai mare prioritate li s-a alocat un quantum, proceselor din următoarea clasă li s-au alocat două cuante și proceselor din clasa următoare au fost alocate trei cuante etc. Când procesul a folosit tot timpul alocat lui este mutat cu o clasă mai jos.

Algoritmul cel mai scurt proces este următorul – este un algoritm asemănător cu algoritmul cea mai scurtă sarcină este prima din sisteme prelucrarea datelor în bloc.

Algoritmul cu planificare cinstită – este algoritmul la care sistemul acordă atenție proprietarului procesului înainte de planificare. În acest model, fiecare utilizator primește o parte din procesor, iar planificatorul selectează procesul în conformitate cu acest fapt. Dacă fiecărui utilizator i s-a promis 50% din procesor, atunci va primi 50% din procesor, indiferent de numărul de procese a utilizatorilor.

Partea practică

Sunt date la realizare 4 procese A, B, C, D cu valorile timpului de răspuns (timpul de realizare) sunt 4, 20, 12, 8 de unități.

1. Algoritmul primul venit, primul realizat

Ordinea de realizare A B C D

Durata medie este de 11 unități. $(4+20+12+8)/4 = 11$

Timpul de realizare pentru fiecare proces:

primul proces - în 4 unități,

al doilea - în $4+20=24$ unități,

al treilea – în $4+20+12= 36$ unități,

al patrulea – în $4+20+12+8= 44$ unități.

Timpul mediu de rulare va fi egal cu $(4a + 3b + 2c + d)/4$.

$(4*4+3*20+2*12+8)/4=27$ unități

2. Algoritmul cea mai scurtă sarcină este prima

Ordinea de realizare A D C B

Durata medie este de 11 unități. $(4+8+12+20)/4 = 11$

Timpul de realizare pentru fiecare proces:

primul proces - în 4 unități,

al doilea - în $4+8=12$ unități,

al treilea – în $4+8+12= 24$ unități,

al patrulea – în $4+8+12+20= 44$ unități.

Timpul mediu de rulare va fi egal cu $(4a + 3b + 2c + d)/4$.

$(4*4+3*8+2*12+20)/4=21$ unități

3. Algoritmul cel mai scurt timp de realizare rămas

Poate fi realizat numai cu cuant de timp. $K= 10$.

Ordinea de realizare A D C(2) C B(10) B

Durata medie este de 11 unități. $(4+8+12+20)/4 = 11$

Timpul de realizare pentru fiecare proces:

primul proces - în 4 unități,

al doilea - în $4+8=12$ unități,

al treilea – în $4+8+12= 24$ unități,

al patrulea – în $4+8+12+20= 44$ unități.

Timpul mediu de rulare va fi egal cu $(6a + 5b + 4c + 3d+2e+f)/6$.

$$(6*4+5*8+4*10+3*2+2*10+10)/6=23,33 \text{ unități}$$

4. Algoritmul de planificare în ciclu (round-robin).

Cuantul de timp este $K=13$

Ordinea de realizare A B(7) C D B

Timpul mediu de rulare va fi egal cu $(5b + 4c + 3d+2e+f)/5$.

$$(5*4+4*13+3*12+2*8+7)/5=26,2$$

5. Algoritmul cu planificare prioritară.

Prioritățile proceselor vor fi A-3 B-8 C-5 D-4

Ordinea de realizare B C D A

Timpul mediu de rulare va fi egal cu $(4a + 3b+2c+d)/4$.

$$(4*20+3*12+2*8+4)/4 = 24$$

6. Algoritmul cel mai scurt proces este următorul

Cuantul de timp este $K=8$

Ordinea de realizare A D C(4) C B(12) B(4) B

Timpul mediu de rulare va fi egal cu $(7a+6b + 5c + 4d+3e+2f+g)/7$.

$$(7*4+6*8+5*8+4*4+3*8+2*8+4)/7 = 25,14$$

7. Algoritmul cu planificare cinstită

Sarcina: La realizare sunt date procesele a trei utilizatori.

Primul utilizator primește 50% din timpul procesorului. El transmite la realizare 3 procese:

A cu timpul de realizare 10 unități, B cu timpul de realizare 12 unități, C cu timpul de realizare 7 unități.

Al doilea utilizator primește 25% din timpul procesorului. El transmite la realizare 2 procese:

X cu timpul de realizare 6 unități, Y cu timpul de realizare 10 unități.

Al treilea utilizator primește 25% din timpul procesorului. El transmite la realizare 1 proces:

Z cu timpul de realizare 26 unități.

Cuantul de timp este $K=8$ unități.

Care va fi ordinea de realizare a proceselor pentru algoritmul primul venit primul realizat?

Răspunsul este: A(2)B(4)XZ(18)CAY(2)Z(10)BYZ(2)Z

Care va fi ordinea de realizare a proceselor pentru algoritmul primul venit primul realizat, dacă toți trei utilizatori primesc câte 33% din timpul utilizatorului?

Răspunsul este: A(2)XZ(18)B(4)Y(2)Z(10)AYZ(2)BZ

1. Algoritmul "Primul venit – primul servit" (First Come, First Served – FCFS)

1. **Date de intrare**: A(5), B(9), C(4), D(7). Găsiți timpul mediu de execuție și timpul pentru fiecare proces.

2. **Date de intrare**: P1(6), P2(3), P3(8), P4(2). Găsiți timpul mediu de execuție și timpul pentru fiecare proces.

3. **Date de intrare**: X(2), Y(6), Z(3), W(4). Găsiți timpul mediu de execuție și timpul pentru fiecare proces.

4. **Date de intrare**: T1(10), T2(2), T3(4), T4(6). Găsiți timpul mediu de execuție și timpul pentru fiecare proces.

5. **Date de intrare**: M(7), N(5), O(8), P(3). Găsiți timpul mediu de execuție și timpul pentru fiecare proces.

2. Algoritmul "Cel mai scurt proces primul" (Shortest Job First – SJF)

1. **Date de intrare**: A(7), B(2), C(4), D(1). Găsiți timpul mediu de execuție și timpul pentru fiecare proces.

2. **Date de intrare**: P1(5), P2(3), P3(1), P4(2). Găsiți timpul mediu de execuție și timpul pentru fiecare proces.

3. **Date de intrare**: X(6), Y(4), Z(3), W(2). Găsiți timpul mediu de execuție și timpul pentru

fiecare proces.

4. **Date de intrare**: T1(8), T2(3), T3(6), T4(1). Găsiți timpul mediu de execuție și timpul pentru fiecare proces.

5. **Date de intrare**: M(9), N(5), O(2), P(4). Găsiți timpul mediu de execuție și timpul pentru fiecare proces.

3. Algoritmul "Cel mai scurt timp rămas" (Shortest Remaining Time First – SRTF)

1. **Date de intrare**: A(8), B(4), C(3), D(7). $K = 5$. Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

2. **Date de intrare**: P1(10), P2(5), P3(6), P4(8). $K = 4$. Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

3. **Date de intrare**: X(7), Y(9), Z(5), W(3). $K = 3$. Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

4. **Date de intrare**: T1(11), T2(6), T3(4), T4(8). $K = 6$. Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

5. **Date de intrare**: M(9), N(2), O(8), P(5). $K = 4$. Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

4. Algoritmul "Round Robin" (Round Robin – RR)

1. **Date de intrare**: A(10), B(15), C(20), D(25). $K = 10$. Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

2. **Date de intrare**: P1(12), P2(8), P3(16), P4(6). $K = 7$. Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

3. **Date de intrare**: X(9), Y(14), Z(11), W(13). $K = 5$. Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

4. **Date de intrare**: T1(20), T2(12), T3(9), T4(16). $K = 8$. Găsiți timpul mediu de execuție și

ordinea de execuție a proceselor.

5. **Date de intrare**: M(18), N(22), O(13), P(17). $K = 9$. Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

5. Algoritmul "Planificare pe bază de prioritate" (Priority Scheduling)

1. **Date de intrare**: A(5, 1), B(9, 3), C(6, 2), D(8, 4). Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

2. **Date de intrare**: P1(10, 2), P2(4, 1), P3(8, 3), P4(6, 4). Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

3. **Date de intrare**: X(12, 2), Y(7, 1), Z(5, 3), W(9, 4). Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

4. **Date de intrare**: T1(11, 3), T2(6, 1), T3(9, 2), T4(8, 4). Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

5. **Date de intrare**: M(13, 4), N(10, 3), O(7, 2), P(5, 1). Găsiți timpul mediu de execuție și ordinea de execuție a proceselor.

6. Probleme pentru algoritmul "Cel mai scurt proces este următorul" (Cuantul de timp este $K=8$)

1. **Date de intrare**: 4 procese cu timpii de execuție: A = 5, B = 12, C = 7, D = 9. Găsiți ordinea de execuție și timpul mediu de execuție.

2. **Date de intrare**: 5 procese cu timpii de execuție: A = 15, B = 3, C = 8, D = 20, E = 4. Găsiți ordinea de execuție și timpul mediu de execuție.

3. **Date de intrare**: 4 procese cu timpii de execuție: A = 10, B = 8, C = 6, D = 2. Găsiți ordinea de execuție și timpul mediu de execuție.

4. **Date de intrare**: 3 procese cu timpii de execuție: A = 6, B = 16, C = 8. Găsiți ordinea de

execuție și timpul mediu de execuție.

5. **Date de intrare**: 4 procese cu timpii de execuție: $A = 3$, $B = 9$, $C = 14$, $D = 6$. Găsiți ordinea de execuție și timpul mediu de execuție.

7. Probleme pentru algoritmul "Planificare pe bază de echitate" (Cuantul de timp este $K=8$)

1. **Date de intrare**: Trei utilizatori. Primul utilizator primește 33% din timpul procesorului și are procesele $A(12)$, $B(8)$. Al doilea utilizator primește 33% din timp și are procesele $X(10)$, $Y(6)$. Al treilea utilizator primește 33% din timp și are procesul $Z(16)$. Care va fi ordinea de execuție a proceselor?

2. **Date de intrare**: Trei utilizatori. Primul utilizator primește 50% din timpul procesorului și are procesele $A(10)$, $B(5)$. Al doilea utilizator primește 25% din timp și are procesele $X(7)$, $Y(3)$. Al treilea utilizator primește 25% din timp și are procesul $Z(12)$. Care va fi ordinea de execuție a proceselor?

3. **Date de intrare**: Trei utilizatori. Primul utilizator primește 33% din timpul procesorului și are procesele $A(8)$, $B(6)$, $C(5)$. Al doilea utilizator primește 33% din timp și are procesele $X(12)$, $Y(10)$. Al treilea utilizator primește 33% din timp și are procesul $Z(20)$. Care va fi ordinea de execuție a proceselor?

4. **Date de intrare**: Patru utilizatori. Primul utilizator primește 40% din timpul procesorului și are procesele $A(15)$, $B(10)$, $C(8)$. Al doilea utilizator primește 20% din timp și are procesele $X(6)$, $Y(5)$. Al treilea utilizator primește 20% din timp și are procesul $Z(18)$. Al patrulea utilizator primește 20% din timp și are procesul $Q(14)$. Care va fi ordinea de execuție a proceselor?

5. **Date de intrare**: Trei utilizatori. Primul utilizator primește 50% din timpul procesorului și are procesele $A(10)$, $B(7)$, $C(6)$. Al doilea utilizator primește 25% din timp și are procesul $X(10)$. Al treilea utilizator primește 25% din timp și are procesul $Z(14)$. Care va fi ordinea de execuție a proceselor?