

ДИСЦИПЛИНА

Операционные системы

(полное наименование дисциплины без сокращений)

ИНСТИТУТ

Институт информационных технологий

КАФЕДРА

информационных технологий в атомной энергетике

(полное наименование кафедры)

ВИД УЧЕБНОГО

Лекция

МАТЕРИАЛА

(в соответствии с пп 1-11)

ПРЕПОДАВАТЕЛЬ

Пугачев Андрей Васильевич

(фамилия, имя, отчество)

СЕМЕСТР

IV семестр 2024 – 2025 учебный год

(указать семестр обучения, учебный год)

Тема № 4: ”Планирование процессов”

«Операционные системы»

МИРЭА – Российский технологический университет

Москва. 2024-2025 у.г.

Определение

Планировщик процессов - программный модуль операционной системы, реализующий политику планирования процессов.

Определение

Политика планирования процессов — алгоритм выбора процессов планировщиком процессов.

Цели планирования

- ▶ максимальная пропускная способность;
- ▶ приемлемое время отклика в интерактивных системах;
- ▶ максимальное использование ресурсов;
- ▶ исключение бесконечного откладывания;
- ▶ минимизация накладных расходов;
- ▶ предсказуемость поведения системы.

Свойства планирования

- ▶ равноправие;
- ▶ предсказуемость;
- ▶ масштабируемость.

Уровни планирования



Типы ограничений для процессов

- ▶ ограничения возможностями процессора;
- ▶ ограничения возможностями устройств ввода-вывода.

Виды планирования по типу ОС

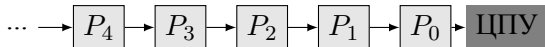
- ▶ планирование в системах пакетной обработки;
- ▶ планирование в интерактивных системах;
- ▶ планирование в системах реального времени.

Планирование в системах пакетной обработки и в интерактивных системах

Очередь (FIFO — first input first output)

Особенности

- + Простота реализации.
- Неэффективное использование ресурсов системы.



Кратчайший процесс - первый (SPF – shortest process first)

Особенности

- + Снижение среднего времени выполнения задачи (T).
- Сложность оценки времени выполнения.

$$T = \frac{\sum_{n=0}^{N-1} t_i + l_i}{N}, \text{ где}$$

t_i – время работы процесса;

l_i - время ожидания процесса.

Снижение среднего времени выполнения задачи

$$l_i = \begin{cases} 0, & \text{при } i = 0 \\ l_{i-1} + t_{i-1}, & \text{при } i > 0 \end{cases}$$

$$N = 3$$

$$T = \frac{\sum_{n=0}^2 t_i + l_i}{3} = \frac{t_0 + (t_1 + t_0) + (t_2 + t_1 + t_1 + t_0)}{3}$$
$$T = \frac{3 \cdot t_0 + 2 \cdot t_1 + t_2}{3} = t_0 + \frac{2}{3} \cdot t_1 + \frac{1}{3} \cdot t_2$$

Сложность оценки времени выполнения

```
#include <time.h>
int func1(void)
{
    int a = 0;
    int i;
    for (i=0; i<7; i++){
        a+=time(NULL);
    }
    return a;
}
```

```
#include <time.h>
int func2(void)
{
    int a = 0;
    a+=time(NULL);
    a+=time(NULL);
    a+=time(NULL);
    a+=time(NULL);
    a+=time(NULL);
    a+=time(NULL);
    a+=time(NULL);
    return a;
}
```

Сложность оценки времени выполнения

```
..... | ;*****  
..... | ; function func2 (global)  
..... | ;*****  
..... | func2:  
..... |     push     ebp  
8048455 |     mov      ebp, esp  
8048457 |     sub      esp, 28h  
804845a |     mov      dword ptr [ebp-0ch], 0  
8048461 |     mov      dword ptr [esp], 0  
8048468 |     call     time  
804846d |     add      [ebp-0ch], eax  
8048470 |     mov      dword ptr [esp], 0  
8048477 |     call     time  
804847c |     add      [ebp-0ch], eax  
804847f |     mov      dword ptr [esp], 0  
8048486 |     call     time  
804848b |     add      [ebp-0ch], eax  
804848e |     mov      dword ptr [esp], 0  
8048495 |     call     time  
804849a |     add      [ebp-0ch], eax  
804849d |     mov      dword ptr [esp], 0  
80484a4 |     call     time  
80484a9 |     add      [ebp-0ch], eax  
80484ac |     mov      dword ptr [esp], 0  
80484b3 |     call     time  
80484b8 |     add      [ebp-0ch], eax  
80484bb |     mov      dword ptr [esp], 0  
80484c2 |     call     time  
80484c7 |     add      [ebp-0ch], eax  
80484ca |     mov      eax, [ebp-0ch]  
80484cd |     leave  
80484ce |     ret  
80484cf |     nop  
80484d0 |  
..... |  
..... | ;*****  
..... | ; function func1 (global)  
..... | ;*****  
..... | func1:  
..... |     push     ebp  
8048421 |     mov      ebp, esp  
8048423 |     sub      esp, 28h  
8048426 |     mov      dword ptr [ebp-0ch], 0  
804842d |     mov      dword ptr [ebp-10h], 0  
8048434 |     jmp      loc_8048449  
8048436 |  
..... | loc_8048436:  
..... |     mov      dword ptr [esp], 0  
804843d |     call     time  
8048442 |     add      [ebp-0ch], eax  
8048445 |     add      dword ptr [ebp-10h], 1  
8048449 |  
..... | loc_8048449:  
..... |     cmp      dword ptr [ebp-10h], 6  
804844d |     jng      loc_8048436  
804844f |     mov      eax, [ebp-0ch]  
8048452 |     leave  
8048453 |     ret  
8048454 |
```

Вытеснение - прерывание выполнения процесса, с целью постановки на выполнения какого-либо иного процесса.

- ▶ приоритетное вытеснение;
- ▶ кооперативное вытеснение.

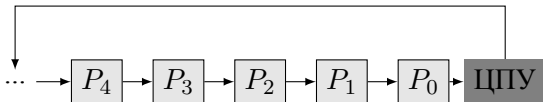
Системный таймер

Квант (времени выполнения) — временной промежуток выделяемый процессу для использования ресурсов процессора.

Циклическая очередь (RR — round robin)

Особенности

- + Простота реализации.
- Неэффективное использование ресурсов системы.



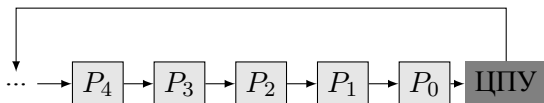
Наименьшее оставшееся время (SRT – shortest remaining time)

Алгоритм аналогичен алгоритму SPF, однако организован на основании циклической очереди.

Эгоистическое планирование (SRR – selfish round robin)

Особенности

- + Простота реализации.
- + Возможность влияния на процесс планирования.
- Неэффективное использование ресурсов системы.



$$F(i) = \begin{cases} a_i < a_{i+1} & \text{– процесс вытесняется} \\ a_i \geq a_{i+1} & \text{– процесс выполняется} \end{cases}$$

где, a_i – “индекс старения” процесса P_i .

Относительное время реакции (HRRN – highest response ratio next)

Особенности

- + Высокая интерактивность.
- Большая задержка для процессов, ограниченных ресурсами процессора.

$$P_i = \frac{t_i + l_i}{t_i}, \text{ где}$$

t_i – время работы процесса;

l_i - время ожидания процесса.

Масштабируемость алгоритмов планирования

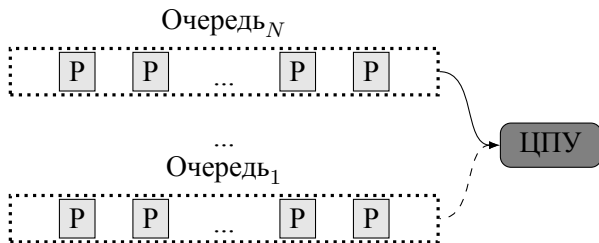
Сложность алгоритма

$$f(x) \approx O(g(x)) \Rightarrow \exists C \in \mathbb{N} : f(x) \leq C \cdot g(x)$$

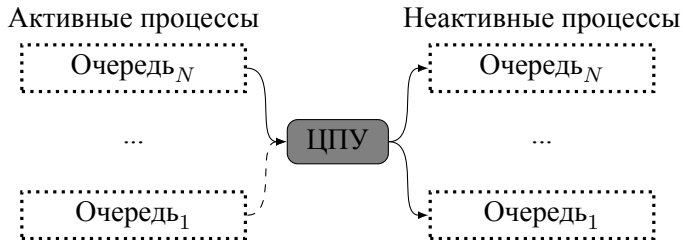
Реализация	Поиск	Вставка	Удаление
Список	$O(n)$	$O(1)$	$O(1)$
Дерево	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$

Многоуровневая очередь с обратной связью

MFQ – multilevel feedback queue



Планировщик Linux ($O(1)$)



Планирование в системах реального времени

Типы алгоритмов планирования

- ▶ мягкое планирование реального времени;
- ▶ жесткое планирование реального времени.

Типа процессов реального времени

- ▶ периодические процессы;
- ▶ асинхронные процессы.

Классификация алгоритмов



<https://cyberleninka.ru/article/n/metody-planirovaniya-vypolneniya-zadach-v-sistemah-realnogo-vremeni>

Вопросы?