KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN MẠNG MÁY TÍNH VÀ TT DỮ LIỆU

Hướng dẫn Lab 6.2 – Cài đặt lập lịch CPU

Trong lập lịch CPU, các yếu tố được xem xét và Bộ lập lịch sẽ dựa vào độ ưu tiên, quy tắc xoay vòng hay FIFO để vừa đảm bảo hiệu năng vừa không mất tính công bằng cho các tiến trình. Trong LAB này, một số tình huống lập lịch được giới thiệu.

Mục tiêu	**Lý thuyết liên quan**	**Tài nguyên**
Tiêu chí định thời CPU	Ch5.3: Scheduling Algorithms 5.7.1 Example: Linux Scheduling	https://github.com/Trantin84/LAB_IntroOS (mã nguồn ví dụ). Sử dụng image Ubuntu 16 / 18
Giải thuật định thời CPU		
Định thời trong POSIX		

Yêu cầu sinh viên: Hiểu và thực thi các đoạn mã đã cung cấp. Áp dụng cho các bài toán liên lạc giữa các tiến trình.

Đánh giá sinh viên: Hỏi đáp các vấn đề lý thuyết. Kỹ năng thực hành. Bài tập.

Yêu cầu nộp bài: các tập tin mã nguồn .c và tập tin khả thực thi .out của các Yêu cầu trong buổi thực hành và Bài tập cuối hướng dẫn trong thời gian cho phép của giảng viên.

Preferences

[1] Abraham Silberschatz, Peter B. Galvin, Greg Gagne, [2018], Operating System Concepts, 10th edition, John Wiley & Sons, New Jersey.

Programming Problems of Chapter 4.

[2] Linux manual page, [2021], on sched(7)

Access https://man7.org/linux/man-pages/man7/sched.7.html

Yêu cầu 1: Thí nghiệm về khả năng Cân bằng tải (Load Balancing) của Hệ điều hành.

Hướng dẫn: Tải về tập tin ***LAB_IntroOS/LAB_6/task2_1.c***, biên dịch và thực thi. Mở ứng dụng "System Monitor" và quan sát. Đây là bài toán ước lượng số Pi trong Lab 6.1 nhưng không chỉ định số tiểu trình mà mặc định là 3. Số điểm mỗi tiểu trình cần sinh ra được truyền vào qua lời gọi. Trong lời gọi sau, có 10,000 điểm được sinh ra ở mỗi tiểu trình.

```
$gcc –o task2_1.out task2_1.c
$sudo ./task2_1.out 10000
```

- Dòng 38, 39 để chỉ định toàn bộ tiểu trình sẽ chạy trên luồng CPU #0. Hãy thực thi và đánh giá thứ tự các tiểu trình, thời gian hoàn thành, số nhân CPU tham gia tính toán giữa hai trường hợp: Gán ái lực (dòng 38, 39 được chạy) và Mặc định (xoá bỏ dòng 38 và 39 đi).

- Dòng 46 – 50 (tương ứng cho dòng 52 – 56 và 58 – 62) là thiết lập giá trị ưu tiên = 50, định thời FCFS rồi gán vào attr[0] chính là đối số thứ hai trong lời gọi tạo tiểu trình pthread_create() sau đó.

- Dòng 74 huỷ bỏ đối tượng thuộc tính.

- Dòng 95 là thu thập thông tin định thời của tiểu trình, kết quả trả về cho một số nguyên khai báo ở dòng 93 (SCHED_OTHER, SCHED_IDLE, SCHED_BATCH, SCHED_FIFO, SCHED_RR)

```
38    CPU_SET(0, & set);
39    sched_setaffinity(getpid(), sizeof(set), & set)
44      struct sched_param shparam;
45      pthread_attr_t attr[3];

46      shparam.sched_priority = 50;
47      pthread_attr_init( & attr[0]);
48      pthread_attr_setschedpolicy( & attr[0], SCHED_FIFO);
49      pthread_attr_setschedparam( & attr[0], & shparam);
50      pthread_attr_setinheritsched( & attr[0],
      PTHREAD_EXPLICIT_SCHED);

74        pthread_attr_destroy( & attr[i]);

93      int policy;
94      struct sched_param shparam;
95      pthread_getschedparam(pthread_self(), & policy, &
      shparam);
97      if (policy == SCHED_OTHER)
```

Yêu cầu 2: Thí nghiệm về lập lịch theo ĐỘ ƯU TIÊN.

Hướng dẫn: Chỉnh sửa tập tin ***LAB_IntroOS/LAB_6/task2_1.c*** theo mô tả sau rồi biên dịch và thực thi.

- Dòng 48, 54 và 60 chỉ ra rằng FCFS được áp dụng.

- Dòng 46, 52 và 58 chỉ ra tiểu trình [0] có độ ưu tiên thấp nhất, còn tiểu trình [2] có độ ưu tiên cao nhất.

?	- Ghi lại thời gian bắt đầu và kết thúc của từng tiểu trình. - Các tiểu trình chạy với thứ tự ra sao?

```
38      CPU_SET(0, & set);
39      sched_setaffinity(getpid(), sizeof(set), & set)

46        shparam.sched_priority = 40;
48        pthread_attr_setschedpolicy( & attr[0], SCHED_FIFO);

52        shparam.sched_priority = 50;
54        pthread_attr_setschedpolicy( & attr[1], SCHED_FIFO);

58        shparam.sched_priority = 90;
60        pthread_attr_setschedpolicy( & attr[2], SCHED_FIFO);
```

Yêu cầu 3: Thí nghiệm về lập lịch theo ĐẾN TRƯỚC CHẠY TRƯỚC.

Hướng dẫn: Tương tự Yêu cầu 2, với độ ưu tiên cài đặt như nhau.

Yêu cầu 4: Thí nghiệm về lập lịch theo XOAY VÒNG.

Hướng dẫn: Tương tự Yêu cầu 2, với độ ưu tiên cài đặt như nhau và thuộc tính SCHED_RR áp dụng cho tất cả.

Mở rộng: SV có thể thử nghiệm với nhiều tổ hợp khác, ví dụ như độ ưu tiên khác nhau và RR cho một nhóm cùng độ ưu tiên; Tiểu trình ưu tiên cao hơn đến sau (dùng sleep()); …

Kết luận

Sinh viên nộp báo cáo về các bảng ghi và kết luận về các Yêu cầu đã thực hiện.