

5.3 进程间通讯 - 信号量同步_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 5.3 进程间通讯 – 信号量同步涵盖海量编程基础技术教程，以图文并茂的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

全部开发者教程

物联网 / 嵌入式工程师

第 24 周 stm32 芯片 – 智能硬件项目实战与企业笔试

未命名节

第 25 周 大厂必备 – linux 内核与文件系统移植

未命名节

第 26 周 嵌入式开发 – 系统移植 – bootloader、yocto

未命名节

第 27 周 嵌入式底层核心技能 – Linux 设备驱动初级

未命名节

第 28 周 嵌入式底层核心技能 – Linux 设备驱动中级

未命名节

第 29 周 嵌入式底层核心技能 – Linux 设备驱动高级 1

未命名节

第 30 周 嵌入式底层核心技能 – Linux 设备驱动高级 2

未命名节

第 31 周 嵌入式人工智能必备 – Python

未命名节

第 32 周 智能家居项目实战之客户端功能开发

未命名节

第 33 周 智能家居项目实战之网关端功能开发

未命名节

第 34 周 智能家居项目实战之设备端功能开发

未命名节

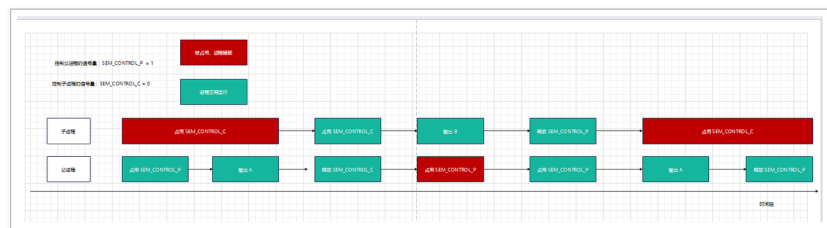
第 35 周 物联网 / 嵌入式项目答辩和就业指导

未命名节

未命名节

- 创建父子进程，输出“ABA”字符串，具体要求如下：
 - 父进程 输出 A
 - 子进程 输出 B
 - 父进程 输出 A ， 输出换行
 - 能够循环输出“ABA”字符
- 基本思路：
 - 通过创建一个信号量集合，包含 2 个信号量，一个信号量 编号为 0（SEM_CONTROL_P）控制父进程的运行与暂停，一个信号量 编号为 1（SEM_CONTROL_C）控制子进程的运行与暂停

- 控制逻辑:



- 信号初始化
 - SEM_CONTROL_P : 初始化为 1
 - SEM_CONTROL_C : 初始化为 0
- 子进程:
 - 占用 SEM_CONTROL_C , 此时子进程阻塞
 - 当父进程释放 SEM_CONTROL_C 时, 子进程 输出 B , 释放 SEM_CONTROL_P
 - 循环占用 SEM_CONTROL_C, 由于之前已经占用, 此时进入子进程阻塞, 等待父进程释放 SEM_CONTROL_C
- ```
while(1){
 sem_p(semid, SEM_CONTROL_C);
 printf("B");
 fflush(stdout);
 sem_v(semid, SEM_CONTROL_P);
}
```
- 父进程:
  - 占用 SEM\_CONTROL\_P, 此时父进程正常运行, 输出 A
  - 释放 SEM\_CONTROL\_C, 占用 SEM\_CONTROL\_P, 此时父进程阻塞, 子进程继续执行
  - 当子进程输出 B 之后, 释放 SEM\_CONTROL\_P, 父进程继续执行, 输出 A
  - 父进程 释放 SEM\_CONTROL\_P 循环结束
- ```
while(1){
    sem_p(semid, SEM_CONTROL_P);
    printf("A");
    fflush(stdout);
    sem_v(semid, SEM_CONTROL_C);
    sem_p(semid, SEM_CONTROL_P);
    printf("A");
    fflush(stdout);
    sem_v(semid, SEM_CONTROL_P);
    sleep(1);
    putchar('\n');
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
```

```
#include "sem.h"

#define SEM_CONTROL_P 0
#define SEM_CONTROL_C 1

int main(void)
{
    pid_t cpid;
    int semid;
    unsigned short values[2] = {1,0};

    semid = sem_create(2,values);
    if (semid == -1)
        exit(EXIT_FAILURE);

    cpid = fork();

    if (cpid == -1){
        perror("[ERROR] fork(): ");
        exit(EXIT_FAILURE);
    }else if (cpid == 0){
        while(1){
            sem_p(semid,SEM_CONTROL_C);
            printf("B");
            fflush(stdout);
            sem_v(semid,SEM_CONTROL_P);
        }
    }else{
        while(1){
            sem_p(semid,SEM_CONTROL_P);
            printf("A");
            fflush(stdout);
            sem_v(semid,SEM_CONTROL_C);
            sem_p(semid,SEM_CONTROL_P);
            printf("A");
            fflush(stdout);
            sem_v(semid,SEM_CONTROL_P);
            sleep(1);
            putchar('\n');
        }

        wait(NULL);
    }

    return 0;
}
```

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

