

[Q]: Как обстоит дело с задачами реального времени

[A]: Julius Goryavsky (2:5030/16.32)

1. Как использовать порты ввода-вывода?

1. Через IOPL-сегменты в прикладной программе. Но при этом остается

возможность переключения контекстов OS-ом. :( (Если фрагмент кода не выполняется под CLI).

2. Из драйвера - ИМНО наиболее корректный метод.

2. Как обстоит дело с прерываниями?

a.) нет ли ограничений со стороны OS на обработку аппаратно генерируемых прерываний?

0. Прерывания может обрабатывать только драйвер. При этом некоторые

(долгоработающие) вызовы DevHlp не доступны в контексте обработки прерывания.

1. Есть ряд ограничений/соглашений по использованию одного общего IRQ

разными драйверами.

b.) Как обрабатываются под OSом прерывания ?

Драйвер регистрирует обработчик нужного IRQ и сообщает OS-у какой размер

стека ему необходим. OS вызывает обработчик, когда происходит прерывание, подготовив для него стек. Обработчик взаимодействует с устройством (с помощью IN/OUT, или через память) и OS-ом (с помощью вызовов DevHlp), а по концу обработки прерывания вызывает DevHlp\_EOI.

3. Что с таймером - вроде как можно получить даже микросекунды ?

1. Можно узнать текущее время с точностью до микросекунд, но организовывать

задержки или прерывания с подобной точностью - невозможно.

2. Если есть Pentium - то можно узнать время с точностью ~20 тактов

процессора.

3. Таймер в смысле прерываний, на высоких частотах, недоступен. Квант

системного таймера = 31 ms.

4. Приоритеты задач: есть ли принципиальная возможность остановить все процессы, кроме одного (в критичные моменты времени) и как это может

согласовываться с виртуализацией памяти ?

Есть 4 класса приоритетов с 32 приоритетами в каждом классе. Задачи класса

Time Critical приостанавливают все остальные, кроме других задач класса Time Critical, с более высоким приоритетом. Приоритет, ясное дело, можно выставлять свой для каждой цепочки (thread).

Виртуальную память - точнее страничный обмен - можно отключать. Но если она

включена - сам виноват, попытаешься читать страницу которая лежит на диске - потеряешь время.

5. Есть ли принципиальная возможность для безусловного запуска процессов с интервалом от секунды до миллисекунды и меньше независимо от обращений к дискам и прочего. (Абсолютные приоритеты ?

Time Critical - почти что абсолютные приоритеты. Но активность дискового

драйвера они "подавить" не могут... Работа драйвера по обслуживанию прерываний, например от диска - приоритетней всех цепочек, с любым приоритетом.

[A]: Andrew Belov (2:5020/181.2)

Помимо time-critical, в OS/2 v 4.00+ есть недокументированная возможность приостановить все треды/процессы, кроме текущего треда:

```
#include <stdio.h> #define INCL_BASE #include <os2.h>
```

```
APIRET APIENTRY DosSysCtl(ULONG entry, PULONG data);
```

```
void main() { ULONG f; int i;
```

```
printf("Freezing...\n"); f=0; DosSysCtl(14, &f); for(i=1; i<=10; i++) {
```

```
DosSleep(1000);  
printf("%d\n", i);
```

```
} f=1; DosSysCtl(14, &f); printf("Defrost!\n"); }
```

...где DosSysCtl==DOSCALL1→876 (в хидерах его нет, но OS2386.LIB о нем уже знает).

From:

<https://osfree.org/doku/> - **osFree wiki**

Permanent link:

<https://osfree.org/doku/doku.php?id=ru:os2faq:os2prog:os2prog.030>

Last update: **2014/06/20 05:08**



