

*Библиотека поддержки
контрольных точек*

С.В. Попова, С.В. Шарф



Введение



- ★ Необходимость контрольных точек и некоторые проблемы их реализации



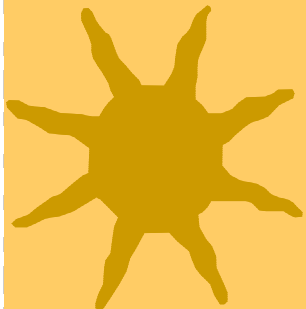
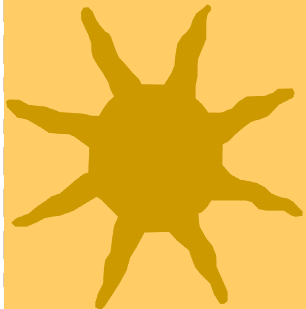
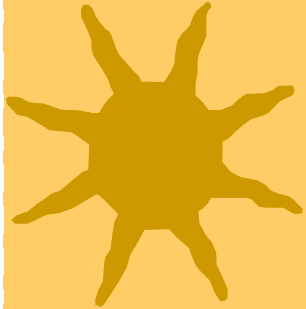
- ★ Класс рассматриваемых задач



- Итерационные задачи, которые могут использовать одни и те же файлы для чтения исходных данных и записи промежуточных результатов.



Примерная схема программы



• • •
<открытие $f1, f2, f3$ на чтение>
<чтение $f1, f2, f3$ >
<закрытие файлов $f1, f2, f3$ >
<начало цикла>
 <длительный счет>
 <открытие $f1, f2, f3$ на запись>
 <запись $f1, f2, f3$ >
 <закрытие $f1, f2, f3$ >
<конец цикла>
• • •



Цели разработки



★ Надежность



★ Простота использования



★ Эффективность



Подготовительные действия

- ★ Файл olddata/info :

```
#список файлов, содержащих данные
```

```
f1
```

```
f2
```

```
f3
```

- ★ Добавить в программу необходимые вызовы
- ★ Транслировать:

```
mpicc -o test test.c -L/common/mpich/lib -lmpidf
```

ИЛИ

```
cc -o test test.c -lrouter -lpthread
```

Дополнения 1. Во вспомогательном каталоге появится служебный файл (кроме перенесенных файлов).

2. Вспомогательный каталог не трогать до полного окончания работы программы.

3. Работает пока только в ИММ УрО РАН



Основные подпрограммы

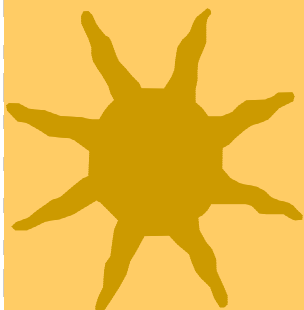
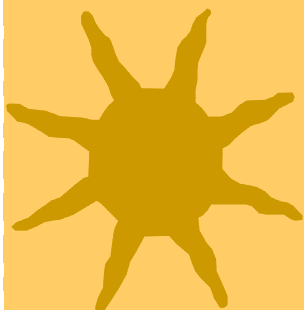
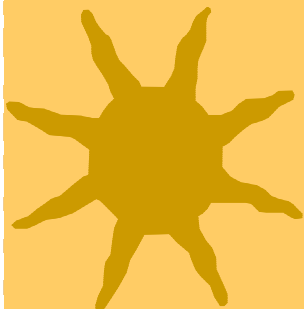
★ Программа содержит:

```
#include <staydf.h>                ( include 'staydf_f.h' )
. . .
iInitDF (nproc);                   ( call iinitdf( n_proc, cc ) )
iRestoreDF();                      ( call irestoredf(cc) )
<открытие f1, f2, f3 на чтение>
<чтение f1, f2, f3 >
<заккрытие файлов f1, f2, f3 >
<начало цикла>
  iHideDF();                       ( call ihidedf(cc) )
  <счет>
  <открытие f1, f2, f3 на запись>
  <запись f1, f2, f3>
  <заккрытие f1, f2, f3 >
< конец цикла>
iEOWDF ();                         ( call ieowdf(cc) )
```

Дополнение: На момент сохранения, необходимые файлы должны быть закрыты.



«Разбор полётов»



. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов *f1, f2, f3* на чтение>

<чтение *f1, f2, f3* >

<заккрытие файлов *f1, f2, f3* >

<начало цикла>

iHideDF();

<счет>

<открытие файлов *f1, f2, f3* на запись>

<запись *f1, f2, f3* >

<заккрытие файлов *f1, f2, f3* >

<конец цикла>

iEOWDF ();

Рабочий каталог

...

f1

f2

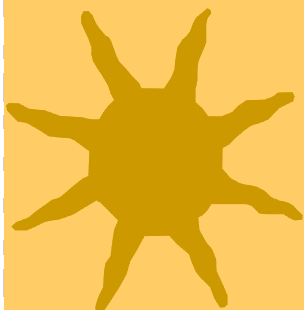
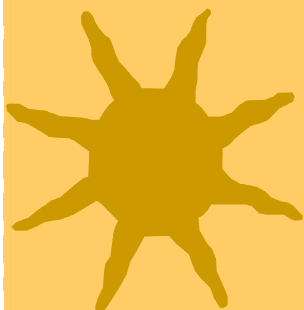
f3

olddata

info	<i>f1</i>
	<i>f2</i>
	<i>f3</i>



«Разбор полётов»



. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов *f1, f2, f3* на чтение>

<чтение *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<начало цикла>

iHideDF();

<счет>

<открытие файлов *f1, f2, f3* на запись>

<запись *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<конец цикла>

iEOWDF ();

Рабочий каталог

...

f1

f2

f3

olddata

info	<i>f1</i>
	<i>f2</i>
	<i>f3</i>



«Разбор полётов»



. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов f1, f2, f3 на чтение>

<чтение f1, f2, f3 >

<закрытие файлов f1, f2, f3 >

<начало цикла>

iHideDF();

<счет>

<открытие файлов f1, f2, f3 на запись>

<запись f1, f2, f3 >

<закрытие файлов f1, f2, f3 >

<конец цикла>

iEOWDF ();

Рабочий каталог

...

f1

f2

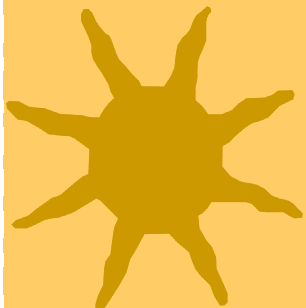
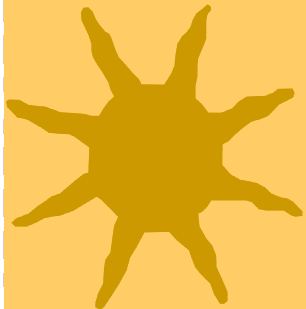
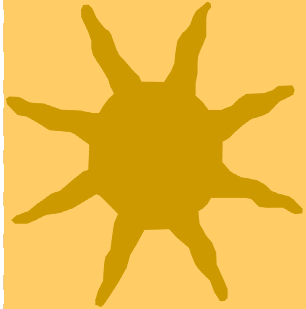
f3

olddata

info	f1
	f2
	f3



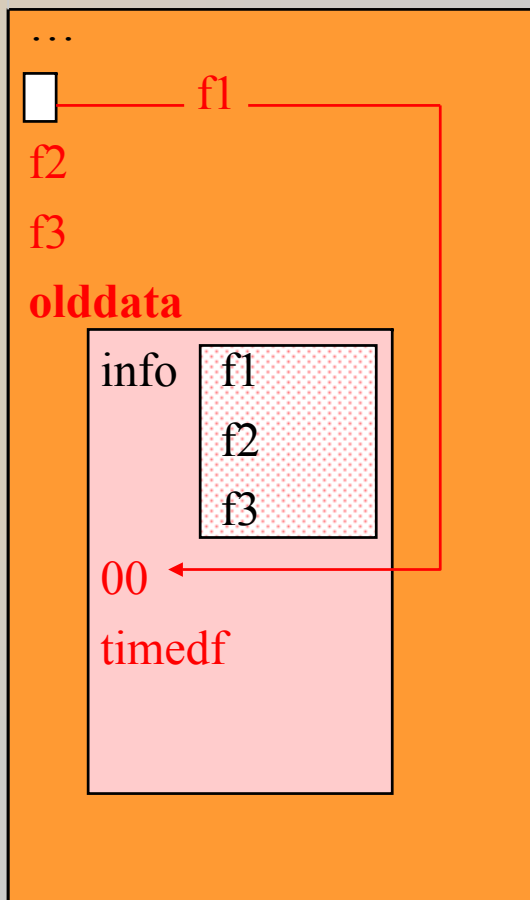
«Разбор полётов»



. . .

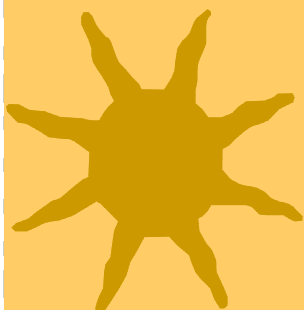
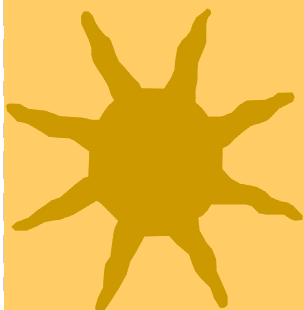
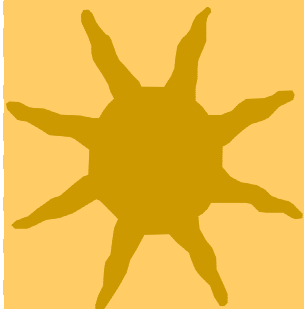
iInitDF (nproc);
iRestoreDF();
<открытие файлов *f1, f2, f3* на чтение>
<чтение *f1, f2, f3*>
<закрытие файлов *f1, f2, f3*>
 <начало цикла>
 iHideDF();
 <счет>
 <открытие файлов *f1, f2, f3* на запись>
 <запись *f1, f2, f3*>
 <закрытие файлов *f1, f2, f3*>
 <конец цикла>
iEOWDF ();

Рабочий каталог





«Разбор полётов»



. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов *f1, f2, f3* на чтение>

<чтение *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<начало цикла>

iHideDF();

<счет>

<открытие файлов *f1, f2, f3* на запись>

<запись *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<конец цикла>

iEOWDF ();

Рабочий каталог

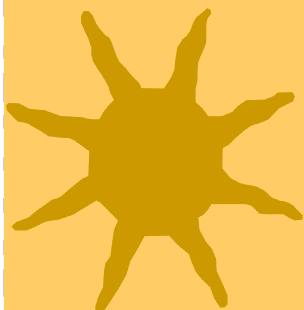
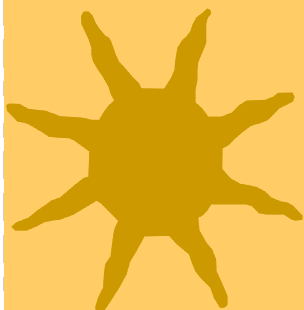
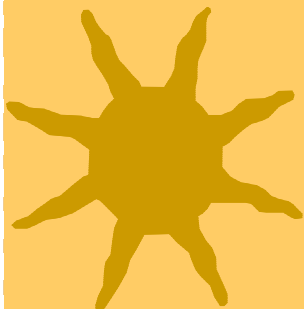
...

olddata

info	f1
	f2
	f3
00	
01	
02	
timedf	



«Разбор полётов»



. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов *f1, f2, f3* на чтение>

<чтение *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<начало цикла>

iHideDF();

<счет>

<открытие файлов *f1, f2, f3* на запись>

<запись *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<конец цикла>

iEOWDF ();

Рабочий каталог

...

f1

f2

f3

olddata

info	<i>f1</i>
	<i>f2</i>
	<i>f3</i>
00	
01	
02	
timedf	



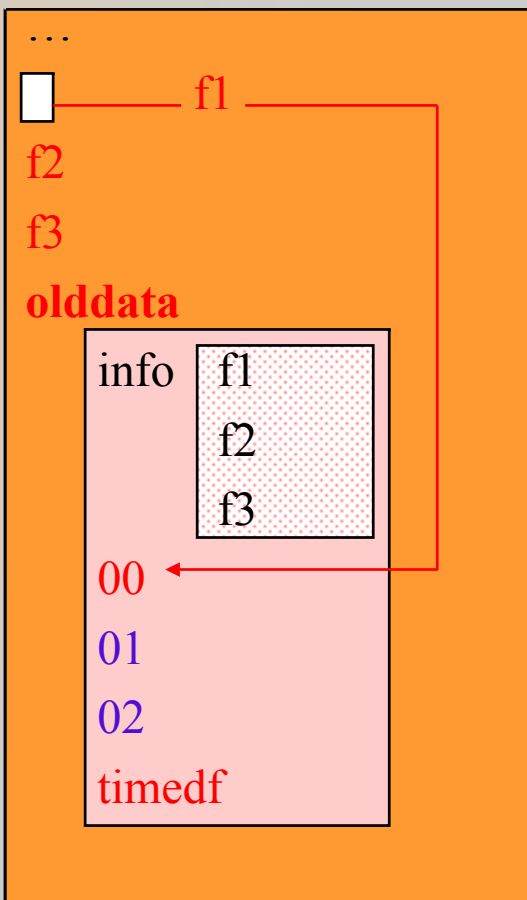
«Разбор полётов»



. . .

iInitDF (nproc);
iRestoreDF();
<открытие файлов *f1, f2, f3* на чтение>
<чтение *f1, f2, f3* >
<закрытие файлов *f1, f2, f3* >
 <начало цикла>
 iHideDF();
 <счет>
 <открытие файлов *f1, f2, f3* на запись>
 <запись *f1, f2, f3* >
 <закрытие файлов *f1, f2, f3* >
 <конец цикла>
iEOWDF ();

Рабочий каталог





«Разбор полётов»



Ситуация 1
(сбой)

. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов *f1, f2, f3* на чтение>

<чтение *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<начало цикла>

iHideDF();

<счет>

<открытие файлов *f1, f2, f3* на запись>

<запись *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<конец цикла>

iEOWDF ();

Рабочий каталог

...

f1

f2

olddata

info

f1

f2

f3

00

01

02

timedf



«Разбор полётов»



Ситуация 1

(инициализация)

. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов *f1, f2, f3* на чтение>

<чтение *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<начало цикла>

iHideDF();

<счет>

<открытие файлов *f1, f2, f3* на запись>

<запись *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<конец цикла>

iEOWDF ();

Рабочий каталог

...

f1

f2

olddata

info

f1

f2

f3

00

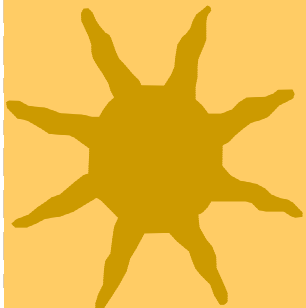
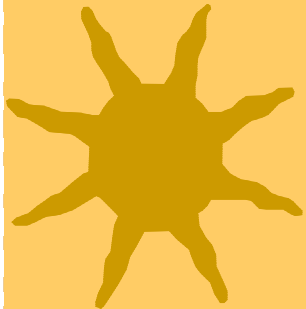
01

02

timedf



«Разбор полётов»



Ситуация 1

(восстановление)

. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов *f1, f2, f3* на чтение>

<чтение *f1, f2, f3*>

<закрытие файлов *f1, f2, f3*>

<начало цикла>

iHideDF();

<счет>

<открытие файлов *f1, f2, f3* на запись>

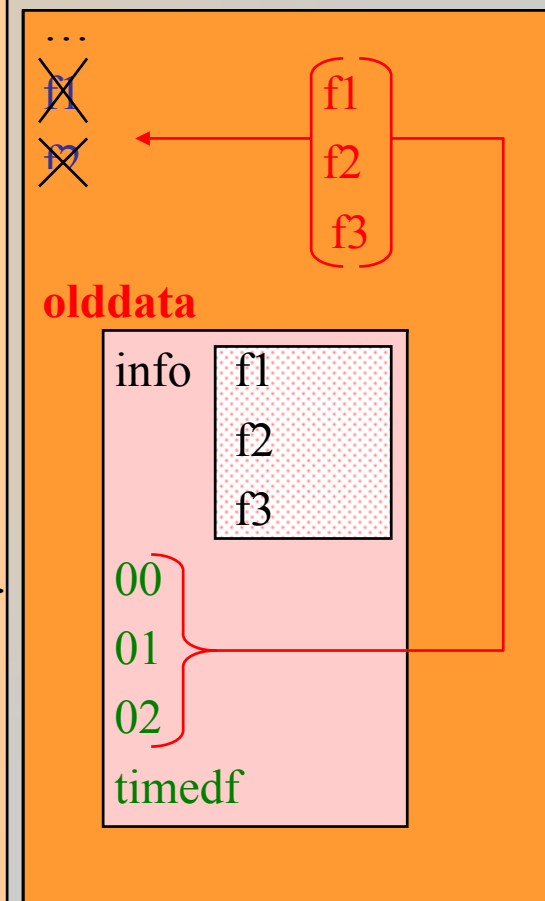
<запись *f1, f2, f3*>

<закрытие файлов *f1, f2, f3*>

<конец цикла>

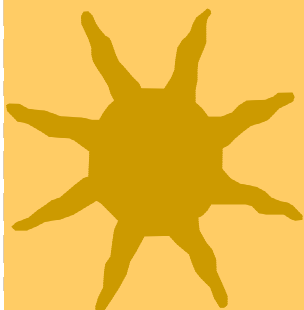
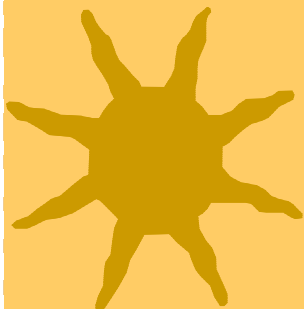
iEOWDF ();

Рабочий каталог





«Разбор полётов»



Ситуация 1
(открытие)

. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов f1, f2, f3 на чтение>

<чтение f1, f2, f3 >

<заккрытие файлов f1, f2, f3 >

<начало цикла>

iHideDF();

<счет>

<открытие файлов f1, f2, f3 на запись>

<запись f1, f2, f3 >

<заккрытие файлов f1, f2, f3 >

<конец цикла>

iEOWDF ();

Рабочий каталог

...

f1

f2

f3

olddata

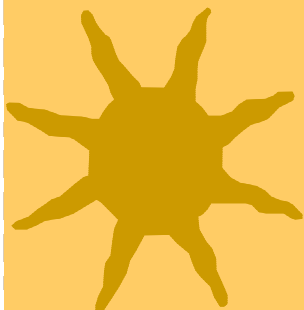
info	f1
	f2
	f3
timedf	



«Разбор полётов»



. . .
iInitDF (nproc);
iRestoreDF();
<открытие файлов *f1, f2, f3* на чтение>
<чтение *f1, f2, f3*>
<закрытие файлов *f1, f2, f3*>
<начало цикла>



iHideDF();

<счет>
<открытие файлов *f1, f2, f3* на запись>
<запись *f1, f2, f3*>
<закрытие файлов *f1, f2, f3*>
<конец цикла>

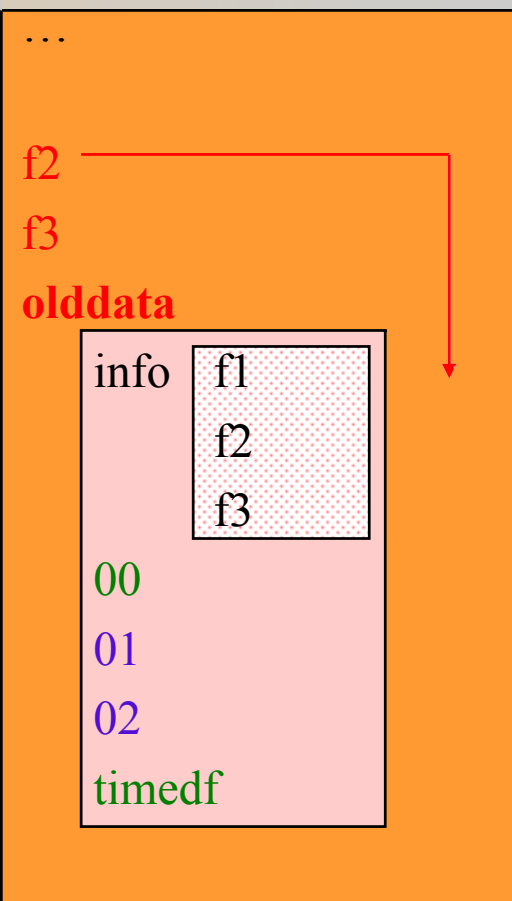


Ситуация 2

(сбой)

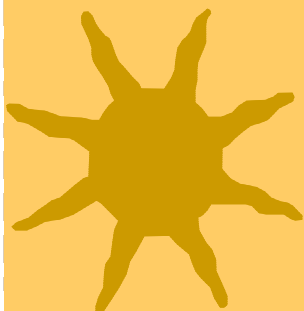
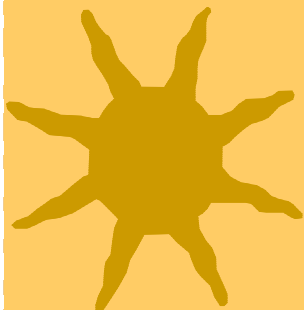
iEOWDF ();

Рабочий каталог





«Разбор полётов»



Ситуация 2

(инициализация)

. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов *f1, f2, f3* на чтение>

<чтение *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<начало цикла>

iHideDF();

<счет>

<открытие файлов *f1, f2, f3* на запись>

<запись *f1, f2, f3* >

<закрытие файлов *f1, f2, f3* >

<конец цикла>

iEOWDF ();

Рабочий каталог

...

f2

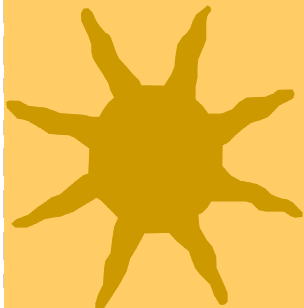
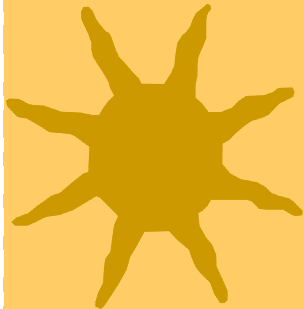
f3

olddata

<i>info</i>	<i>f1</i>
	<i>f2</i>
	<i>f3</i>
<i>00</i>	
<i>01</i>	
<i>02</i>	
<i>timedf</i>	



«Разбор полётов»



Ситуация 2

(восстановление)

. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов *f1, f2, f3* на чтение>

<чтение *f1, f2, f3* >

<заккрытие файлов *f1, f2, f3* >

<начало цикла>

iHideDF();

<счет>

<открытие файлов *f1, f2, f3* на запись>

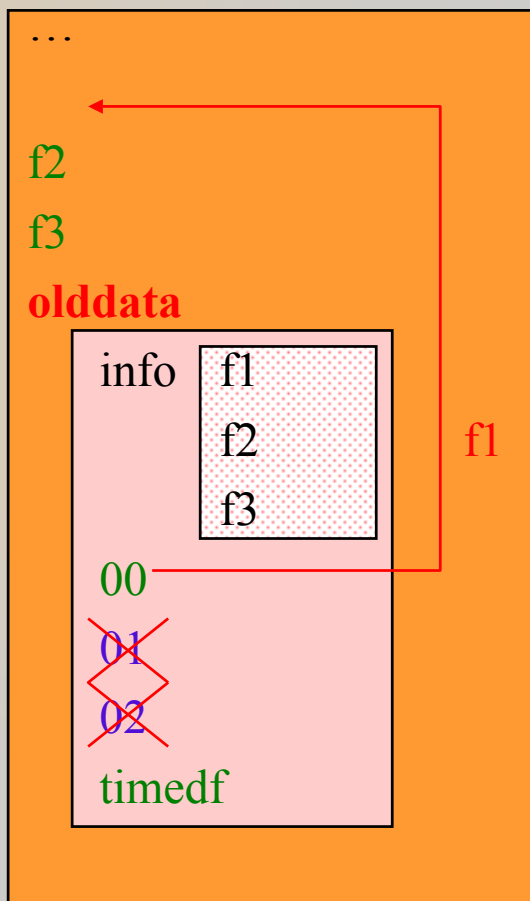
<запись *f1, f2, f3* >

<заккрытие файлов *f1, f2, f3* >

<конец цикла>

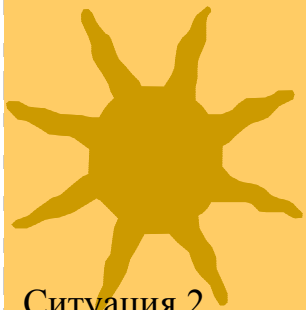
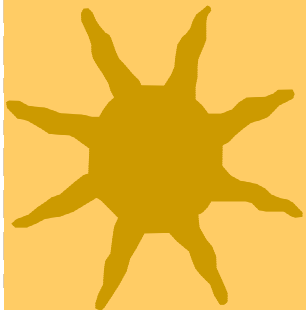
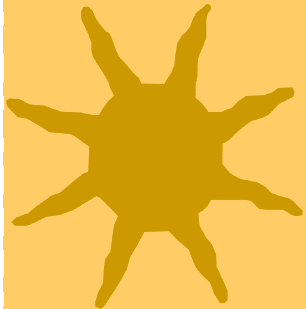
iEOWDF ();

Рабочий каталог





«Разбор полётов»



Ситуация 2
(открытие)

. . .

iInitDF (nproc);

iRestoreDF();

<открытие файлов f1, f2, f3 на чтение>

<чтение f1, f2, f3 >

<закрытие файлов f1, f2, f3 >

<начало цикла>

iHideDF();

<счет>

<открытие файлов f1, f2, f3 на запись>

<запись f1, f2, f3 >

<закрытие файлов f1, f2, f3 >

<конец цикла>

iEOWDF ();

Рабочий каталог

...

f1

f2

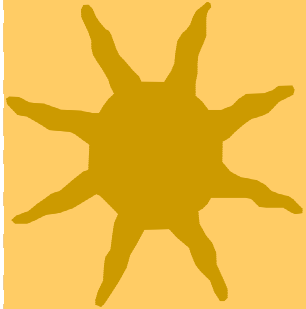
f3

olddata

info	f1
	f2
	f3
timedf	



Установки по умолчанию и вспомогательные подпрограммы

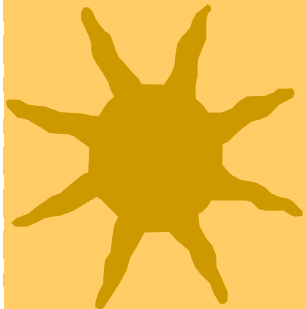


Установки по умолчанию

Подпрограммы управления

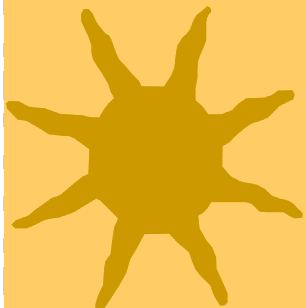
Прерывать работу в случае ошибки

set_errors(0 | 1)
call set_errors(0 | 1)



Информация об удачных действиях

set_no_diag(0 | 1)
call set_no_diag(0 | 1)



Отсутствие файла – не ошибка

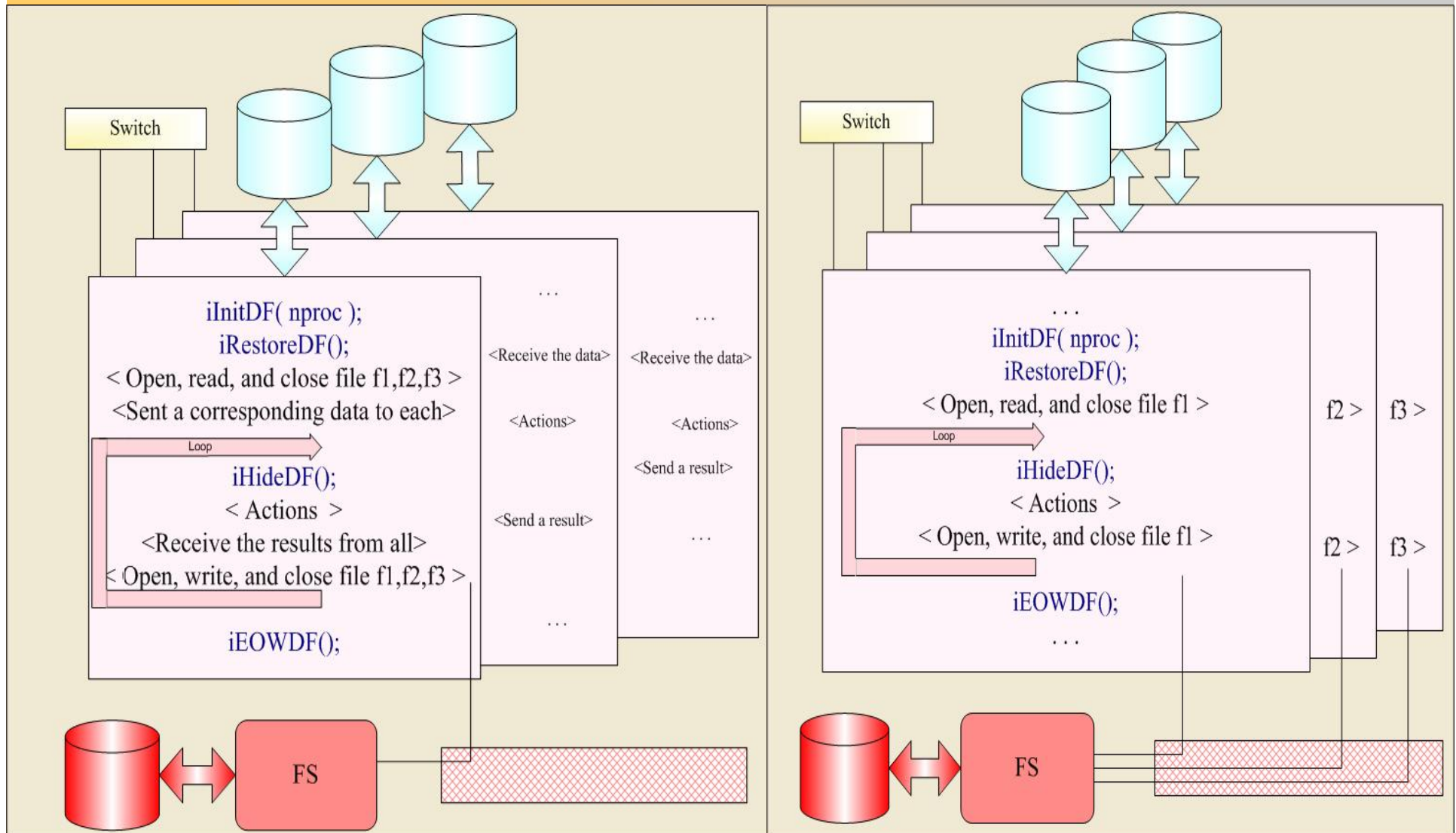
set_error_no_file(0 | 1)
call set_error_no_file(0 | 1)

Нет синхронизации процессов при сохранении

set_barrier(0 | 1)
call set_barrier (0 | 1)



Схемы распараллеливания задач





Опыт модификации программ



★ Анализировать все записи в файлы для выяснения связей между ними.



★ Обязательное закрытие файлов, указываемых в информационном файле, после чтения(записи).



★ Особое внимание — на циклы, в которых есть запись/чтение в файлы.



Основные достоинства



★ Сохранение лишь необходимых данных



★ Легкость использования



★ Нарращивание функциональности без изменения интерфейса

Некоторые недостатки: Невозможность использования открытых файлов. Необходимость внести изменения в текст программы



Предполагаемое развитие



★ Сохранение на каждом процессоре



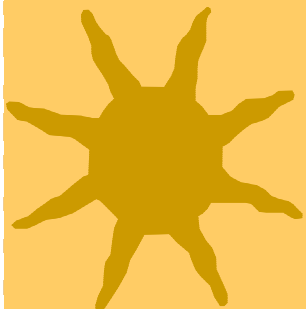
★ Работа с открытыми файлами



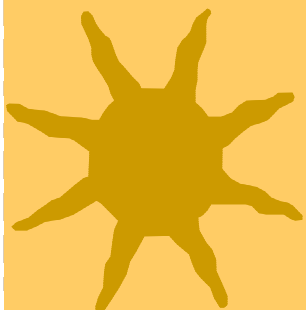
★ Сжатие информации



Информация и контакты



- ★ Руководство пользователям можно посмотреть на parallel.uran.ru



- ★ Со всеми вопросами и пожеланиями по поводу данной библиотеки можно обращаться:

Шарфу Сергею Владимировичу e-mail: scharf@imm.uran.ru

Поповой Светлане Владимировне e-mail: posvet@imm.uran.ru

По телефонам: 35-11,
49-32-48,
75-35-11.

