

## Запуск задач (практика)

*Олешко С.Б.  
Петербургский институт ядерной физики  
г.Гатчина*

- PERL файл (tstp.pl):

```
#!/usr/bin/perl -w
```

```
my $inpf=$ARGV[0];
```

```
$inpf=~/(.*)\.;/
```

```
my $outf=$1.'.out';
```

```
open(INP,"$inpf"); open(OUT,">$outf");
```

```
while (my $s=<INP>){
```

```
  $s*=$s; print(OUT "$s\n");
```

```
};
```

```
close(INP); close(OUT);
```

- **JDL файл (tstp.jdl):**

**Executable = "tstp.pl";**

**Arguments = "tstp.inp";**

**StdOutput = "std.out";**

**StdError = "std.err";**

**InputSandbox={"tstp.pl","tstp.inp"};**

**OutputSandbox = {"std.out", "std.err","tstp.out"};**

**RetryCount = 3;**

- INP файл (tstp.inp):

1

2

3

4

5

6

После завершения работы команды **glite-wms-job-output** в директорию, указанную в команде, скопируются 3 файла.

- std.err
- std.out
- tstp.out

Содержимое выходного файла с результатами счёта:

```
$ cat tstp.out
```

```
1
```

```
4
```

```
9
```

```
16
```

```
25
```

```
36
```

- Есть файл с исходным текстом программы на языке C `ctst.c`, которая выводит строку “Hello world” на стандартный вывод.
- Есть `Makefile` для сборки этой программы.
- Необходимо обеспечить сборку и запуск программы на удаленном ресурсе.

- JDL файл (subm2.jdl):

**JobType="Normal";**

**Executable = "startC.sh";**

**StdOutput = "ctst.out";**

**StdError = "ctst.err";**

**OutputSandbox = {"ctst.out","ctst.err"};**

**InputSandbox = {"startC.sh","ctst.c","Makefile"};**

**RetryCount=3;**

- Стартовый скрипт startC.sh:

```
#!/bin/bash
```

```
make           //сборка программы
```

```
chmod +x ctst //разрешаем запускать ее
```

```
./ctst         //запускаем
```

```
exit 0
```



- Текст программы на C - ctst.c:

```
#include <stdio.h>
int main(int argc, char **argv)
{
    char *name = argv[1];
    printf("\n\n\n");
    printf("Hello world!\n");
    printf("\n\n\n");
    // exit(0);
}
```

- **Makefile**

```
ctst: ctst.o
```

```
    g++ ctst.o -o ctst -lm
```

```
clean:
```

```
    rm ctst.o ctst
```

- Есть файл со скриптом, который генерит набор JDL файлов, количество которых определяется входным параметром
- Затем для каждого задания случайным образом выбираются слова из системного словаря `/usr/share/dict/words`, которые передаются, как аргументы для каждого из запускаемых заданий.
- Все задания запускаются и контролируется процесс их выполнения.
- После завершения всех заданий (успешного или нет) – выводится результат.

- Файл `echoword.sh`

```
#!/bin/bash
```

```
echo "Word $1 is $2";
```

- Запуск задания (вторым входным аргументом является имя ВО)

```
./submit-dictionary-jobs.sh 3 nw_ru
```