

```

1  -- xlo.hs -- low-level CGI program
2
3  import System.Environment ( getEnvironment
4                             , getEnv
5                             )
6  import System.IO
7  import qualified Data.ByteString as BS
8
9  main =
10     putStrLn "Content-type: text/plain\n"
11     >> hSetBuffering stdin NoBuffering -- для загрузки длинных HE-txt файлов
12     >> lookupEnv' "REQUEST_METHOD"
13     >>= query
14     >>= putStrLn
15
16 query (Just "GET") = getEnv "QUERY_STRING"
17 {-
18 query (Just "POST") = hGetContents stdin
19
20     Внимание! hGetContents stdin сохранено только для демонстрационных целей.
21     Действительно, работает с короткими текстовыми файлами без использования
22     CONTENT_LENGTH (за счет "ленивости"), но не может работать с длинными
23     и бинарными файлами
24 -}
25 query (Just "POST") = do
26     contentLength <- getEnv("CONTENT_LENGTH")
27     buffer <- BS.hGet stdin (read contentLength)
28     return $ show buffer
29     -- не идеально, но результат уже пригоден к анализу ;)
30 query Nothing = return "No REQUEST_METHOD"
31
32 -- lookupEnv not implemented in my GHC 7.0.3, (added to GHC 7.6.1)
33 lookupEnv' :: String -> IO (Maybe String)
34 lookupEnv' name = fmap (lookup name) getEnvironment
35
36 {- From forms.html, GET method:
37
38     t=the+text&r=on&b2=on&h=hidden+text&s=0ption+2
39
40 -- From forms.html, POST method (hGetContents stdin):
41
42 -----WebKitFormBoundary29jn2n7fliJaJ00G
43 Content-Disposition: form-data; name="upload"; filename="lo.hs"
44 Content-Type: text/x-haskell
45
46 -- lo.hs -- low-level CGI program
47
48 import System.Environment (getEnvironment)
49
50 main =
51     putStrLn "Content-type: text/plain\n"
52     >> getEnvironment
53     >>= mapM_ printPair
54
55 printPair (name, value) = putStrLn $ name ++ " = " ++ value
56
57 {-
58     ... здесь идёт текст "как есть" ...
59 -}
60
61 -----WebKitFormBoundary29jn2n7fliJaJ00G
62 Content-Disposition: form-data; name="comment"
63
64 A Haskell source
65 -----WebKitFormBoundary29jn2n7fliJaJ00G--
66
67
68 -}

```