Computação em Larga Escala

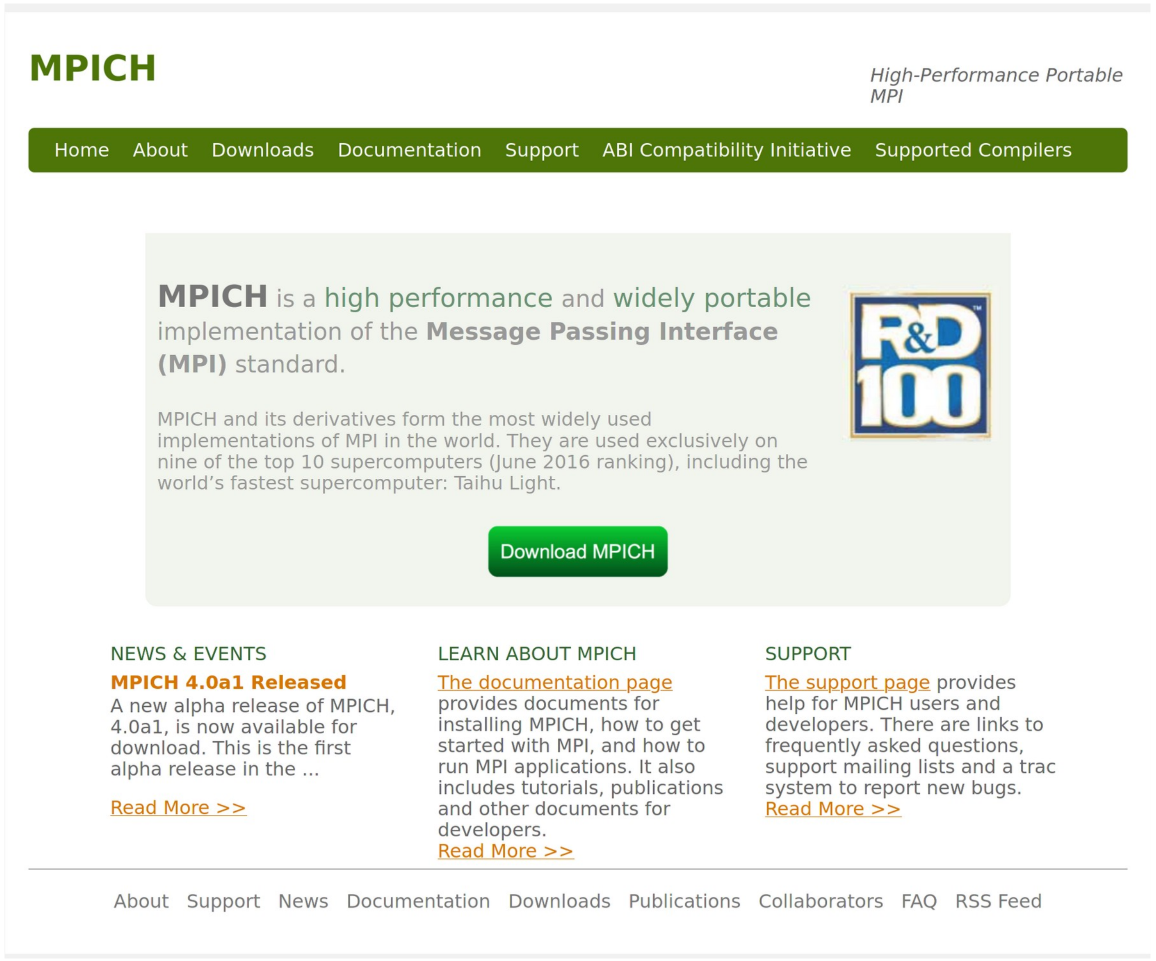
Instructions to install mpich

António Rui Borges

Summary

- *MPICH Site*
- *Source code installation*
  - *README file*
- *Binary installation for different operating systems*
- *Downloading examples archive*
  - *Program hello*
  - *Program sendRecData*
- *MPI library*

MPICH Site



MPICH is a high performance and widely portable implementation of the Message Passing Interface (MPI) standard.

MPICH and its derivatives form the most widely used implementations of MPI in the world. They are used exclusively on nine of the top 10 supercomputers (June 2016 ranking), including the world's fastest supercomputer: Taihu Light.

NEWS & EVENTS

MPICH 4.0a1 Released

A new alpha release of MPICH, 4.0a1, is now available for download. This is the first alpha release in the ...

Read More >>

LEARN ABOUT MPICH

The documentation page provides documents for installing MPICH, how to get started with MPI, and how to run MPI applications. It also includes tutorials, publications and other documents for developers.

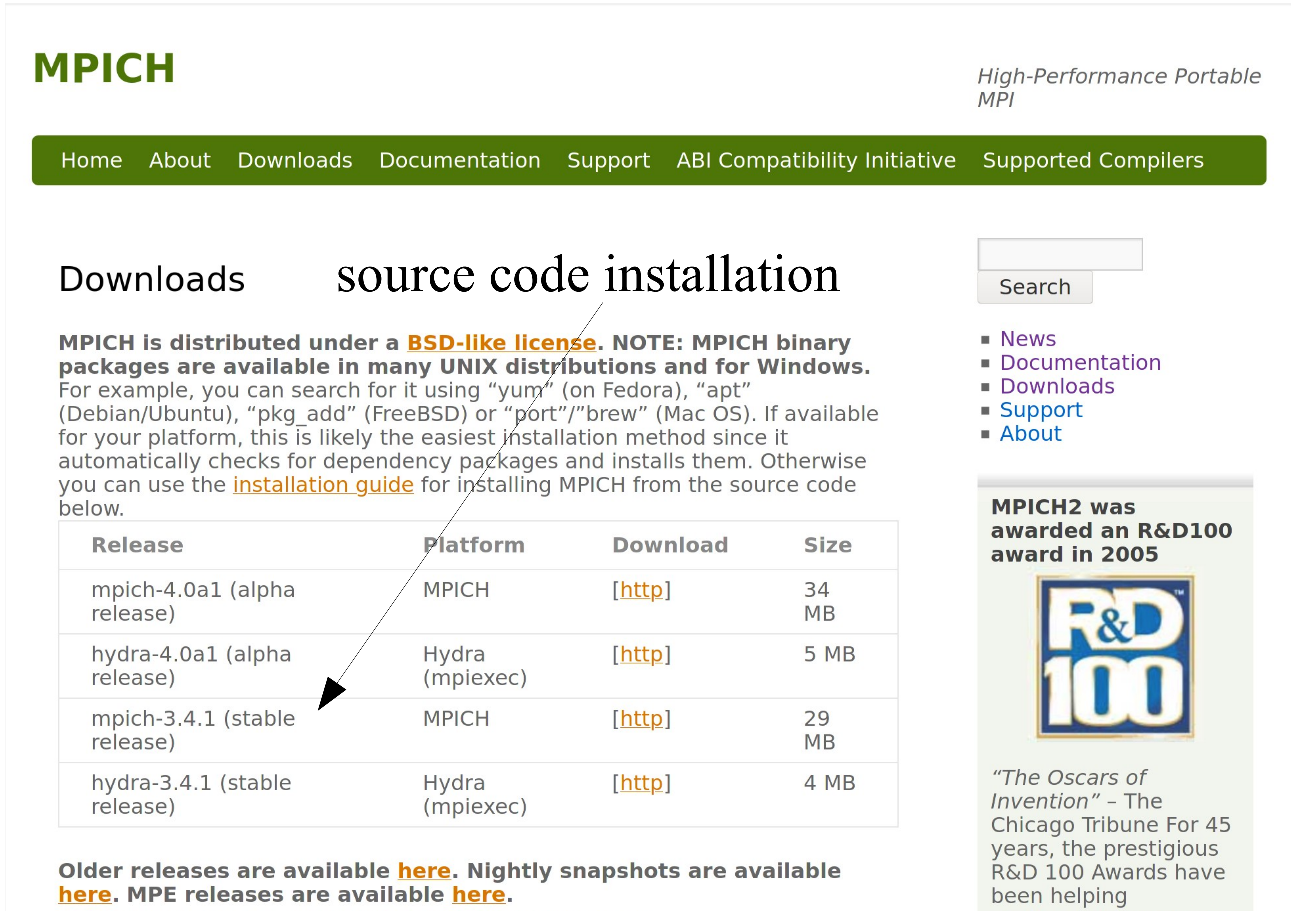
Read More >>

SUPPORT

The support page provides help for MPICH users and developers. There are links to frequently asked questions, support mailing lists and a trac system to report new bugs.

Read More >>

Source code installation in Linux - 1



source code installation

MPICH

High-Performance Portable MPI

Home About Downloads Documentation Support ABI Compatibility Initiative Supported Compilers

Downloads

MPICH is distributed under a BSD-like license. NOTE: MPICH binary packages are available in many UNIX distributions and for Windows. For example, you can search for it using "yum" (on Fedora), "apt" (Debian/Ubuntu), "pkg_add" (FreeBSD) or "port"/"brew" (Mac OS). If available for your platform, this is likely the easiest installation method since it automatically checks for dependency packages and installs them. Otherwise you can use the installation guide for installing MPICH from the source code below.

Release	Platform	Download	Size
mpich-4.0a1 (alpha release)	MPICH	[http]	34 MB
hydra-4.0a1 (alpha release)	Hydra (mpiexec)	[http]	5 MB
mpich-3.4.1 (stable release)	MPICH	[http]	29 MB
hydra-3.4.1 (stable release)	Hydra (mpiexec)	[http]	4 MB

Older releases are available here. Nightly snapshots are available here. MPE releases are available here.

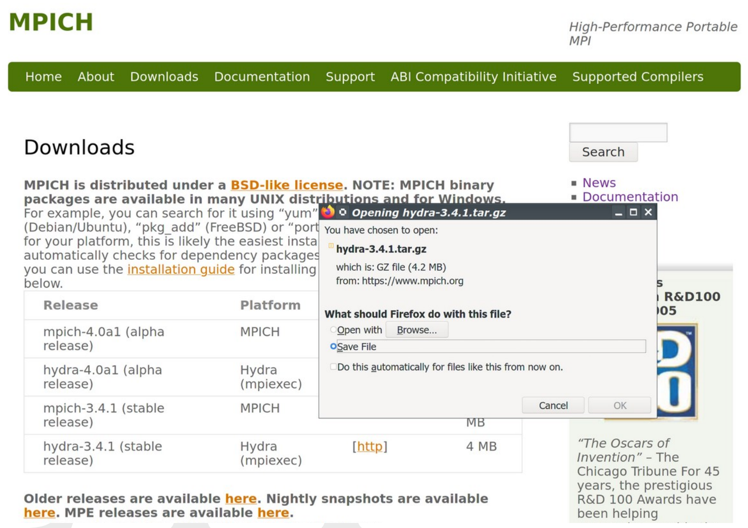
Search

- News
- Documentation
- Downloads
- Support
- About

MPICH2 was awarded an R&D100 award in 2005

"The Oscars of Invention" – The Chicago Tribune For 45 years, the prestigious R&D 100 Awards have been helping

Source code installation in Linux - 2


MPICH

High-Performance Portable MPI

Downloads

MPICH is distributed under a BSD-like license. NOTE: MPICH binary packages are available in many UNIX distributions and for Windows. For example, you can search for it using "yum" (Debian/Ubuntu), "pkg_add" (FreeBSD) or "port for your platform, this is likely the easiest insta automatically checks for dependency packages you can use the installation guide for installing below.

Release	Platform		
mpich-4.0a1 (alpha release)	MPICH		
hydra-4.0a1 (alpha release)	Hydra (mpiexec)		
mpich-3.4.1 (stable release)	MPICH		MB
hydra-3.4.1 (stable release)	Hydra (mpiexec)	[http]	4 MB

Older releases are available here. Nightly snapshots are available here. MPE releases are available here.

R&D100 05

"The Oscars of Invention" – The Chicago Tribune For 45 years, the prestigious R&D 100 Awards have been helping

Opening hydra-3.4.1.tar.gz

You have chosen to open:

hydra-3.4.1.tar.gz

which is: GZ file (4.2 MB)
from: https://www.mpich.org

What should Firefox do with this file?

Open with [Browse...]
[✓ Save File]
Do this automatically for files like this from now on.

[Cancel] [OK]

- copy `mpich-3.4.1.tar.gz` to your base directory `/home/<username>`
- unpack it with the command `tar -zxvf mpich-3.4.1.tar.gz`
- enter the directory `/home/<username>/mpich-3.4.1.tar.gz`
- open the text file `README` with a text editor
- follow the instructions

README file

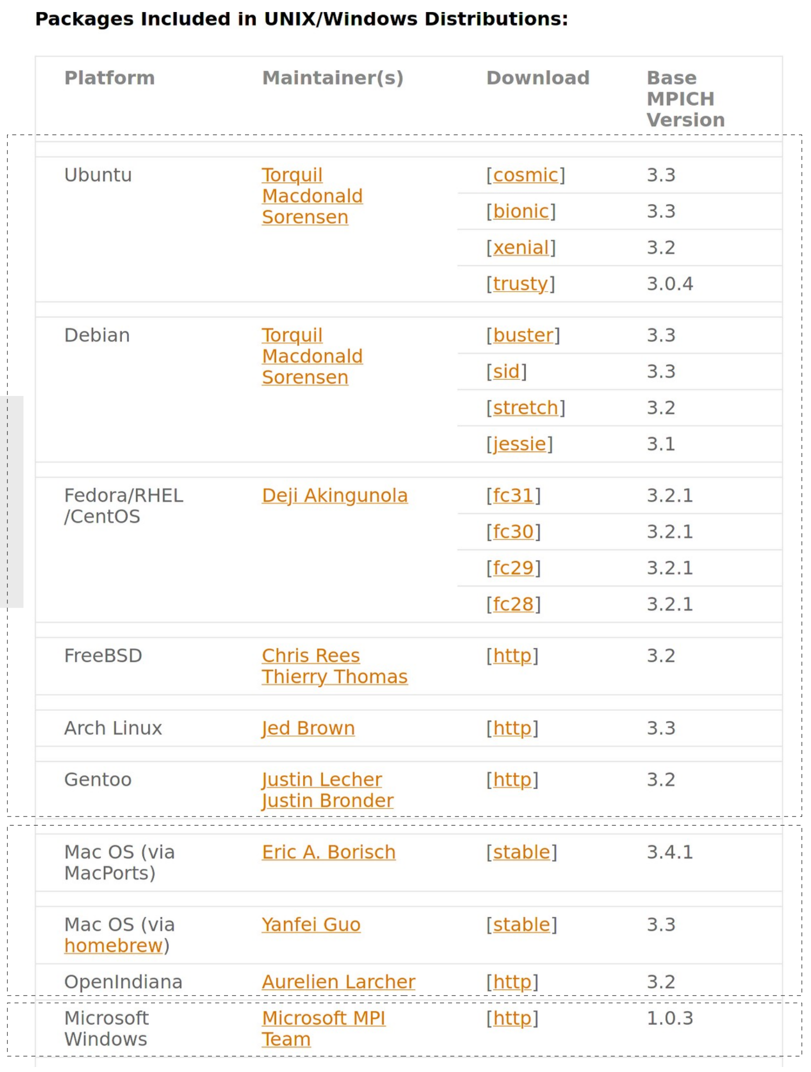
```
1. Getting Started

==================
The following instructions take you through a sequence of steps to get
the default configuration (ch3 device, nemesis channel (with TCP and
shared memory), Hydra process management) of MPICH up and running.
(a) You will need the following prerequisites.
   - REQUIRED: This tar file mpich-3.3.2.tar.gz
   - REQUIRED: A C compiler (C99 support is required. See
     https://wiki.mpich.org/mpich/index.php/Shifting_toward_C99)
   - OPTIONAL: A C++ compiler, if C++ applications are to be used
     (g++, etc.). If you do not require support for C++ applications,
     you can disable this support using the configure option
     --disable-cxx (configuring MPICH is described in step 1(d)
     below).
   - OPTIONAL: A Fortran compiler, if Fortran applications are to be
     used (gfortran, ifort, etc.). If you do not require support for
     Fortran applications, you can disable this support using
     --disable-fortran (configuring MPICH is described in step 1(d)
     below).

                              . . .
```

Binary installation for different operating systems

different Linx distributions

Mac OS

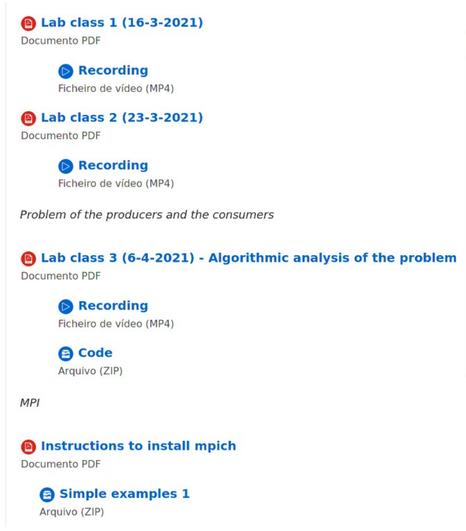
Windows



Packages Included in UNIX/Windows Distributions:

Platform	Maintainer(s)	Download	Base MPICH Version
Ubuntu	Torquil Macdonald Sorensen	[cosmic]	3.3
		[bionic]	3.3
		[xenial]	3.2
		[trusty]	3.0.4
Debian	Torquil Macdonald Sorensen	[buster]	3.3
		[sid]	3.3
		[stretch]	3.2
		[jessie]	3.1
Fedora/RHEL /CentOS	Deji Akingunola	[fc31]	3.2.1
		[fc30]	3.2.1
		[fc29]	3.2.1
		[fc28]	3.2.1
FreeBSD	Chris Rees Thierry Thomas	[http]	3.2
Arch Linux	Jed Brown	[http]	3.3
Gentoo	Justin Lecher Justin Bronder	[http]	3.2
Mac OS (via MacPorts)	Eric A. Borisch	[stable]	3.4.1
Mac OS (via homebrew)	Yanfei Guo	[stable]	3.3
OpenIndiana	Aurelien Larcher	[http]	3.2
Microsoft Windows	Microsoft MPI Team	[http]	1.0.3

Downloading examples archive



Lab class 1 (16-3-2021)
Documento PDF

Recording
Ficheiro de vídeo (MP4)

Lab class 2 (23-3-2021)
Documento PDF

Recording
Ficheiro de vídeo (MP4)

Problem of the producers and the consumers

Lab class 3 (6-4-2021) - Algorithmic analysis of the problem
Documento PDF

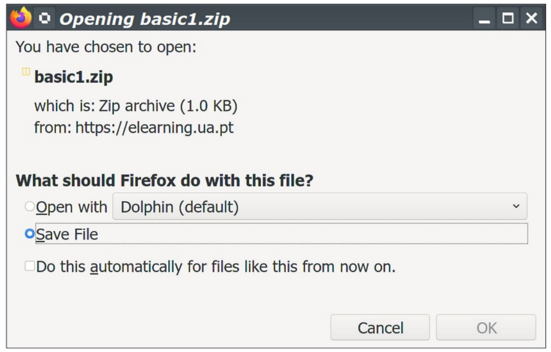
Recording
Ficheiro de vídeo (MP4)

Code
Arquivo (ZIP)

MPI

Instructions to install mpich
Documento PDF

Simple examples 1
Arquivo (ZIP)



Opening basic1.zip

You have chosen to open:

basic1.zip

which is: Zip archive (1.0 KB)
from: https://elearning.ua.pt

What should Firefox do with this file?

Open with Dolphin (default)
[✓ Save File]

Do this automatically for files like this from now on.

- create the directory `/home/<username>/mpi/examples`
- copy `basic1.zip` to this directory
- enter the directory `/home/<username>/mpi/examples`
- unpack it with the command `unzip basic1.zip`

Program hello - 1

```
#include <mpi.h>
#include <stdio.h>

int main (int argc, char ** argv)
{
   int rank, size;

   MPI_Init (&argc, &argv);
   MPI_Comm_rank (MPI_COMM_WORLD, &rank);
   MPI_Comm_size (MPI_COMM_WORLD, &size);
   printf ("Hello! I am %d of %d.\n", rank + 1, size);
   MPI_Finalize ();
   return 0;
}
```

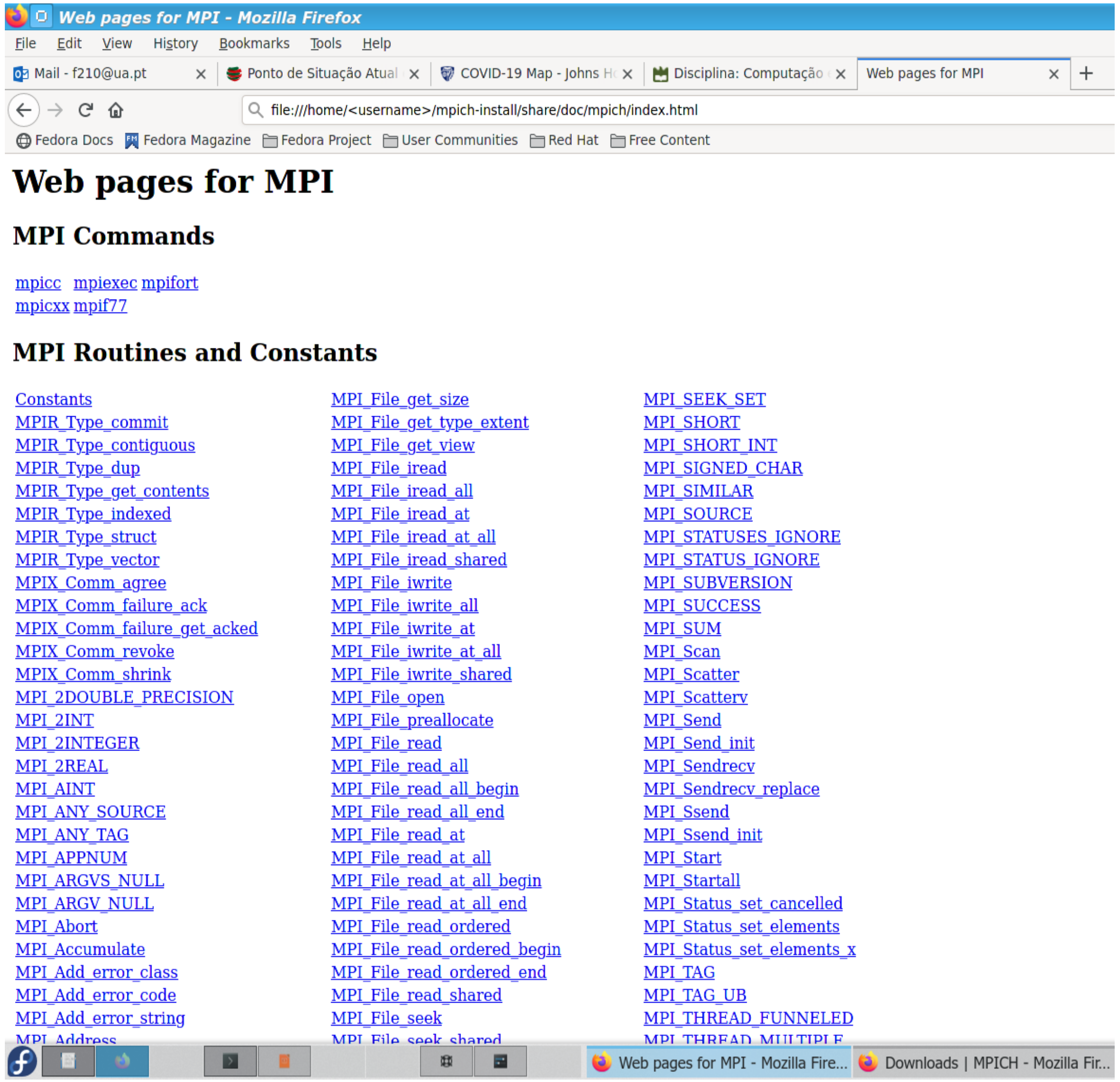
Program hello - 2

```
[ruib@ruib-laptop basic1]$ ll
total 8
-rw-rw-r--. 1 ruib ruib 295 Mar 11  2019 hello.c
-rw-rw-r--. 1 ruib ruib 665 Mar 11  2019 sendRecData.c
[ruib@ruib-laptop basic1]$ mpicc -Wall -o hello hello.c
[ruib@ruib-laptop basic1]$ ll
total 28
-rwxrwxr-x. 1 ruib ruib 19672 Apr 20 09:43 hello
-rw-rw-r--. 1 ruib ruib   295 Mar 11  2019 hello.c
-rw-rw-r--. 1 ruib ruib   665 Mar 11  2019 sendRecData.c
[ruib@ruib-laptop basic1]$ mpiexec -n 4 ./hello
Hello! I am 2 of 4.
Hello! I am 3 of 4.
Hello! I am 1 of 4.
Hello! I am 4 of 4.
[ruib@ruib-laptop basic1]$ mpiexec -n 8 ./hello
Hello! I am 2 of 8.
Hello! I am 3 of 8.
Hello! I am 6 of 8.
Hello! I am 8 of 8.
Hello! I am 4 of 8.
Hello! I am 7 of 8.
Hello! I am 1 of 8.
Hello! I am 5 of 8.
[ruib@ruib-laptop basic1]$
```

spawning of 4 processes

spawning of 8 processes

MPI library - 1


Web pages for MPI - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Mail - f210@ua.pt
Ponto de Situação Atual
COVID-19 Map - Johns H
Disciplina: Computação
Web pages for MPI
file:///home/<username>/mpich-install/share/doc/mpich/index.html
Fedora Docs
Fedora Magazine
Fedora Project
User Communities
Red Hat
Free Content
Web pages for MPI - Mozilla Fire...
Downloads | MPICH - Mozilla Fir...

Web pages for MPI

MPI Commands

mpicc mpiexec mpifort
mpicxx mpif77

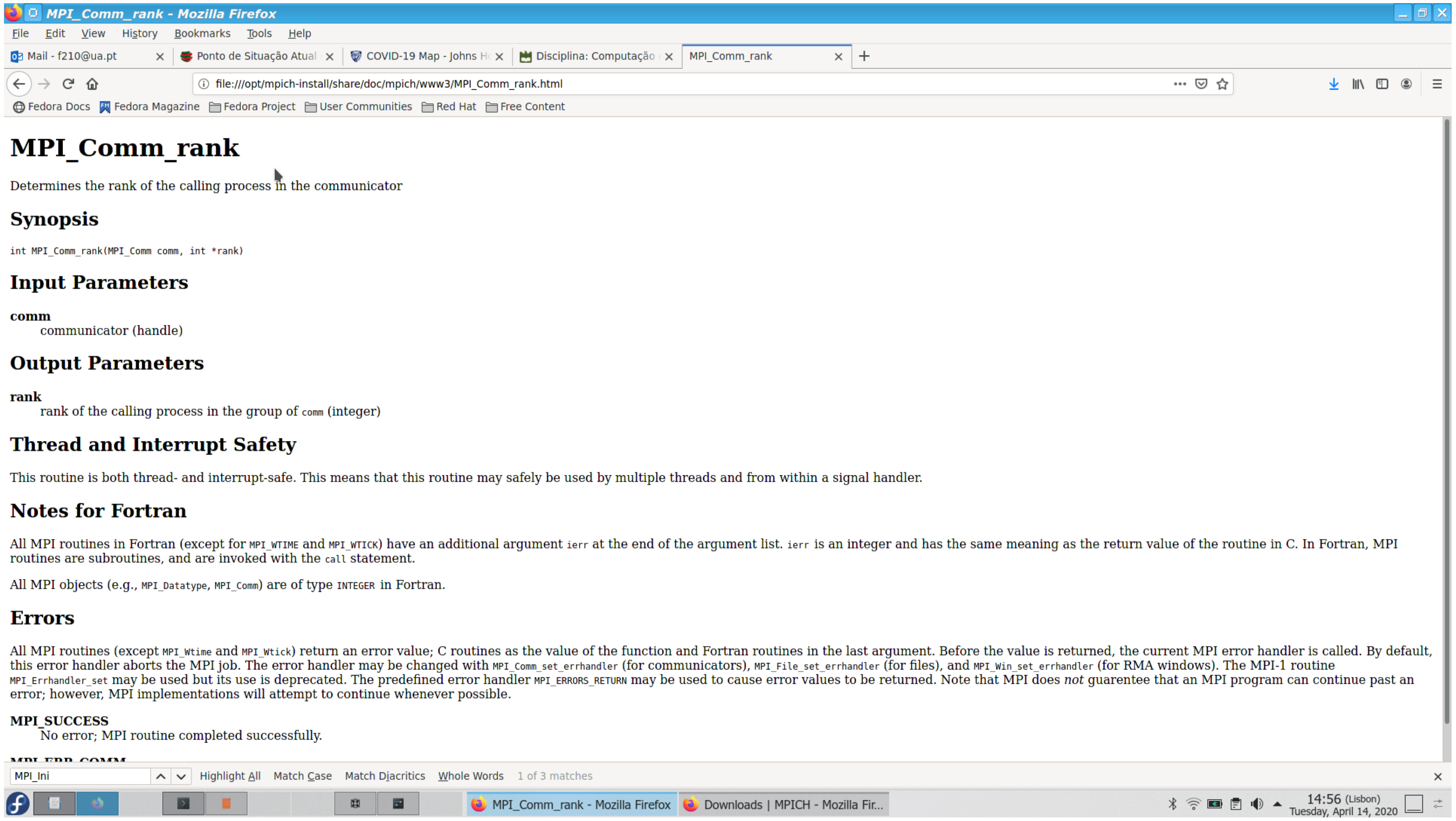
MPI Routines and Constants

Constants
MPIR_Type_commit
MPIR_Type_contiguous
MPIR_Type_dup
MPIR_Type_get_contents
MPIR_Type_indexed
MPIR_Type_struct
MPIR_Type_vector
MPIX_Comm_agree
MPIX_Comm_failure_ack
MPIX_Comm_failure_get_acked
MPIX_Comm_revoke
MPIX_Comm_shrink
MPI_2DOUBLE_PRECISION
MPI_2INT
MPI_2INTEGER
MPI_2REAL
MPI_AINT
MPI_ANY_SOURCE
MPI_ANY_TAG
MPI_APPNUM
MPI_ARGVS_NULL
MPI_ARGV_NULL
MPI_Abort
MPI_Accumulate
MPI_Add_error_class
MPI_Add_error_code
MPI_Add_error_string
MPI_Address

MPI_File_get_size
MPI_File_get_type_extent
MPI_File_get_view
MPI_File_iread
MPI_File_iread_all
MPI_File_iread_at
MPI_File_iread_at_all
MPI_File_iread_shared
MPI_File_iwrite
MPI_File_iwrite_all
MPI_File_iwrite_at
MPI_File_iwrite_at_all
MPI_File_iwrite_shared
MPI_File_open
MPI_File_preallocate
MPI_File_read
MPI_File_read_all
MPI_File_read_all_begin
MPI_File_read_all_end
MPI_File_read_at
MPI_File_read_at_all
MPI_File_read_at_all_begin
MPI_File_read_at_all_end
MPI_File_read_ordered
MPI_File_read_ordered_begin
MPI_File_read_ordered_end
MPI_File_read_shared
MPI_File_seek
MPI_File_seek_shared

MPI_SEEK_SET
MPI_SHORT
MPI_SHORT_INT
MPI_SIGNED_CHAR
MPI_SIMILAR
MPI_SOURCE
MPI_STATUSES_IGNORE
MPI_STATUS_IGNORE
MPI_SUBVERSION
MPI_SUCCESS
MPI_SUM
MPI_Scan
MPI_Scatter
MPI_Scatterv
MPI_Send
MPI_Send_init
MPI_Sendrecv
MPI_Sendrecv_replace
MPI_Ssend
MPI_Ssend_init
MPI_Start
MPI_Startall
MPI_Status_set_cancelled
MPI_Status_set_elements
MPI_Status_set_elements_x
MPI_TAG
MPI_TAG_UB
MPI_THREAD_FUNNELED
MPI_THREAD_MULTIPLE

MPI library - 2



MPI_Comm_rank

Determines the rank of the calling process in the communicator

Synopsis

```
int MPI_Comm_rank(MPI_Comm comm, int *rank)
```

Input Parameters

comm
communicator (handle)

Output Parameters

rank
rank of the calling process in the group of `comm` (integer)

Thread and Interrupt Safety

This routine is both thread- and interrupt-safe. This means that this routine may safely be used by multiple threads and from within a signal handler.

Notes for Fortran

All MPI routines in Fortran (except for `MPI_WTIME` and `MPI_WTICK`) have an additional argument `ierr` at the end of the argument list. `ierr` is an integer and has the same meaning as the return value of the routine in C. In Fortran, MPI routines are subroutines, and are invoked with the `call` statement.

All MPI objects (e.g., `MPI_Datatype`, `MPI_Comm`) are of type `INTEGER` in Fortran.

Errors

All MPI routines (except `MPI_Wtime` and `MPI_Wtick`) return an error value; C routines as the value of the function and Fortran routines in the last argument. Before the value is returned, the current MPI error handler is called. By default, this error handler aborts the MPI job. The error handler may be changed with `MPI_Comm_set_errhandler` (for communicators), `MPI_File_set_errhandler` (for files), and `MPI_Win_set_errhandler` (for RMA windows). The MPI-1 routine `MPI_Errhandler_set` may be used but its use is deprecated. The predefined error handler `MPI_ERRORS_RETURN` may be used to cause error values to be returned. Note that MPI does *not* guarentee that an MPI program can continue past an error; however, MPI implementations will attempt to continue whenever possible.

MPI_SUCCESS
No error; MPI routine completed successfully.

Program sendRecData - 1

```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main (int argc, char ** argv)
{
   int rank;
   char data[] = "I am here!",
        *recData;

   MPI_Init (&argc, &argv);
   MPI_Comm_rank (MPI_COMM_WORLD, &rank);
   if (rank == 0)
      { printf ("Transmitted message: %s \n", data);
        MPI_Send (data, strlen (data), MPI_CHAR, 1, 0, MPI_COMM_WORLD);
      }
      else if (rank == 1)
              { recData = malloc (100);
                MPI_Recv (recData, 100, MPI_CHAR, 0, 0, MPI_COMM_WORLD,
                          MPI_STATUS_IGNORE);
                printf ("Received message: %s \n", data);
              }
   MPI_Finalize ();
   return 0;
}
```

Program sendRecData - 2

```
[ruib@ruib-laptop basic1]$ ll
total 28
-rwxrwxr-x. 1 ruib ruib 19672 Apr 20 09:43 hello
-rw-rw-r--. 1 ruib ruib   295 Mar 11  2019 hello.c
-rw-rw-r--. 1 ruib ruib   665 Mar 11  2019 sendRecData.c
[ruib@ruib-laptop basic1]$ mpicc -Wall -o sendRecData sendRecData.c
[ruib@ruib-laptop basic1]$ ll
total 48
-rwxrwxr-x. 1 ruib ruib 19672 Apr 20 09:43 hello
-rw-rw-r--. 1 ruib ruib   295 Mar 11  2019 hello.c
-rwxrwxr-x. 1 ruib ruib 19824 Apr 20 09:48 sendRecData
-rw-rw-r--. 1 ruib ruib   665 Mar 11  2019 sendRecData.c
[ruib@ruib-laptop basic1]$ mpiexec -n 2 ./sendRecData
Transmitted message: I am here!
Received message: I am here!
[ruib@ruib-laptop basic1]$
```