



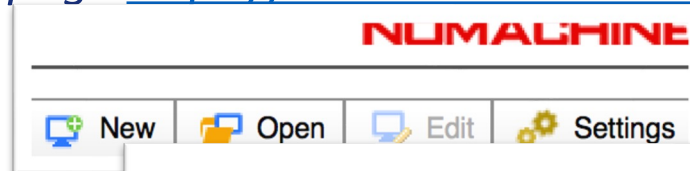
CFD on HPC - OpenFOAM

Using HPCFS

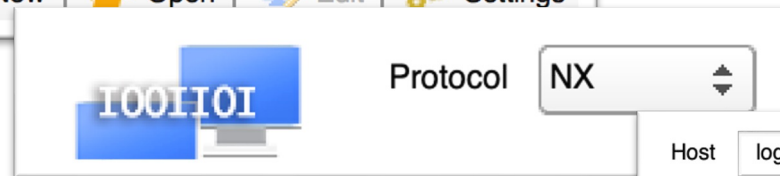
Setting up the NoMachine client

Available for installation at page <https://www.nomachine.com/download-enterprise#NoMachine-Enterprise-Client>

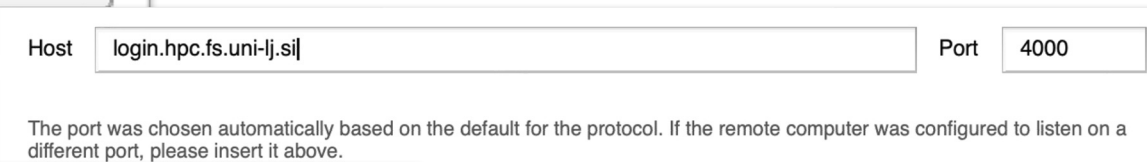
I. *Select New*



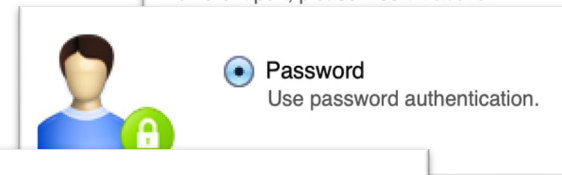
II. *Protocol NX*



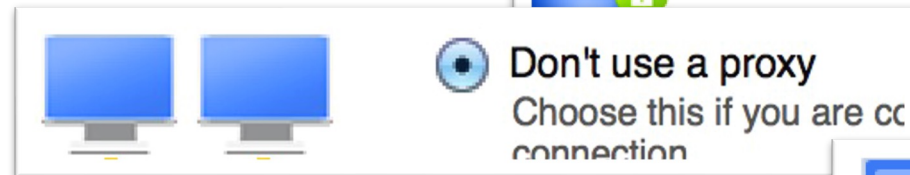
III. *Host: login.hpc.fs.uni-lj.si Port: 4000*



IV. *Use Password authentication*



V. *Don't use proxy*



VI. *Done with Connection to login.hpc.fs.uni-lj.si*



Connecting to HPCFS

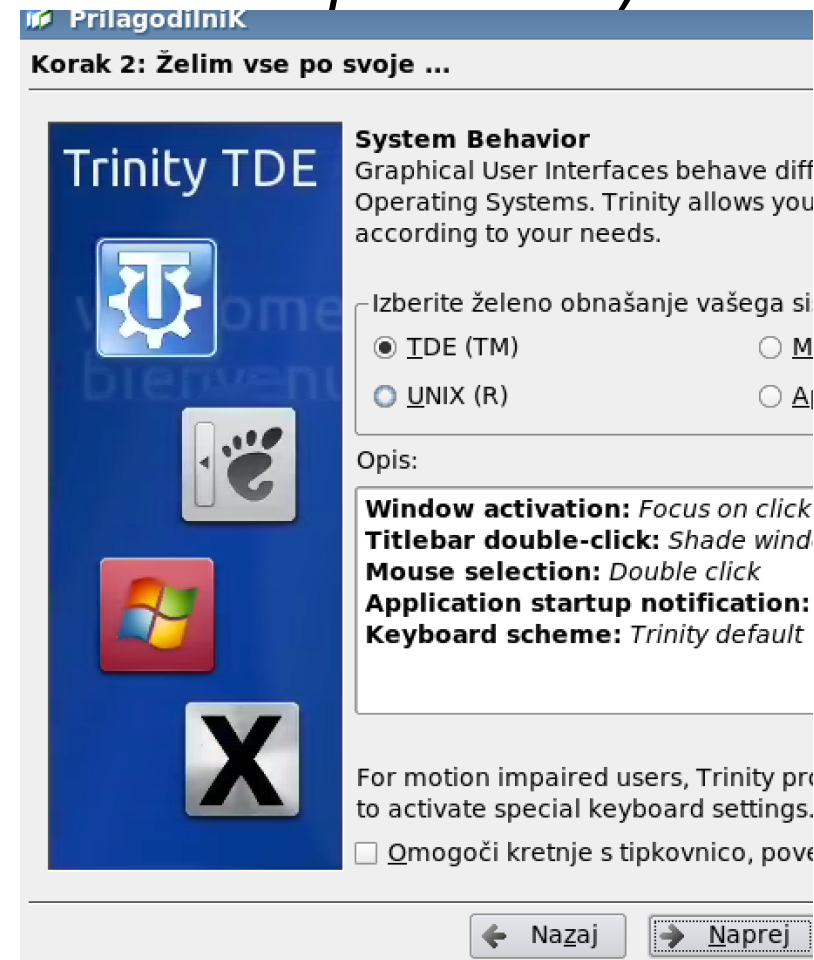
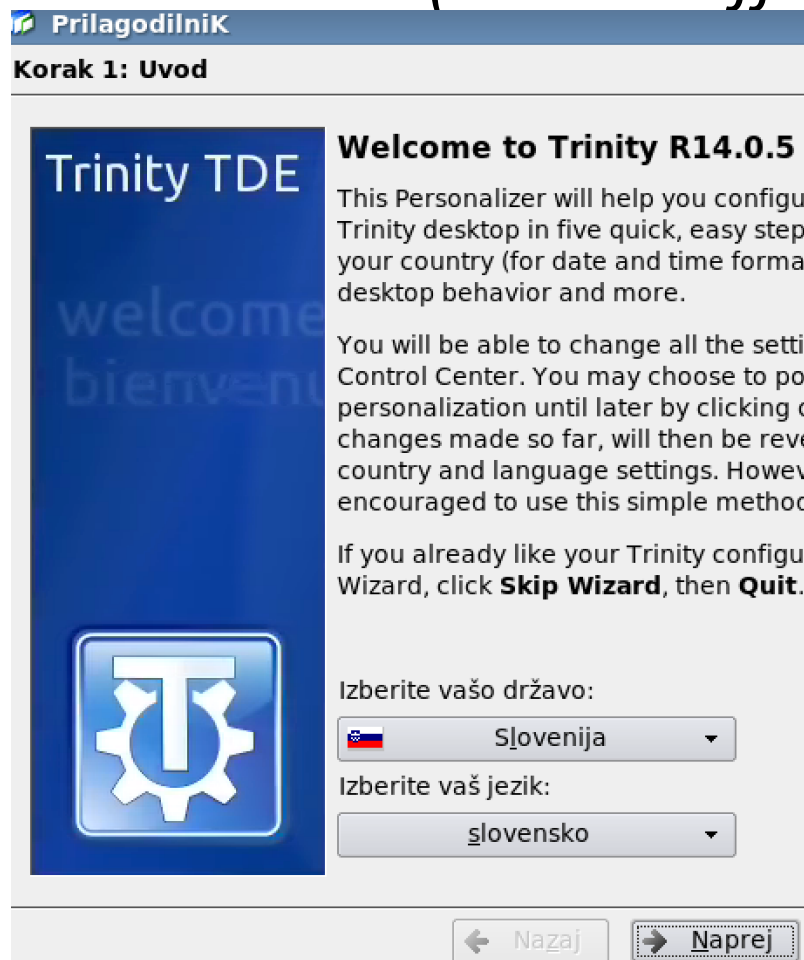
- I. *Select and Connect*
- II. Use **your** account credentials
- III. Create **New** desktop **once**
- IV. Use the Trinity (KDE) desktop
- V. To **Disconnect** press *Ctrl+Alt+T*
- VI. To **Reconnect** select previous virtual desktop

The screenshot displays the connection interface for HPCFS. It shows a list of connections, with the selected one being 'Connection to prelog.fs.uni-lj.si'. The login form includes fields for Username (campus79) and Password (masked with asterisks), and a checkbox for 'Save this password in the connection file'. A 'connect from the server' button is visible at the bottom left, and a 'Create a new virtual desktop' button is visible at the top right of the connection window.

Tuning desktop with KPersonalizer for remote speed

(use less effects=slower processor)

Use Trinity Control Center to setup colors for Non TDE programs:
 Uncheck "Enforce colors for Non-TDE programs"



Tuning desktop with KPersonalizer for remote speed

(use less effects=slower processor)

The screenshot displays the KPersonalizer application interface, which is used for customizing the Trinity TDE desktop environment. It is divided into several panels:

- Korak 3: Cukrček-meter**: A panel titled "Trinity TDE" with a "welcome Trinity" message. It includes a "Pokaži podrobnosti >>" button and a "Trinity" speedometer graphic.
- Korak 4: Vsi imajo radi teme**: A panel titled "Trinity TDE Style" with a "welcome Trinity" message. It features a table of themes and a "Ogled" (Preview) section for button groups.
- Barve [spremenjena] - Nadzorno središče**: A panel titled "Barve" (Colors) for customizing the desktop's color scheme. It includes a "Barvna shema" (Color Scheme) list, a "Barve gradnikov" (Widget Colors) section, and a "Uveljavi barve pri ne-TDE programih" checkbox.

The "Barve" panel includes a "Barvna shema" list with the following items:

- TDE privzeta
- Aqua Blue
- Aqua Graphite
- BeOS
- Bleda sivina
- Buča
- CDE
- Digitalni CDE
- Domino

The "Barve gradnikov" section includes a "Navadno ozadje" (Default Background) field and a "Šibek" (Contrast) slider.

The "Ogled" section shows a preview of button groups with the following options:

- Zavihek 1
- Zavihek 2
- Skupina gumbov
- Radijski gumb
- Radijski gumb
- Izbirno polje

Basic HPCFS cluster usage

- *Setting GNOME or KDE desktop locale preferences for keyboard, LANG environment*
- *Using NX client (Disconnect, Terminate, Logout)*
- *Console commands in Linux*
- *Editors for programming (emacs, gedit, kate, eclipse, vi, pico, ...) on login only!*

Modules (LUA)

- module avail
- module help/info
- module show
- module load/unload
- module list
- module purge

SLURM batch scheduler - Compiled-in OpenMPI support

- `srun --nodes=N --ntasks=n cmd`
- `sbatch script.sh`
- `sinfo`
- `squeue`
- Alias for interactive usage of nodes:
`alias node='srun -N1 --time=1:00:00 --pty bash -i'`

Using SLURM (interactively) and Message Passing Interface (MPI)

```
[leon@viz mpi]$ module purge && module load foss/2019a
[leon@viz mpi]$ cat hello.f90
program hello
    use mpi
    integer rank, size, ierror, strlen, status(MPI_STATUS_SIZE)
    character(len=MPI_MAX_PROCESSOR_NAME) :: hostname

    call MPI_INIT(ierror)
    call MPI_COMM_SIZE(MPI_COMM_WORLD, size, ierror)
    call MPI_COMM_RANK(MPI_COMM_WORLD, rank, ierror)
    call MPI_GET_PROCESSOR_NAME( hostname, strlen, ierror )
    print*, trim(hostname), rank, size
    call MPI_FINALIZE(ierror)
end
[leon@viz mpi]$ mpif90 hello.f90
[leon@viz mpi]$ LD_PRELOAD= srun -n 4 --tasks-per-node=2 --kill-on-badexit
--partition=haswell ./a.out
cn80      2      4
cn79      0      4
cn80      3      4
cn79      1      4
```

OpenMP

```
#include <stdio.h>
#include <math.h>
#define N 1000000
int main()
{
    double area = 0.0;
    #pragma omp parallel for reduction(+:area)
    for(int i = 0; i < N; i++)
    {
        double x = (i+0.5)/N;
        area += sqrt(1.0 - x*x);
    }
    printf("Pi : %14lf\n", 4.0*area/N);
    return 0;
}
```

```
[leon@cn36 pi]$ module purge && module load foss/2019a
```

```
[leon@cn36 pi]$ gcc -fopenmp pi-openmp.c -lm -o pi-openmp
```

```
[leon@cn36 pi]$ OMP_NUM_THREADS=4 ./pi-openmp
```




Thank you for attention!



EuroHPC
Joint Undertaking



REPUBLIC OF SLOVENIA
**MINISTRY OF HIGHER EDUCATION,
SCIENCE AND INNOVATION**

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia.