In this post, we demonstrate how to install, register with FlexiBLAS, and finally switch to Intel's Math Kernel Library (MKL) in a few steps. First, we prepare a proper environment using docker:

```
$ docker run --rm -it fedora:33
$ dnf install 'dnf-command(config-manager)' # install config manager
$ dnf install R-flexiblas # install R and the FlexiBLAS API interface
for R
```

Then we add Intel's YUM repository, import the public key and install MKL:

```
$ dnf config-manager --add-repo https://yum.repos.intel.com/mkl/setup/intel-mkl.repo
$ rpm --import https://yum.repos.intel.com/intel-gpg-keys/GPG-PUB-KEY-INTEL-SW-
PRODUCTS-2019.PUB
$ dnf install intel-mkl # or a specific version, e.g. intel-
mkl-2020.0-088
```

Then, in an R session:

```
library(flexiblas)

flexiblas_load_backend("/opt/intel/mkl/lib/intel64/libmkl_rt.so")
#> flexiblas BLAS /opt/intel/mkl/lib/intel64/libmkl_rt.so not found in
config.
#>  BLAS /opt/intel/mkl/lib/intel64/libmkl_rt.so does not provide an
integer size hint. Assuming 4 Byte.
#> [1] 2

backends <- flexiblas_list_loaded()
backends
#> [1] "OPENBLAS-OPENMP"
#> [2] "/opt/intel/mkl/lib/intel64/libmkl_rt.so"
```

And that's it: now, we are able to switch between the default one and MKL. As in our previous post, let's compare them with a simple GEMM benchmark:

```
n <- 2000
runs <- 10

A <- matrix(runif(n*n), nrow=n)
B <- matrix(runif(n*n), nrow=n)

# benchmark
timings <- sapply(seq_along(backends), function(i) {
  flexiblas_switch(i)

  # warm-up
  C <- A[1:100, 1:100] %*% B[1:100, 1:100]

  unname(system.time({
    for (j in seq_len(runs))
      C <- A %*% B
  })[3])
})

results <- data.frame(
  backend = backends,
  `timing [s]` = timings,
  `performance [GFlops]` = (2 * (n / 1000)^3) / timings,
  check.names = FALSE)

results[order(results$performance),]
#>                                    backend timing [s] performance
[GFlops]
#> 2 /opt/intel/mkl/lib/intel64/libmkl_rt.so      3.487
4.588471
#> 1                         OPENBLAS-OPENMP      0.754
21.220159
```

And still OpenBLAS rocks!