Linux サーバ版 GAMESS インストールマニュアル

2024 年 12 月 26 日

本マニュアルの目的

本マニュアルでは、単独ユーザが独占的に Linux サーバ（Rocky Linux 9.4）を使用して GAMESS 2021 のジョブを並列実行するための環境構築方法と Winmostar のリモートジョブ投入機能から Linux サーバへジョブを投入する方法を示しています。計算環境は全てユーザのホームディレクトリ配下で行うことを想定しています。複数ユーザが使用する共用サーバの環境を構築する方法、複数ノードを利用する環境を構築する方法などは本マニュアルに含まれませんので、ハードウェアベンダにお問い合わせください。

また、本マニュアルではサーバホスト名は「remote_server」、ユーザ名は「winmostar_user」であると仮定します。

1. コンパイラの確認・インストール

① Linux サーバにログインし、以下のコマンドで make がインストールされているか確認する。

```
$ which make
```

インストールされていない場合は dnf などのパッケージ管理ソフトウェアを用いてインストールする。Rocky 9.4 で dnf を使う場合は以下のように実行する。

```
$ sudo dnf install make
```

② 以下のコマンドで gfortran、 g++がインストールされているか確認する。

```
$ which gfortran
$ which g++
```

インストールされていない場合は dnf などのパッケージ管理ソフトウェアを用いてインストールする。Rocky 9.4 で dnf を使う場合は以下のように実行する。

```
$ sudo dnf install gcc
$ sudo dnf install gcc-c++
$ sudo dnf install gcc-gfortran
```

③ 以下のコマンドで tcsh がインストールされているか確認する。

```
$ which tcsh
```

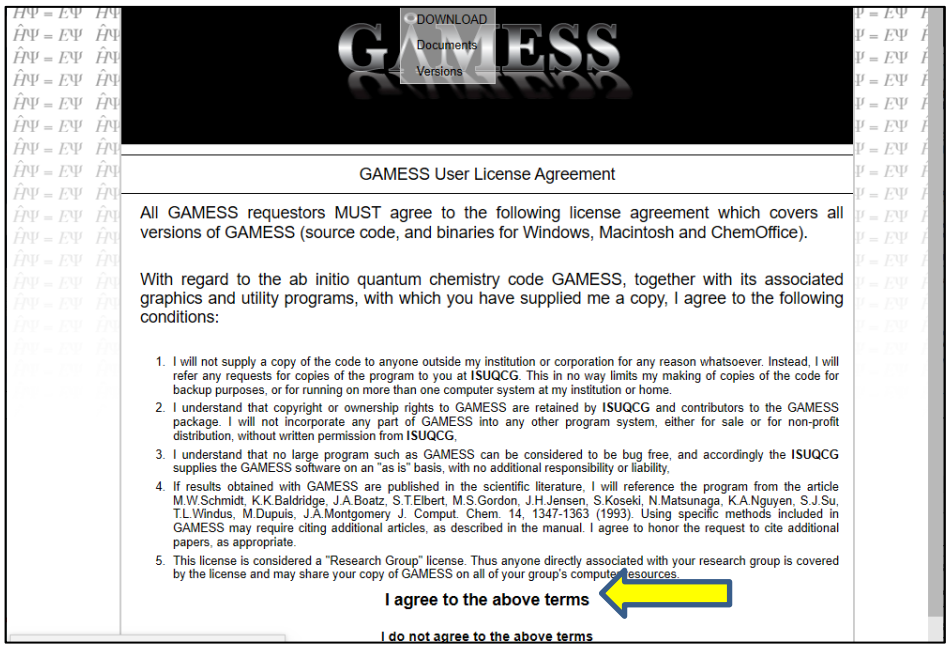
インストールされていない場合は dnf などのパッケージ管理ソフトウェアを用いてインストールする。Rocky 9.4 で dnf を使う場合は以下のように実行する。

```
$ sudo dnf install tcsh
```

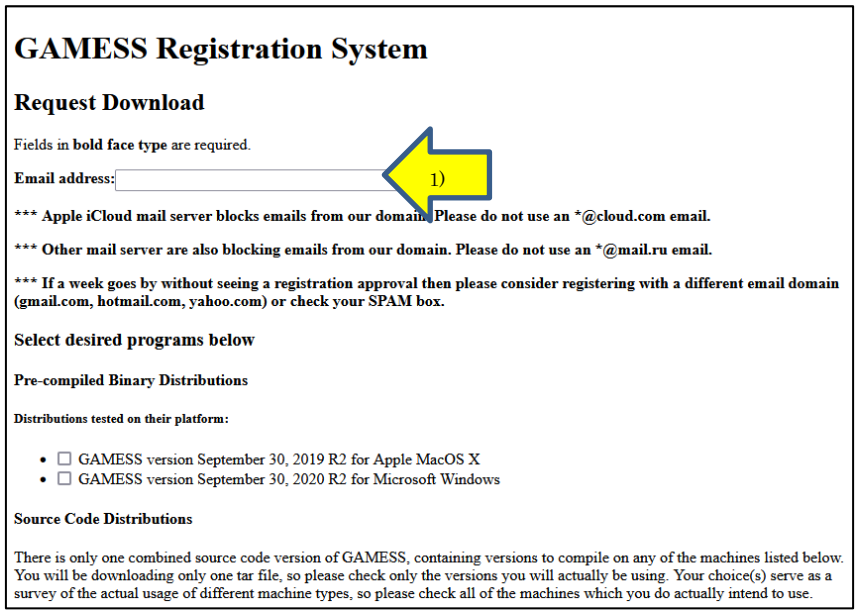
2. GAMESS のインストール

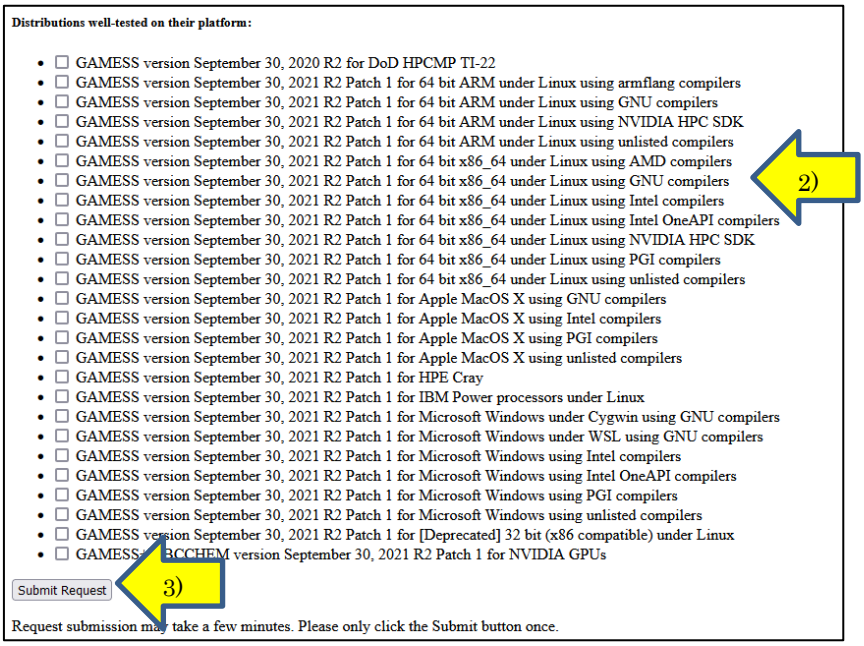
① GAMESS のサイト内の[GAMESS User License Agreement]

https://www.msg.chem.iastate.edu/gamess/License_Agreement.html にブラウザを用いてアクセスする。同意できる場合は”**I agree to the above terms**”をクリックする。サーバのメンテナンスにより数日間アクセスできないことがあるので、その場合はアクセスできるようになるまで待つ。

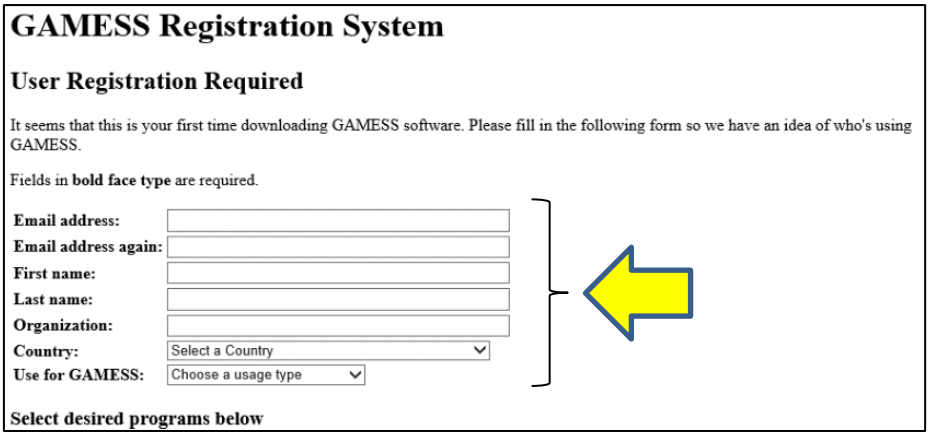


次に、**Email address** にメールアドレスを入力し、**GAMESS version .....for 64 bit x86_64 under Linux with GNU compilers** にチェックを入れ（64 bit Linux の場合）、**Submit Request** をクリックする。

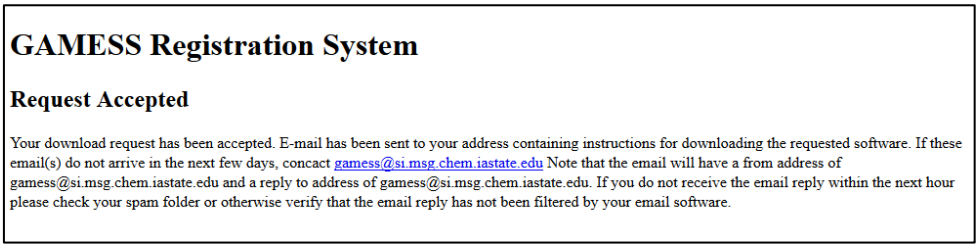




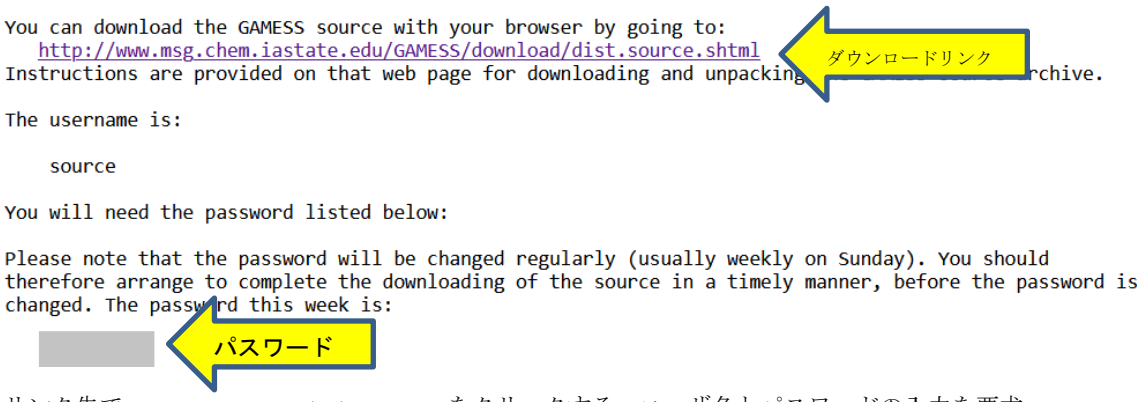
初回は登録フォームが表示される。太字の欄をすべて入力し、再度 **Submit Request** をクリックする。もしエラーが出て登録フォームが表示されない場合、他のブラウザで試してください。Microsoft Edge の場合は、Internet Explorer モードでアクセスし直してください。



以下のような受付完了の画面が表示される。



② 以下のようなメールが届くのを待つ（数日程度掛かる可能性がある）。届いたら、メールに記載されているパスワードをメモした上でダウンロードリンクをクリックする。



リンク先で `games-current.tar.gz` をクリックする。ユーザ名とパスワードの入力を要求されるので、以下のように入力する。

ユーザ名： `source`

パスワード：（メールに記述されたもの）

④ `gamess.Sep302021R2.tar.gz` を SCP, FTP などで Linux サーバに転送する。ここではホームディレクトリ直下に置いたと仮定する。

⑤ Linux サーバにログインし、以下のコマンドで圧縮ファイルを解凍する。

```
$ tar  xzvf  gamess.Sep302021R2.tar.gz
```

⑥ 使用するコンパイラのバージョンを調べてメモする。本書の手順に従いコンパイラに gfortran を利用する場合は以下のコマンドを実行する。

```
$ gfortran  -v
```

⑦　以下のコマンドで GAMESS のビルド設定を開始する。

```
$ cd  ~/gamess
```

```
$ ./config
```

以下の画面が表示されたら[Return]キーを押す。

```
[winmostar_user@remote_server]$ ./config
This script asks a few questions, depending on your computer system,
to set up compiler names, libraries, message passing libraries,
and so forth.
You can quit at any time by pressing control-C, and then <return>.

Please open a second window by logging into your target machine,
in case this script asks you to 'type' a command to learn something
about your system software situation.   All such extra questions will
use the word 'type' to indicate it is a command for the other window.

After the new window is open, please hit <return> to go on.
```

以下の画面が表示されたら「linux64」と入力する。

```
     GAMESS can compile on the following 32 bit or 64 bit machines:
hpe-cray-ex   - HPE Cray's EX architecture (e.g., narwhal, spock)
hpe-cray-cs   - HPE Cray's CS architecture (e.g., tulip)
cray-xt       - Cray's XT massively parallel system, running CNL
cray-xc       - Cray's XC massively parallel system (e.g., DoE Theta, DoD Onyx)
fj-a64fx      - Fujitsu PRIMEHPC with A64FX cpus and Fujitsu Compilers (e.g., FX1000, FX700)
ibm64          - IBM, Power8 chip or newer, running AIX or Linux (e.g., DoE Summit, DoD Hokulea)
linux32        - Linux (any 32 bit distribution), for x86
linux64        - Linux (any 64 bit distribution), for x86_64, ia64, or arm64 chips,
                   using gfortran, ifort, or perhaps PGI compilers.
mac64          - Apple Mac, any chip, running OS X 10.5 or newer
win64          - Windows 64-bit (Windows 10)
singularity   - GAMESS Singularity container image

      type 'uname -a' to partially clarify your computer's flavor.

please enter your target machine name: linux64
```

以下の画面が表示されたら[Return]キーを押す。

```
Where is the GAMESS software on your system?
A typical response might be /u1/mike/gamess,
most probably the correct answer is /home/ winmostar_user /gamess

GAMESS directory? [/home/winmostar_user/gamess]
```

以下の画面が表示されたら[Return]キーを押す。

```
Setting up GAMESS compile and link for GMS_TARGET=linux64
GAMESS software is located at GMS_PATH=/home/winmostar_user/gamess

Please provide the name of the build location.
This may be the same location as the GAMESS directory.

GAMESS build directory? [/home/winmostar_user/games]
```

以下の画面が表示されたら[Return]キーを押す。

```
Please provide a version number for the GAMESS executable.
This will be used as the middle part of the binary's name,
for example: gamess.00.x

Version? [00]
```

以下の画面が表示されたら「gfortran」と入力する。

```
Linux FORTRAN compiler selection
================================

aocc            - AMD Fortran compiler
armflag         - ARM Fortran compiler
gfortran        - GNU Fortran compiler                      (minimum support v. GCC-5)
ifort           - Intel Fortran compiler                    (minimum support v. 2018.3)
oneapi-ifort    - Intel OneAPI Classic Fortran compiler     (minimum support v. 2021.3.0)
oneapi-ifx      - Intel OneAPI Beta Fortran compiler        (minimum support v. 2021.3.0)
pgfortran       - NVIDIA PGI Fortran compiler               (minimum support v. 19.10
Community)
nvfortran       - NVIDIA HPC SDK Fortran compiler           (minimum support v. 21.7-0)

Please enter your choice of FORTRAN:: gfortran
```

以下の画面が表示されたら 2. ⑤で調べたコンパイラのバージョンを入力する（gfortran の場合）。1. ②の手順でインストールした場合は「4.8」と入力する。

```
gfortran is very robust, so this is a wise choice.

Please type 'gfortran -dumpversion' or else 'gfortran -v' to
detect the version number of your gfortran.
This reply should be a string with at least two decimal points,
such as 4.9.4 or 6.3.0.
The reply may be labeled as a 'gcc' version,
but it is really your gfortran version.

Please enter only the first decimal place, such as 4.9: 4.8
```

以下の画面が表示されたら[Return]キーを押す。

```
Please enter only the first decimal place, such as 4.9: 4.8
    GAMESS is old but your GNU compiler shouldn't be.
    Might be time to upgrade.
Hit <ENTER> to continue to the math library setup.
```

以下の画面が表示されたら「none」と入力する[1]。その後、 [Return]キーを押す。

```
Linux distributions do not include a standard math library.

There are several reasonable add-on library choices,
          MKL from Intel                for 32 or 64 bit Linux (very fast)
         ACML from AMD                  for 32 or 64 bit Linux (free)
       LibFLAME from AMD                for 64 bit Linux (free)
       ATLAS from www.rpmfind.net for 32 or 64 bit Linux (free)
   PGI BLAS from Portland Group    for 32 or 64 bit Linux
       ArmPL from ARM                 for 64 bit Linux
and one very unreasonable option, namely 'none', which will use
some slow FORTRAN routines supplied with GAMESS.    Choosing 'none'
will run MP2 jobs 2x slower, or CCSD(T) jobs 5x slower.

Some typical places (but not the only ones) to find math libraries are
Type 'echo $MKLROOT'                          to look for MKL
Type 'ls -d /opt/acml*'                       to look for ACML
Type 'ls -d /usr/local/acml*'                 to look for ACML
Type 'ls /usr/lib64/atlas'                    to look for Atlas
Type 'ls /opt/pgi/linux86-64/*/lib/*          to look for libblas.a from PGI
Type 'ls /opt/pgi/osx86-64/*/lib/*            to look for libblas.a from PGI
Type 'echo $ARMPL_DIR'                        to look for ArmPL

Enter your math library choice from one of the options below:

'acml', 'atlas', 'libflame', 'mkl', 'openblas', 'pgiblas', 'armpl', 'none'

: none
```

以下の画面が表示されたら「sockets」と入力する。

```
If you have a slow network, like Gigabit Ethernet (GE), or
if you have so few nodes you won't run extensively in parallel, or
if you have no MPI library installed, or
if you want a fail-safe compile/link and easy execution,
        choose 'sockets'
to use good old reliable standard TCP/IP networking.

If you have an expensive but fast network like Infiniband (IB), and
if you have an MPI library correctly installed,
        choose 'mpi'.

If you wish to use a combination of TCP/IP networking for small
messages and MPI for large messages in a 'mixed' fashion,
        choose 'mixed'.

communication library ('serial','sockets' or 'mpi' or 'mixed')? sockets
```

以後、表示される各種の質問について全て「no」と入力する。（LibXC から NBO Option まで）そして、下記メッセージが出力されればビルドの設定は完了し install.info に情報が書き込まれる。

```
Your configuration for GAMESS compilation is now in
        /home/winmostar_user/gamess/install.info
Now, please follow the directions in
        /home/winmostar_user/gamess/machines/readme.unix
```

[1] GAMESS2022 バージョンでは math library に none を選択すると cmake 3.2 以降が必要となります。

⑧　以下のコマンドを実行し GAMESS をビルドする。make が正常終了すると、実行ファイルの gamess.00.x が生成される。生成されていない場合は make.log を確認しエラーの内容を確認する。

```
$ make  2>&1 | tee  make.log
```

⑨　gamess ディレクトリ以下の rungms の内容を以下のように修正する。GAMESS をビルドしたディレクトリの名前を gamess から変更した場合は rungms 内の set GMSPATH の行の値も追加で修正する（本書の手順に従った場合、GMSPATH の変更は不要)。

```
set SCR=~/gamess/restart
set USERSCR=~/gamess/restart
```

↓

```
set SCR=$HOME/scr
set USERSCR=$HOME/scr
```

⑩　以下のコマンドを実行しユーザのホームディレクトリに scr ディレクトリを作成する。

```
$ mkdir  $HOME/scr
```

⑪　動作確認のため以下のコマンドでテスト計算を実行する。

⑫　

```
$ rm ~/scr/*
$ ./runall  00
```

終了後、テスト結果を確認するために次のコマンドを実行する。

```
$ ./tests/standard/checktst
```

最後に”All 48 test results are correct!”と表示されていれば全て正常に計算されたことを意味する。

```
Checking the results of your sample GAMESS calculations,
the output files (exam??.log) will be taken from .
All jobs terminated normally, now checking detailed numerical results
exam01: Eerr=0.0e+00 Gerr=0.0e+00.                                  Passed.
exam02: Eerr=0.0e+00 Gerr=0.0e+00 Serr=0.0e+00 Lerr=1.8e-03+6.6e-05.  Passed.
(省略)
exam47: Serr=0.0e+00 Perr=6.0e-10.                                  Passed.
exam48: E0err=0.0e+00 E1err=0.0e+00 Gerr=0.0e+00.                   Passed.
All 48 test results are correct!
```

テストに失敗して exam01.log に

```
DDI Process 0: shmget returned an error.
Error EINVAL: Attempting to create 160525768 bytes of shared memory.
Check system limits on the size of SysV shared memory segments.
```

と書かれている場合は、sudo で/etc/sysctl.conf に

```
kernel.shmmax = （物理メモリサイズを byte 単位で記入）
kernel.shmall = （kernel.shmmax÷4096 の値を記入）
```

と追記または変更し`$ sudo sysctl -p`と実行する。

3. Linux サーバ上での最終確認

① Winmostar から接続するときの状況を再現するため、Linux サーバにログインしなおす。

② GAMESS の動作を確認するため、GAMESS の起動に必要な環境を設定するコマンドを入力しメモに控えておく。本書の手順に従った場合は、以下のコマンドを入力する。

```
$ export PATH=$HOME/gamess:$PATH
```

③ 以下のコマンドを実行してインストールした GAMESS のパスが表示されることを確認する。確認できない場合は②の設定を見直す。

```
$ which rungms
```

4. Winmostar からの設定および動作確認

ユーザマニュアルの 7.2. リモートジョブの設定手順に従って設定し動作確認を行う。

テンプレートスクリプトを編集する際には、「# Insert commands here」 から「# Do not modify the followings」の間に 3-②でメモに控えた内容を追記する。本書の手順に従った場合は以下のようになる。

```
echo "************************************************"
echo "***       Set user-defined variables         ***"
echo "************************************************"
set -v
# Insert commands here

export PATH=$HOME/gamess:$PATH

# Do not modify the followings
```

以上