2023年2月版

SSH接続とコンパイル環境の整備

更新履歴（のようなもの）

- 2022年4月：2021年7月版に対して最終ページのみ微修正
- 2022年6月：MobaXtermにおけるSSH鍵形式の変換の説明とlocal terminalからのssh接続についての情報を追加
- 2023年2月：VSCodeを用いた「不老」へのアクセスについての情報を追加

この資料で説明している内容（目次）

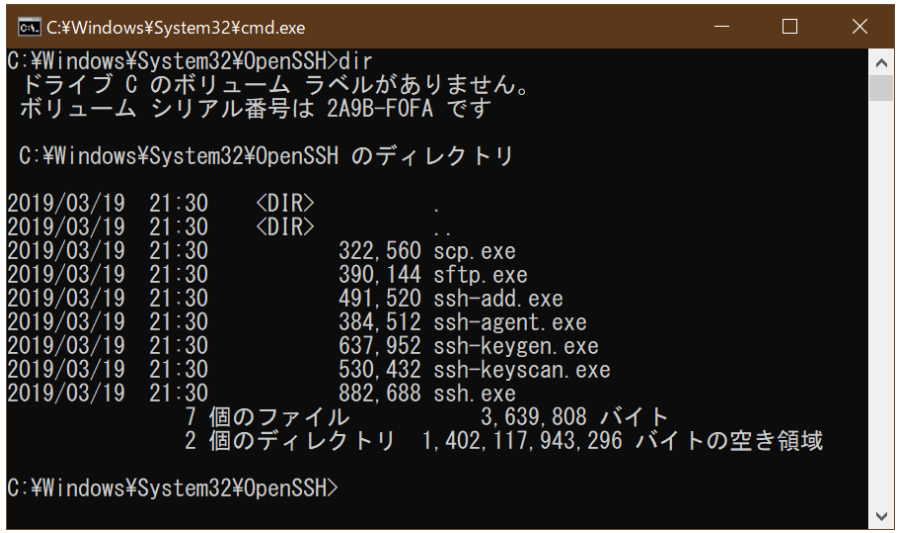
- SSH接続環境とC/C++, Fortranプログラムのコンパイル環境（開発ツール）の準備方法
  - Linuxを使う場合
    - SSH接続環境：CentOSやUbuntuを普通にインストールすれば使えるはず。入っていなければyumやaptでopensshをインストールする。
    - 開発ツール：CentOSではyum groupinstall "Development Tools"、Ubuntuではapt install "build-essential"で入るはず。もちろんgccなどを個別にinstallしても良い。
  - Macを使う場合
    - SSH接続環境：デフォルトでTerminal上からsshコマンドが使えるはず
    - 開発ツール：Xcodeやhomebrewを使う（詳細は省略）
  - Windowsを使う場合
    - 対応するソフトウェアをインストールする必要がある　→　次ページを参照

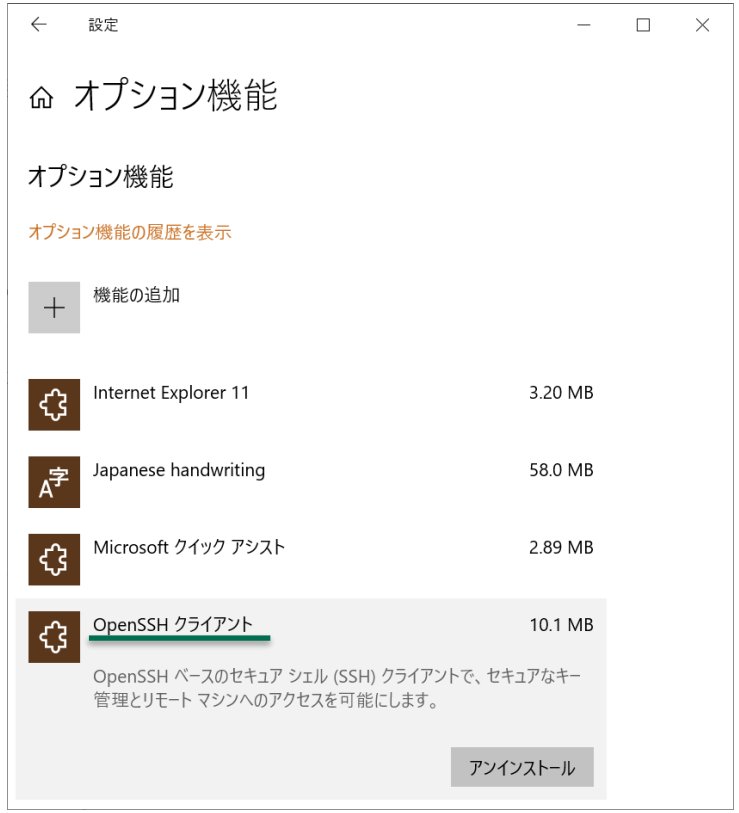
この資料で説明しているWindows向けソフトウェア

- MobaXterm
  - SSH接続とファイルの送受信が行えるため、これを使うのを推奨。
  - 手元で使える開発ツールも整備できる（実はPutty+CygwinにGUIをつけたようなもの）
- Putty
  - 長く使われている
  - SSH接続してスパコン上で作業をするだけならこれで良い。ファイルの送受信には別のソフトが必要。あくまで接続用のソフトウェアであり、手元で使える開発ツールは付いていない。
- Cygwin
  - Window上で使えるLinuxのようなもの
  - 対応するソフトウェアを入れればほぼフルのLinuxとして使える
- WSL, WSL2
  - Windows上で動作するLinux。まだ新しく利用できない環境もあるため今回は扱わない。
  - 使う場合はexport LANG=ja_JP.UTF-8しておくことを推奨
- VSCode (Visual Studio Code)
  - コードエディタ・統合開発環境だが、拡張機能を使えばssh接続・遠隔ファイル編集なども可能

補足

- 実は、最近のWindows（Windows10、2018年4月のアップデート以降）は標準でsshコマンドが利用可能
  - 使い方がわかっている人はこれを使っても良い
    - cmd.exeが使いにくいのでpowershellとあわせて使うと良い
  - 利用者により状況が異なるため、本資料では扱いません





パスワード認証と公開鍵認証に関する基礎知識

- よく使われるSSHの認証方法はパスワード認証と公開鍵認証
  - パスワード認証：ユーザIDとパスワードを使う
  - 公開鍵認証：ユーザIDと鍵ファイルを使う
  - 「不老」に限らず、多くのスパコンへの接続には公開鍵認証を使う

- 秘密鍵と公開鍵
  - 秘密鍵
    - 他人に絶対に見せてはいけない情報。秘密鍵を使うときに使うパスワードをパスフレーズと呼ぶ。パスフレーズなしの秘密鍵を作ることもできるが、セキュリティ的に絶対にパスフレーズなしにしてはいけない。
    - LinuxやCygwinではid_rsaというファイル名が基本。MobaXtermやPuttyではid_rsa.ppkというファイル名が基本。Linux/CygwinとMobaXterm/Puttyの秘密鍵は異なる形式だが変換は可能。
  - 公開鍵
    - 接続先のホストに設置する情報。接続先の~/.ssh/authorized_keysというファイルに書き込んで使う。id_rsa.pubというファイル名が基本。

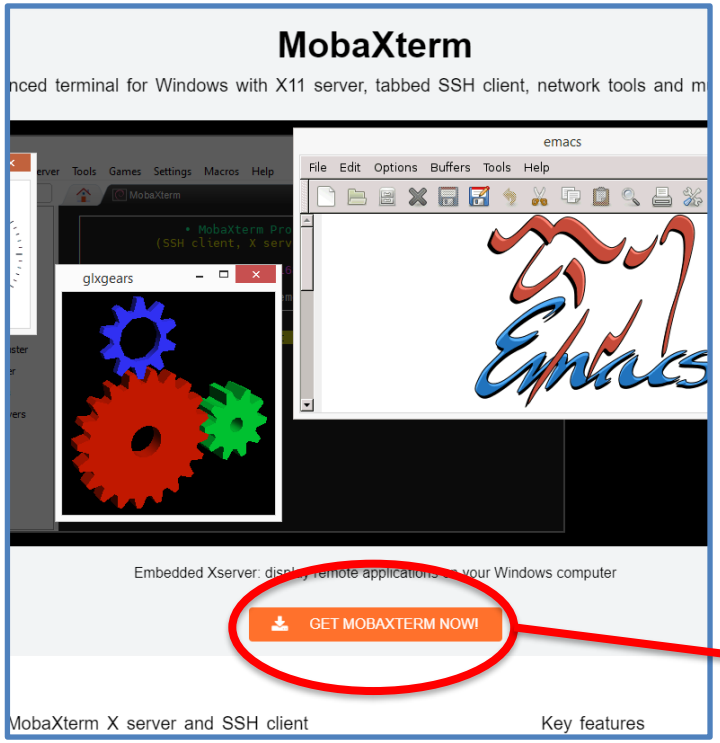
MobaXtermの使い方

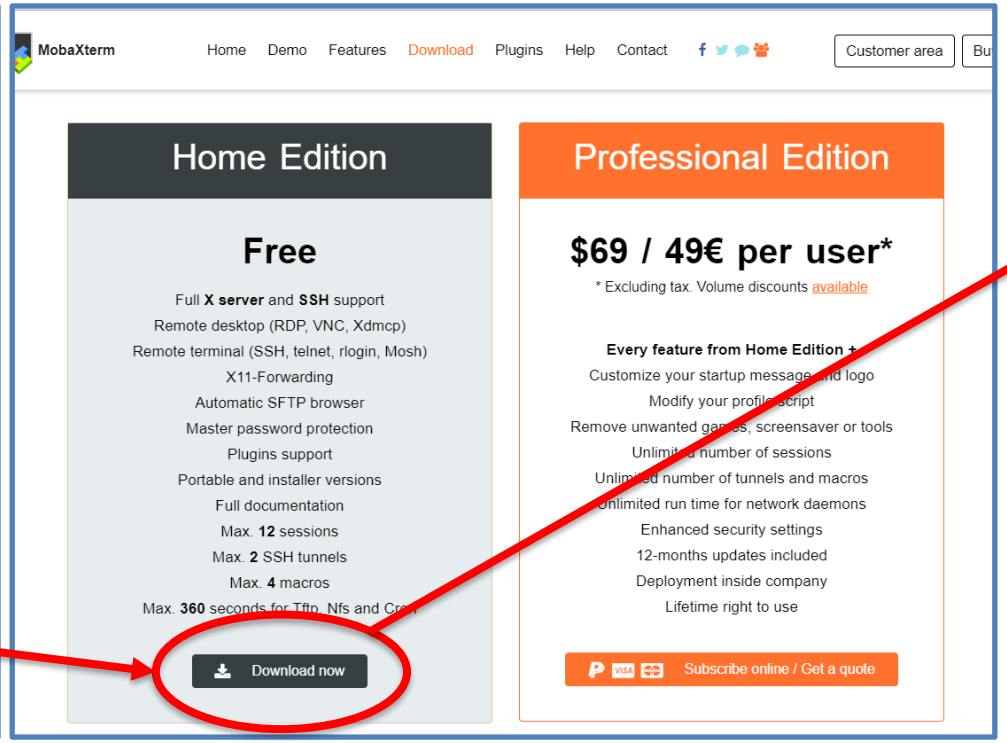
MobaXtermの準備

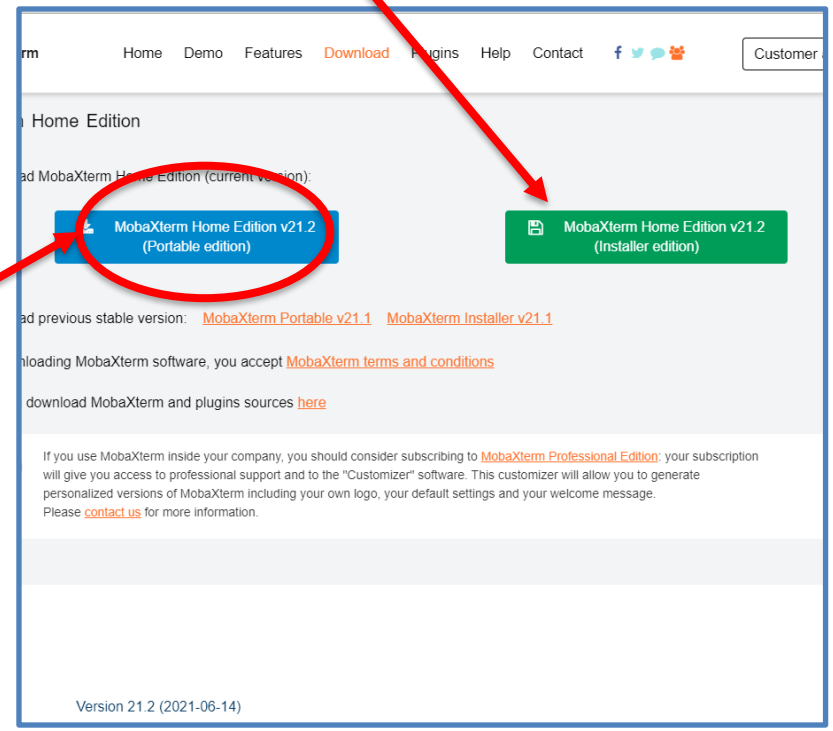
21.2は本ページ作成時点でのバージョン。更新されたら読み替えること。

- https://mobaxterm.mobatek.net/ からダウンロードする
  - 以下の手順で選んでいくと MobaXterm_Portable_v21.2.zip がダウンロードできる
  - ダウンロードしたファイルを右クリック→「すべて展開」
  - その中の MobaXterm_Personal_21.2.exe を起動する

こちらからはインストーラーが入手できる。こちらでももちろん良い。（インストールするとスタートメニューへの登録などが行われる。）



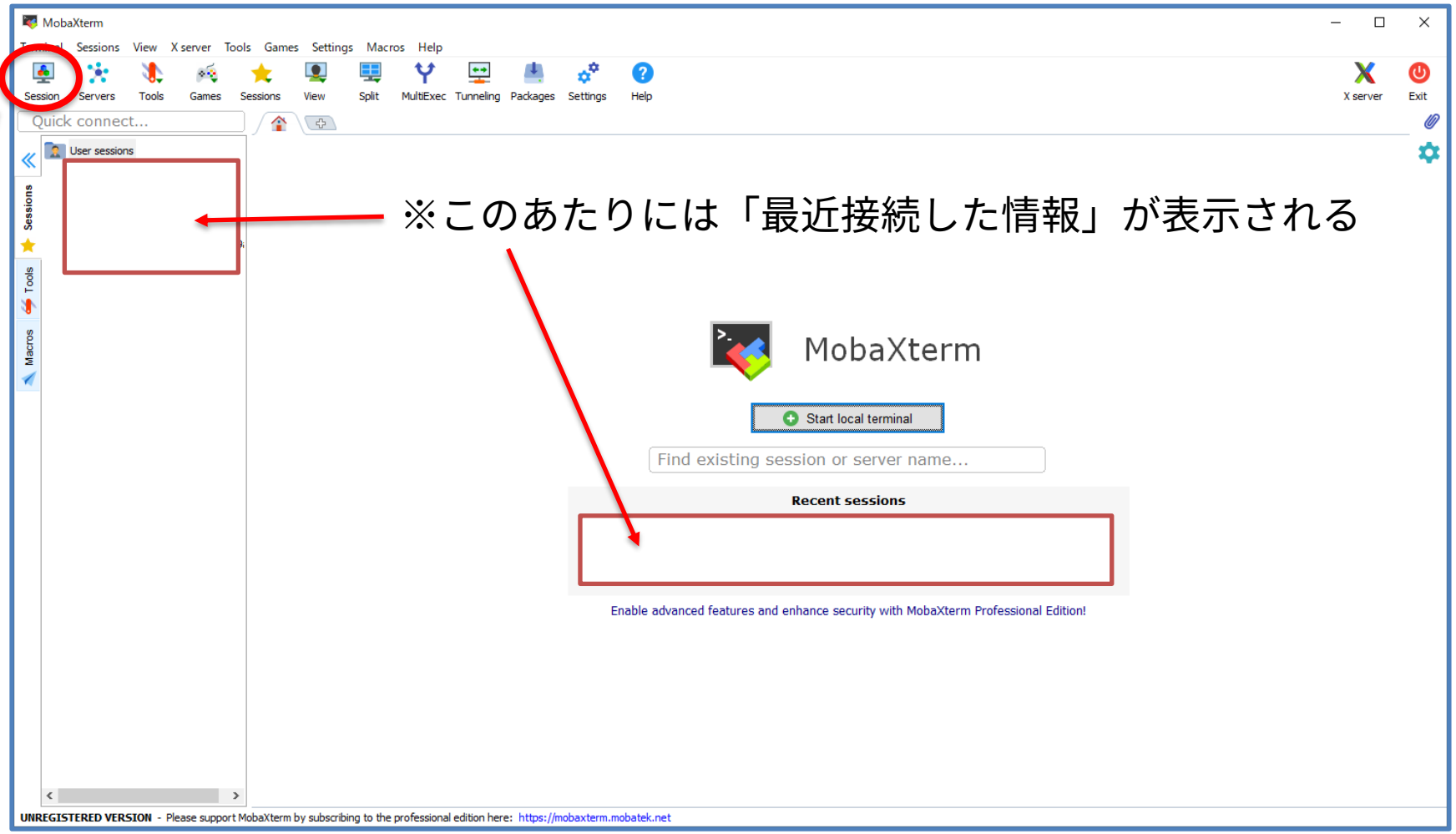




MobaXtermを起動したところ

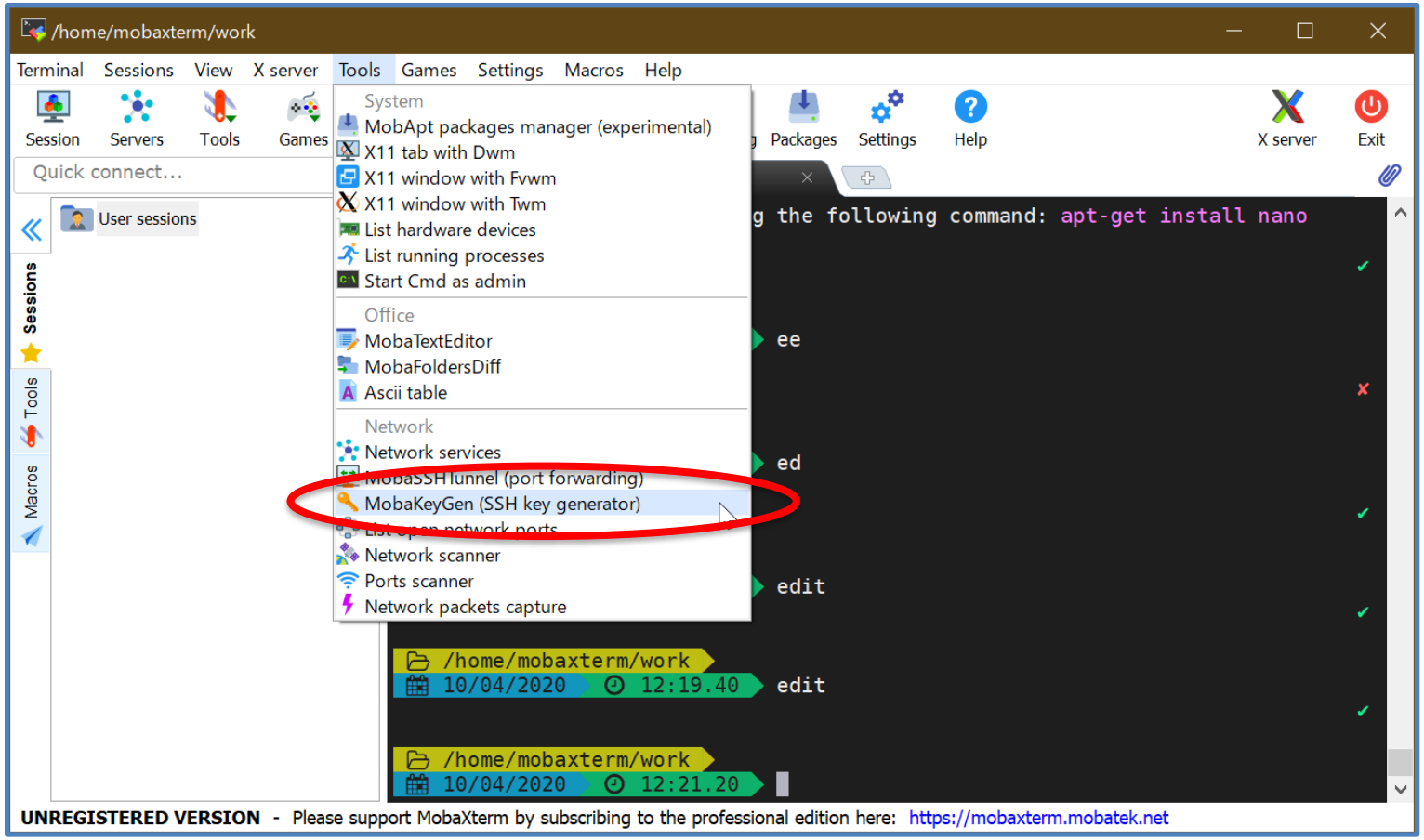
ここからSSH接続を開始する

※このあたりには「最近接続した情報」が表示される



MobaXtermを使ったSSH接続の手順：公開鍵認証を使う場合

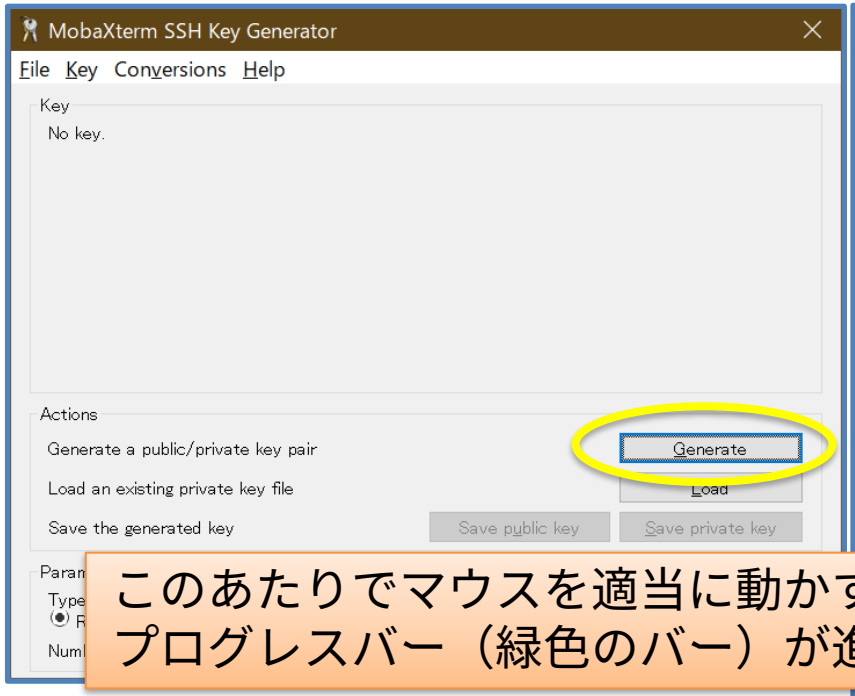
- 接続前に鍵ファイルを作成する
  - 「Tools」-「MobaKeyGen」から鍵を生成する
  - 生成方法は次ページ

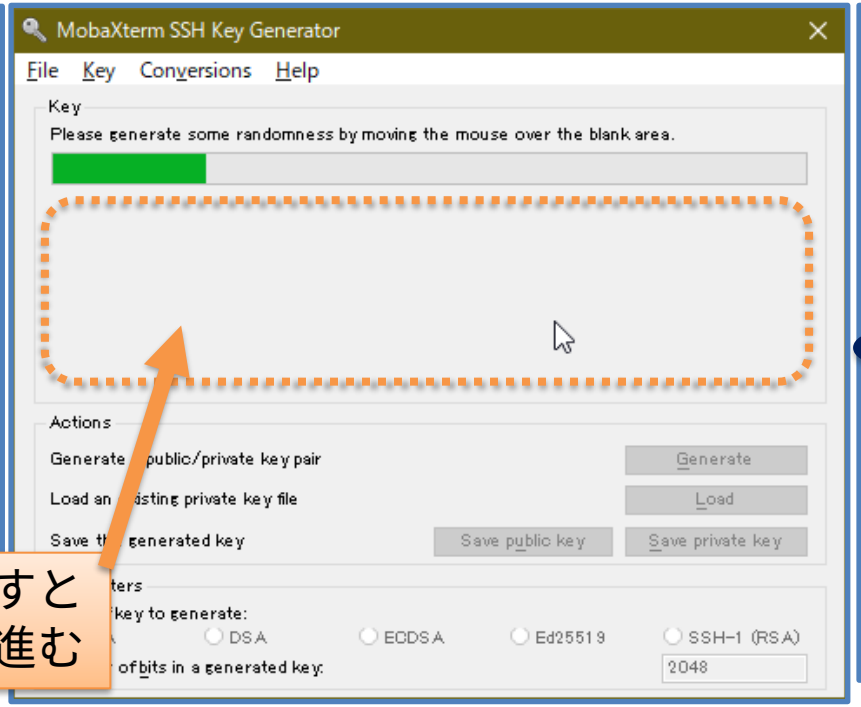


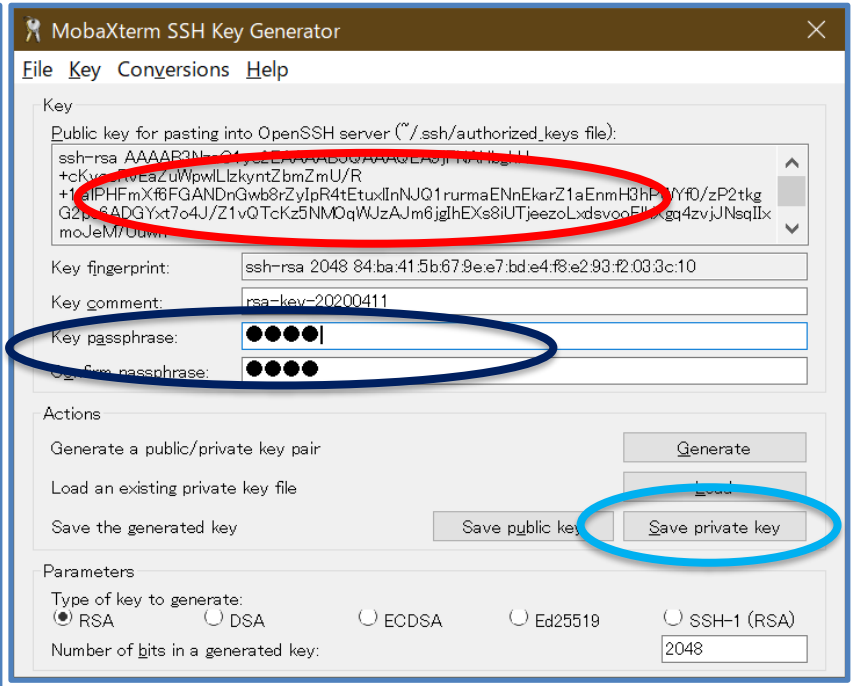
鍵ファイルの生成方法

※ 明確な制限はないが、万が一ファイルが流出しても問題が起きにくいように、短すぎるものや簡単過ぎるものは避ける

- 「generate」ボタンを押す　→　マウスを動かす
- 「passphrase（パスフレーズ）」（鍵を使うためのパスワード、自分で決める）を入力し「Save private key（秘密鍵を保存）」ボタンで拡張子ppkのファイルを保存する
- 「Public key（公開鍵）」をどこかに保存しておく
  - これは接続先ホストに書き込む情報（「不老」ではHPCポータルから登録する、実際には~/.ssh/authorized_keysファイルに書き込まれることになる）



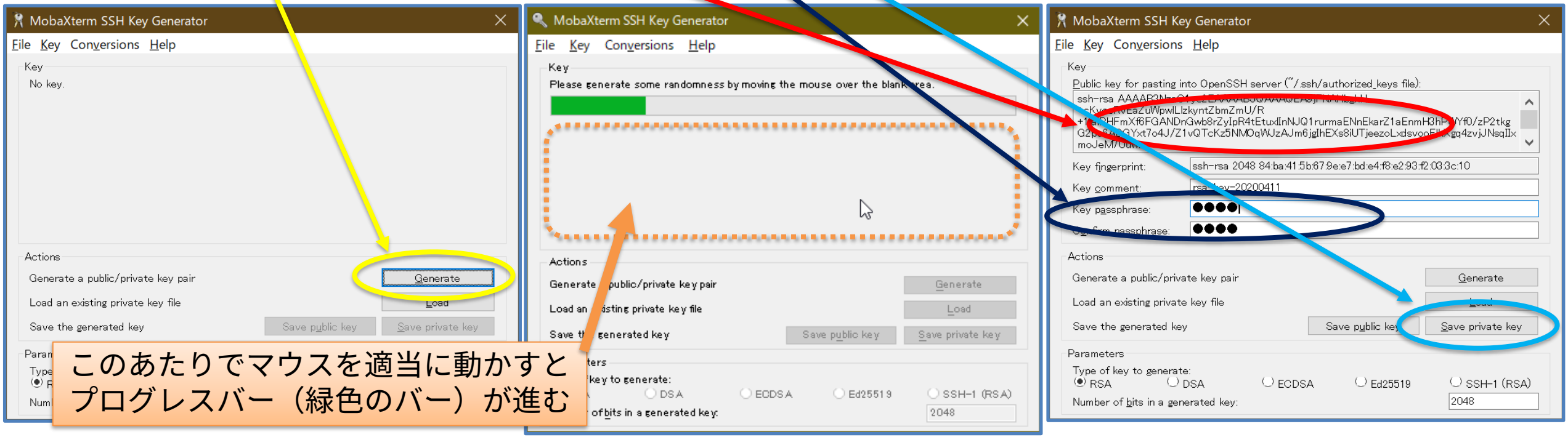




このあたりでマウスを適当に動かすとプログレスバー（緑色のバー）が進む

鍵ファイルの生成方法（ボタンなどの配置の確認用）

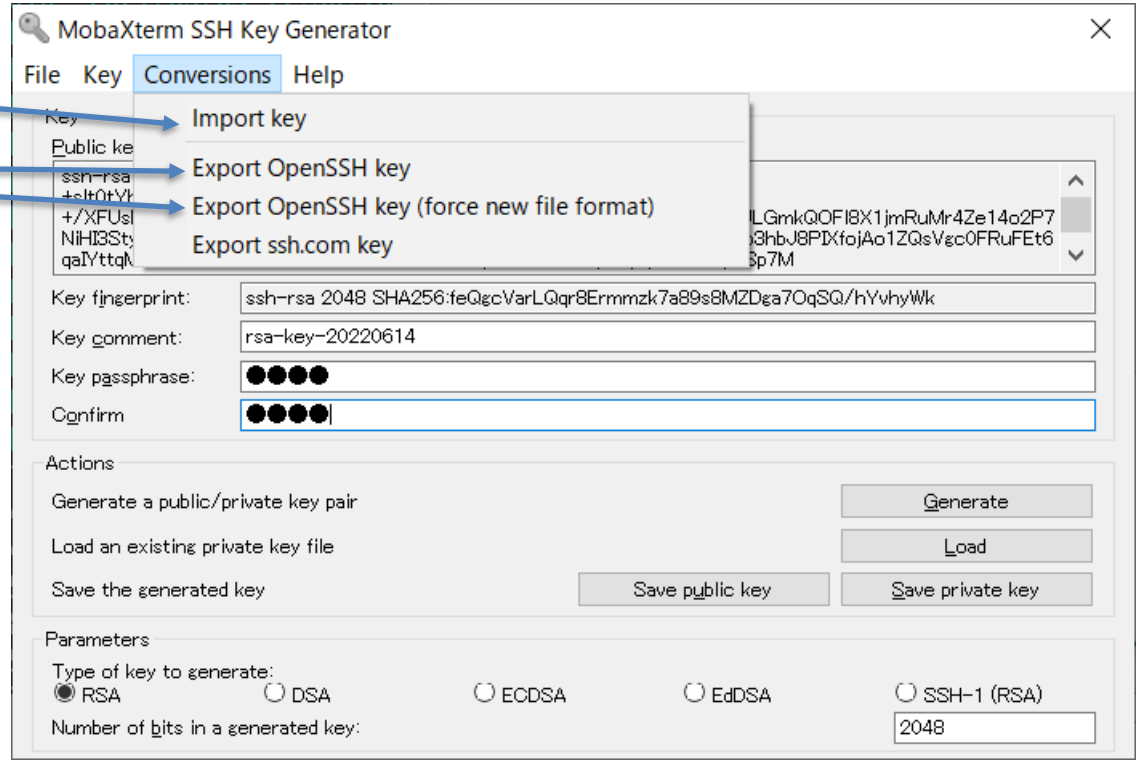
- 「generate」ボタンを押す　→　マウスを動かす
- 「passphrase（パスフレーズ）」（鍵を使うためのパスワード、自分で決める）を入力し「Save private key（秘密鍵を保存）」ボタンで拡張子ppkのファイルを保存する
- 「Public key（公開鍵）」をどこかに保存しておく
  - これは接続先ホストに書き込む情報（「不老」ではHPCポータルから登録する、実際には~/.ssh/authorized_keysファイルに書き込まれることになる）



鍵ファイルの生成方法（補足、鍵形式の変換など）

- 「Conversions」メニューから作成済の鍵を読み込んだりOpenSSH形式の鍵※の保存ができる
  - ※一般的にid_rsaという拡張子なしのファイル名で扱われるもの。sshコマンドでSSH接続を行う（後述）する際に使える。スパコン上（Linux上）でsshコマンドを使うときの秘密鍵としても使える。

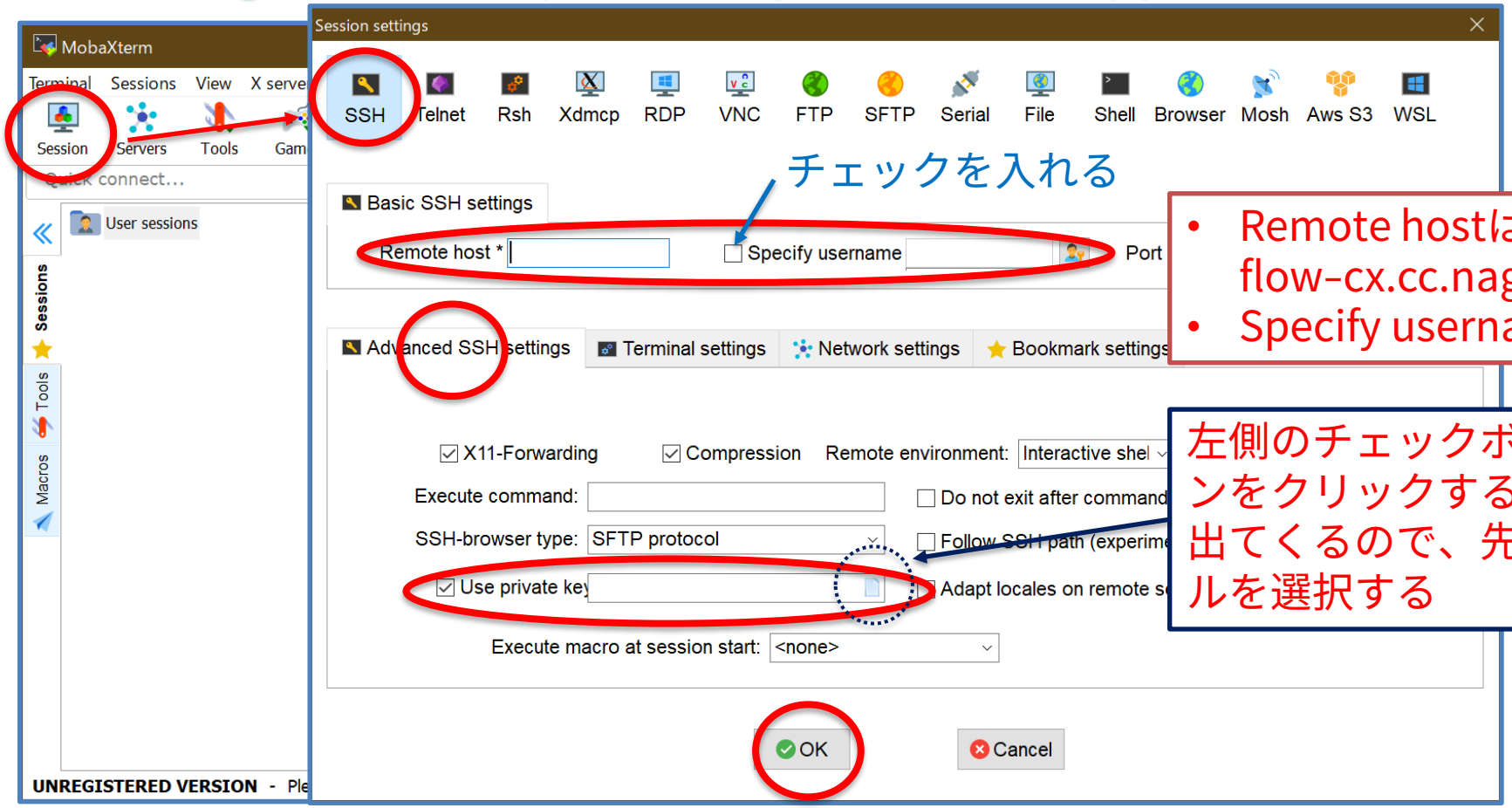
  - MobaXtermで作成した鍵ファイルの読み込み
  - OpenSSH形式での保存
    （どちらを選んでも良い。
    とても古い機器に繋がないのであれば下が良い？）



- 本資料の終盤にある
  「秘密鍵ファイルの形式変換について」
  のページも参考にしてください

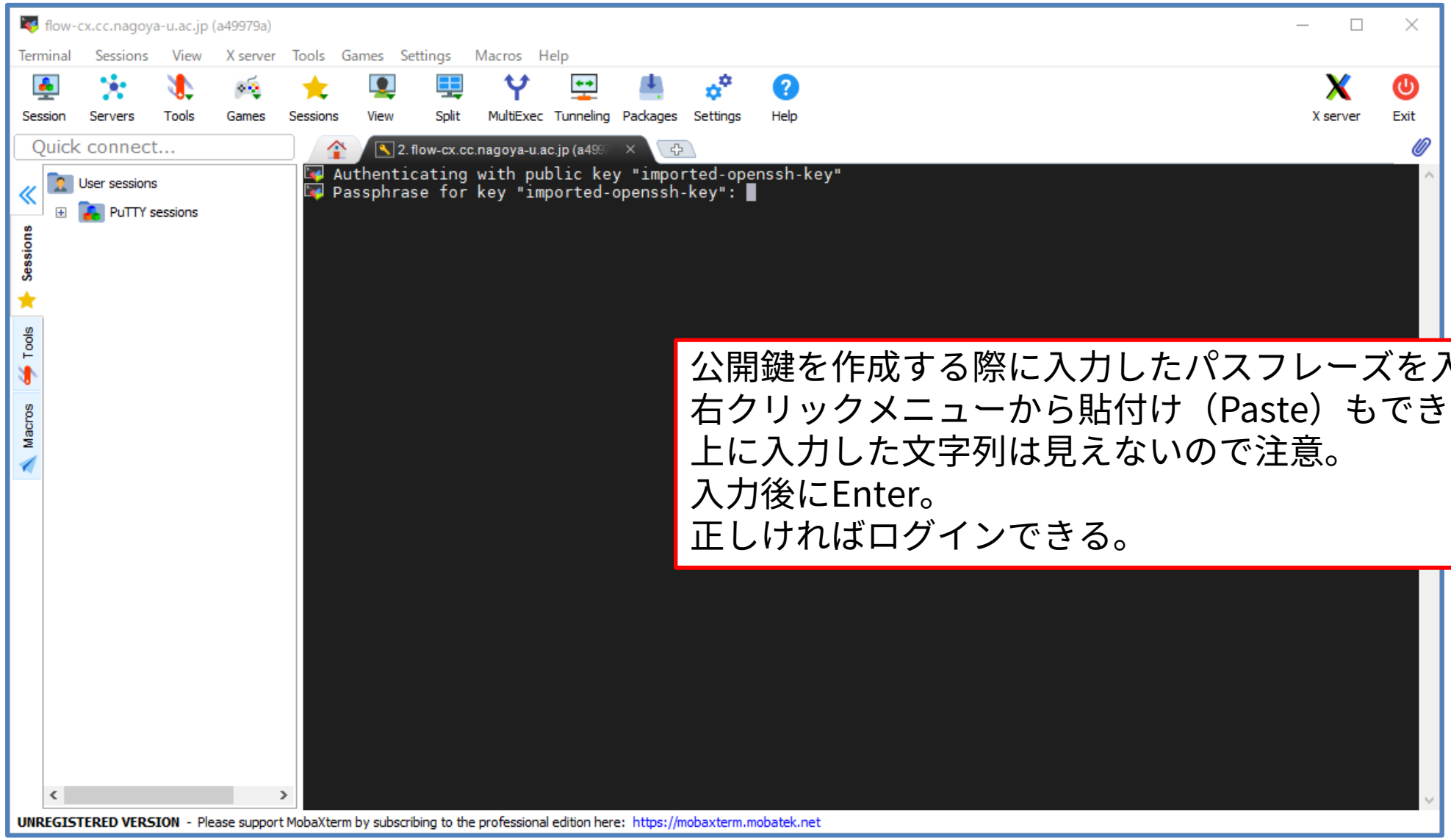
MobaXtermを使ったSSH接続の手順：公開鍵認証を使う場合

- 「Session」ボタンを押すとSession settings画面が開く。
- 「SSH」を選択し、「Remote host」と「Specify username」を入力。さらに「Advanced Settings」の「Use private key」で保存したppkファイルを選択し、「OK」。



- Remote hostはflow-fx.cc.nagoya-u.ac.jpやflow-cx.cc.nagoya-u.ac.jpなど
- Specify usernameはユーザ名（ログインID）

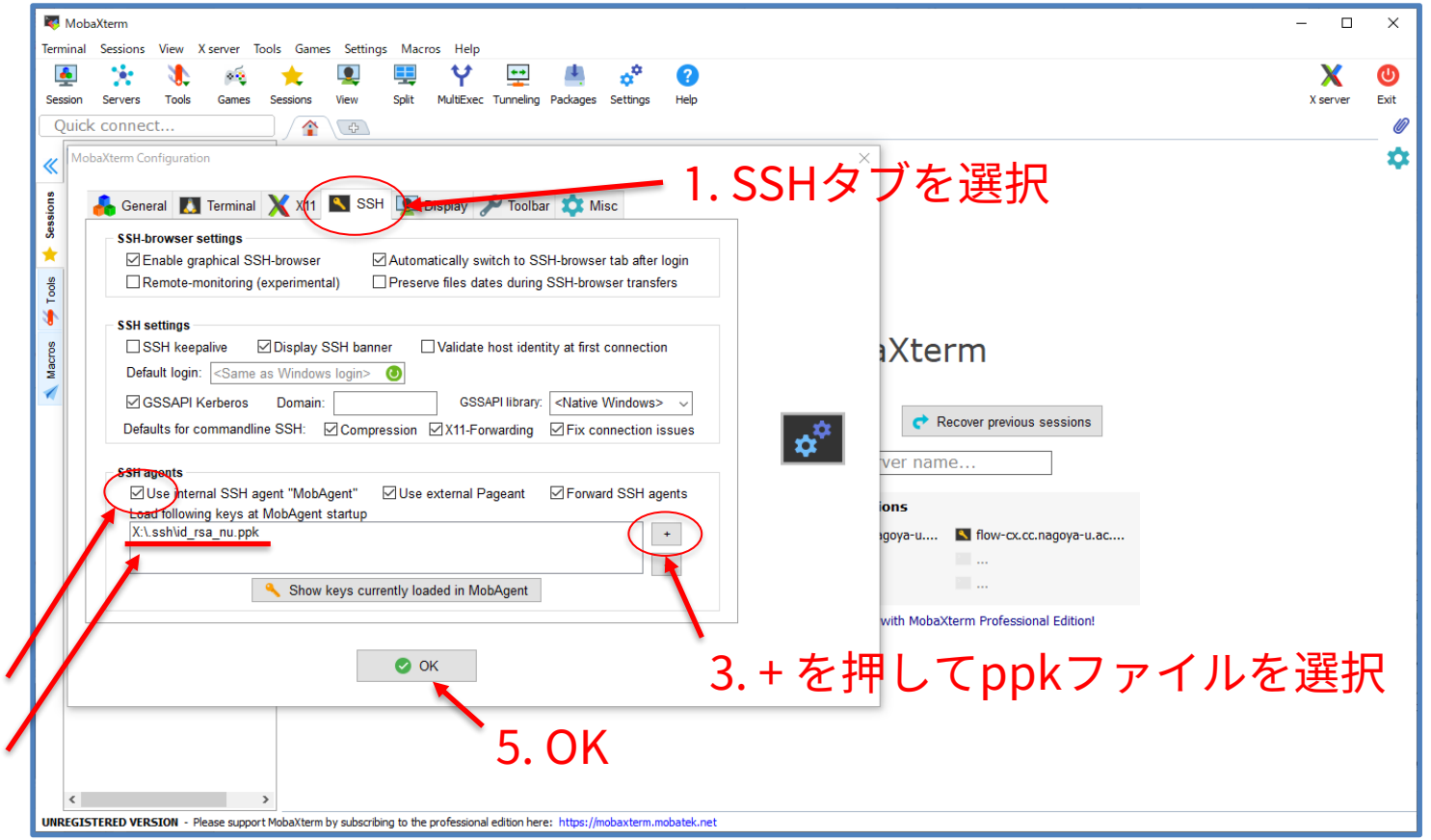
左側のチェックボックスをONにして右端のアイコンをクリックするとファイルを開くウィンドウが出てくるので、先ほど作成したid_rsa.ppkファイルを選択する



公開鍵を作成する際に入力したパスフレーズを入力する。
右クリックメニューから貼付け（Paste）もできる。画面上に入力した文字列は見えないので注意。
入力後にEnter。
正しければログインできる。

パスフレーズの扱いについての補足

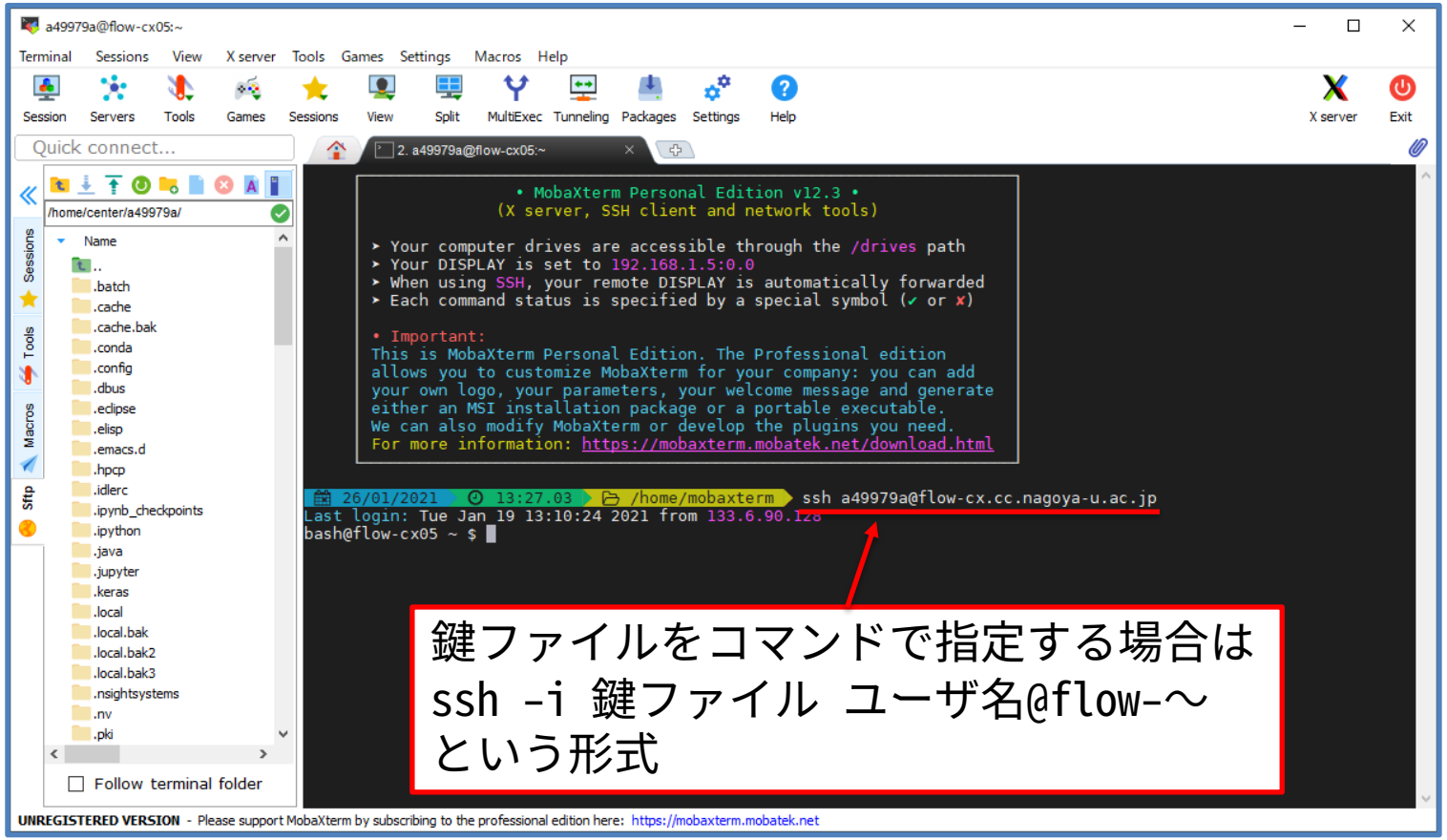
- 前述の使い方ではSSH接続の度にパスフレーズの入力が必要
- あらかじめ登録しておくことで接続の度に入力する手間を省ける
- メニューの「Settings」→「Configuration」で開く設定画面の「SSH」タブでppkファイルを登録しておくことができる
- 右図の通り登録する
- MobaXtermが再起動し、起動時にパスフレーズの入力を求められる
- 正しいパスフレーズを入力すればその後はsshの度にパスフレーズを入力しなくても良くなる
- 再起動の度に入力を求められるため不要になったら登録した情報を削除すると良い



MobaXtermを使ったSSH接続の手順：local terminalを使う

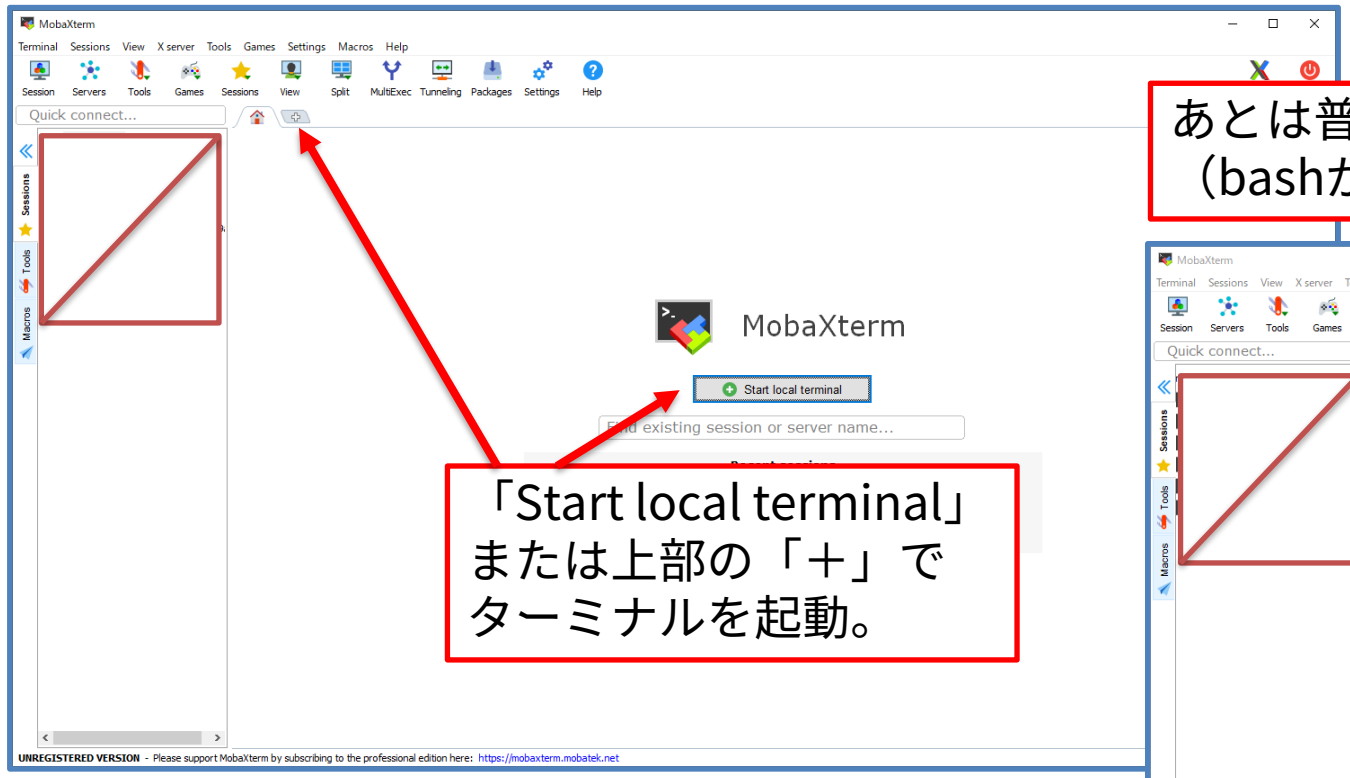
- 「local terminal」からsshコマンドで接続することも可能
- ここで公開鍵認証を使うには、-iオプションで鍵ファイルを指定するか、前ページで説明したように「Settings」の「Configuration」の「SSH」の「SSH agents」で鍵を登録しておく必要がある

- （より具体的な接続例は数ページあとで紹介する）



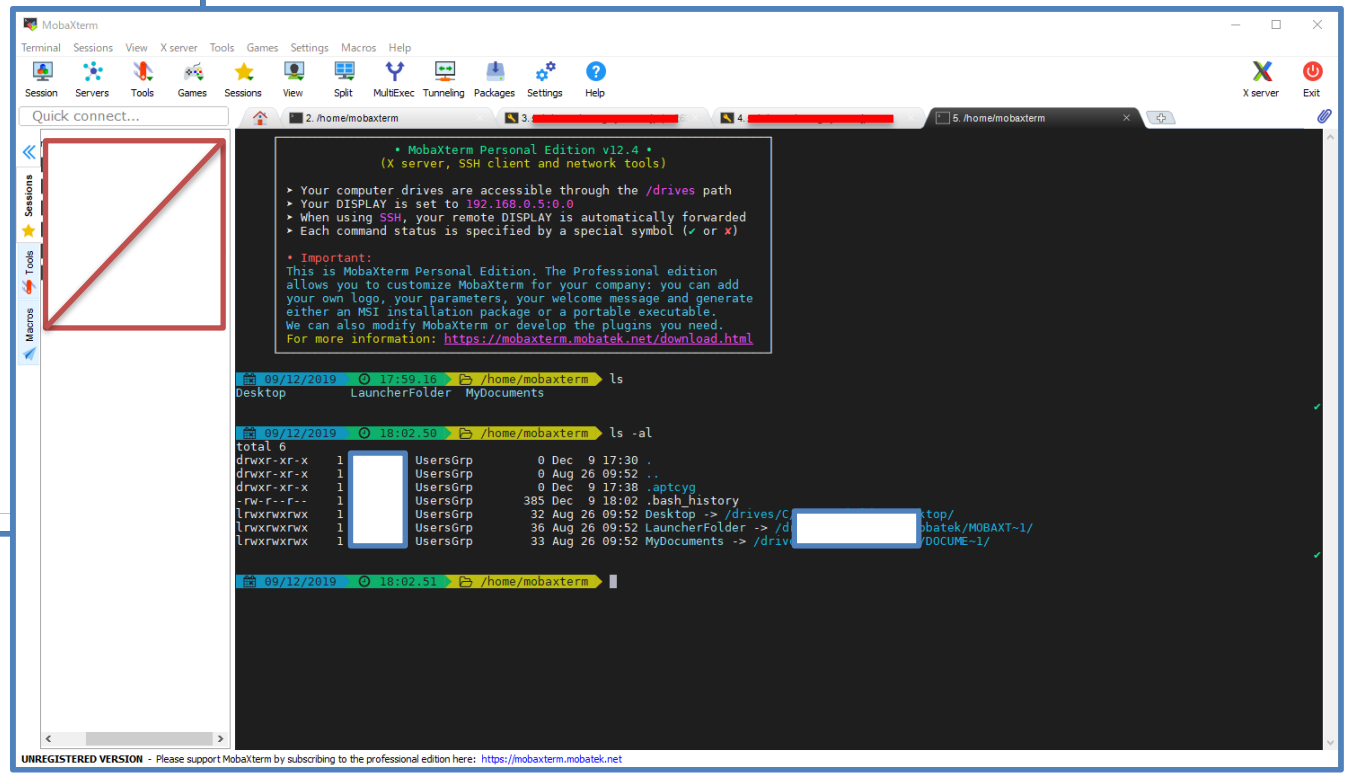
MobaXtermをローカル開発環境として使う

- MobaXtermはssh接続せずに手元で利用することもできる（PuTTYではできない）



「Start local terminal」
または上部の「＋」で
ターミナルを起動。

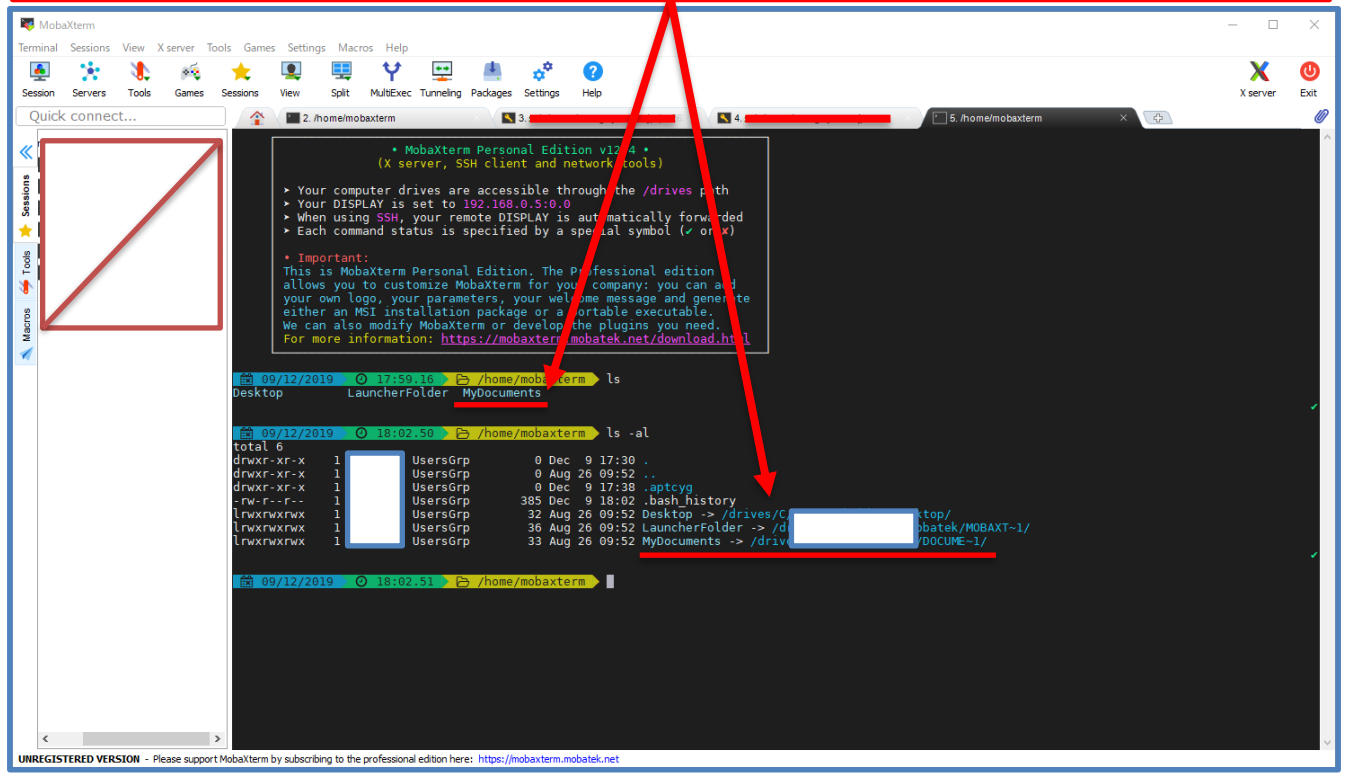
あとは普通のLinuxのCUIのように利用することができる
（bashが起動する）



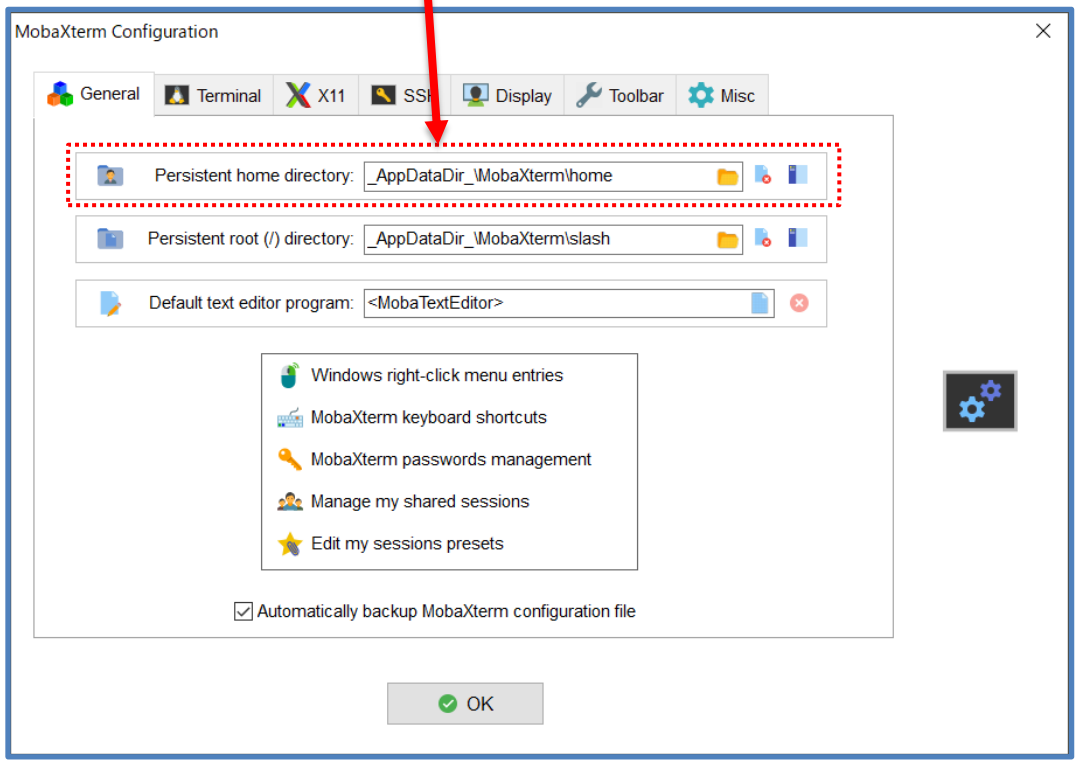
local terminalからのファイル操作

- Windowsで普段使っているファイルを参照することもできる

- lsコマンドを実行するといくつかのリンクがみえる（Windowsの「ドキュメント」が「MyDocuments」として見えているなど）
- Cドライブが /drives/C で参照できることもわかる

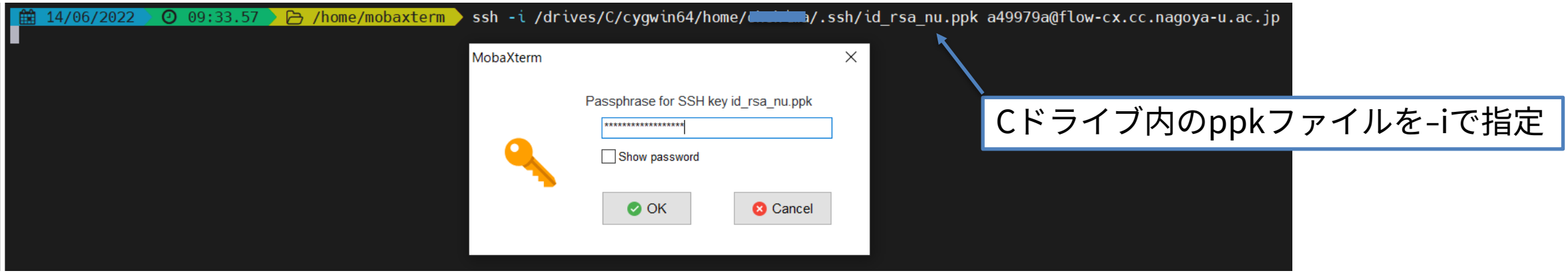


local terminal起動時のホームディレクトリは設定から変更することもできる。

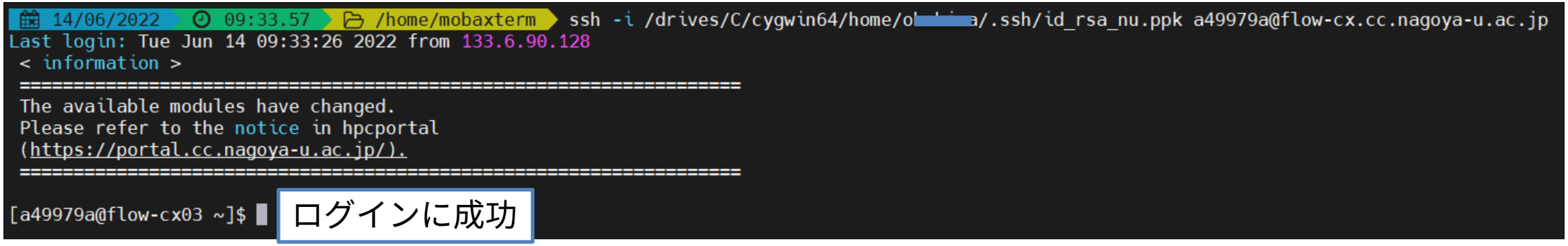


local terminalからのSSH接続（1/2）

- local terminalからppk形式の公開鍵を指定して「不老」に接続する例

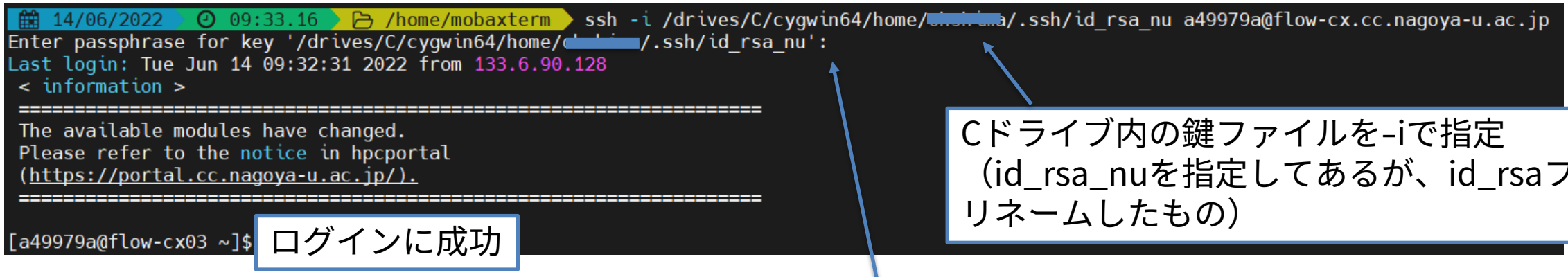


パスフレーズ入力画面が出てくる。
正しいパスフレーズを入力すると接続できる。



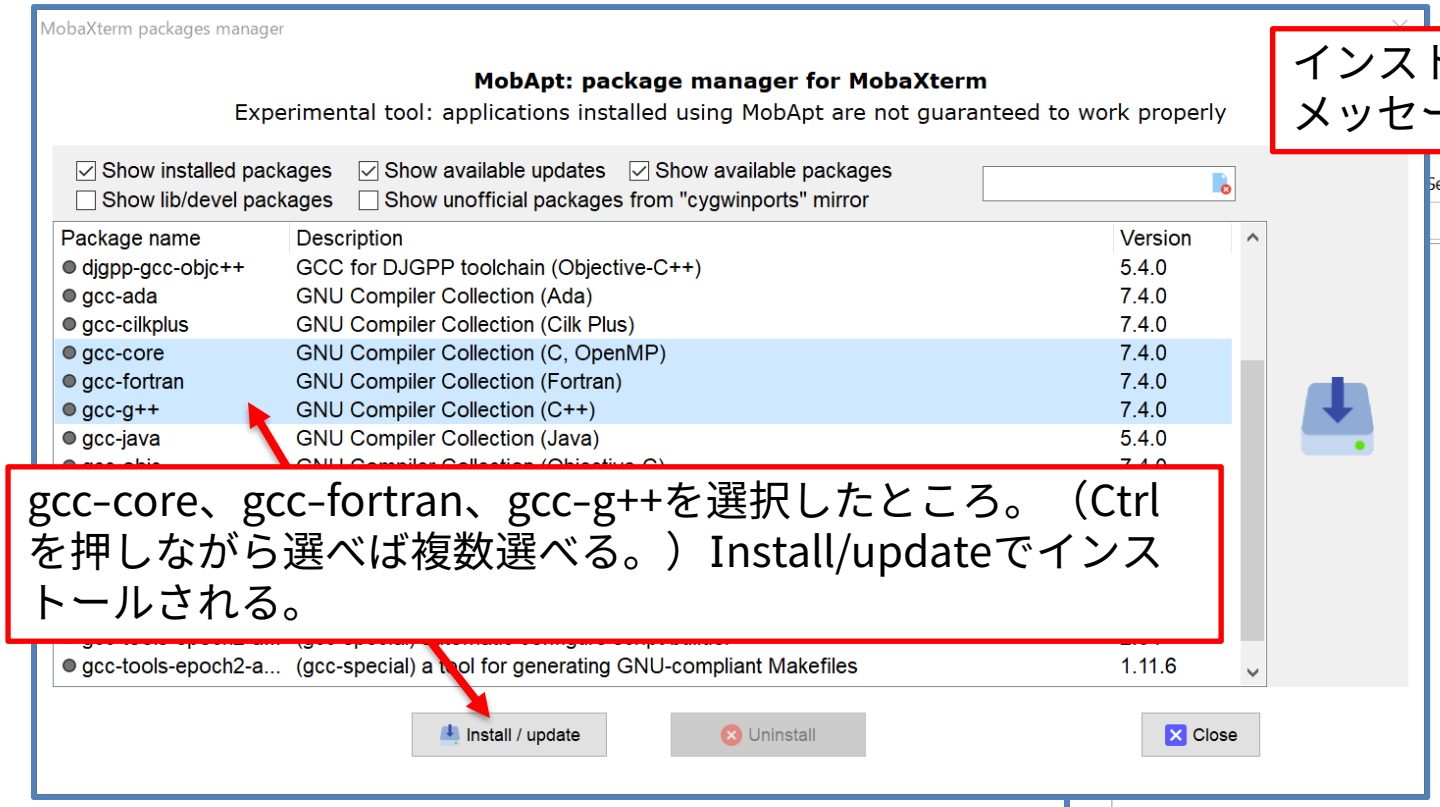
local terminalからのSSH接続（2/2）

- id_rsa形式（OpenSSH形式）の公開鍵を指定して「不老」に接続する例

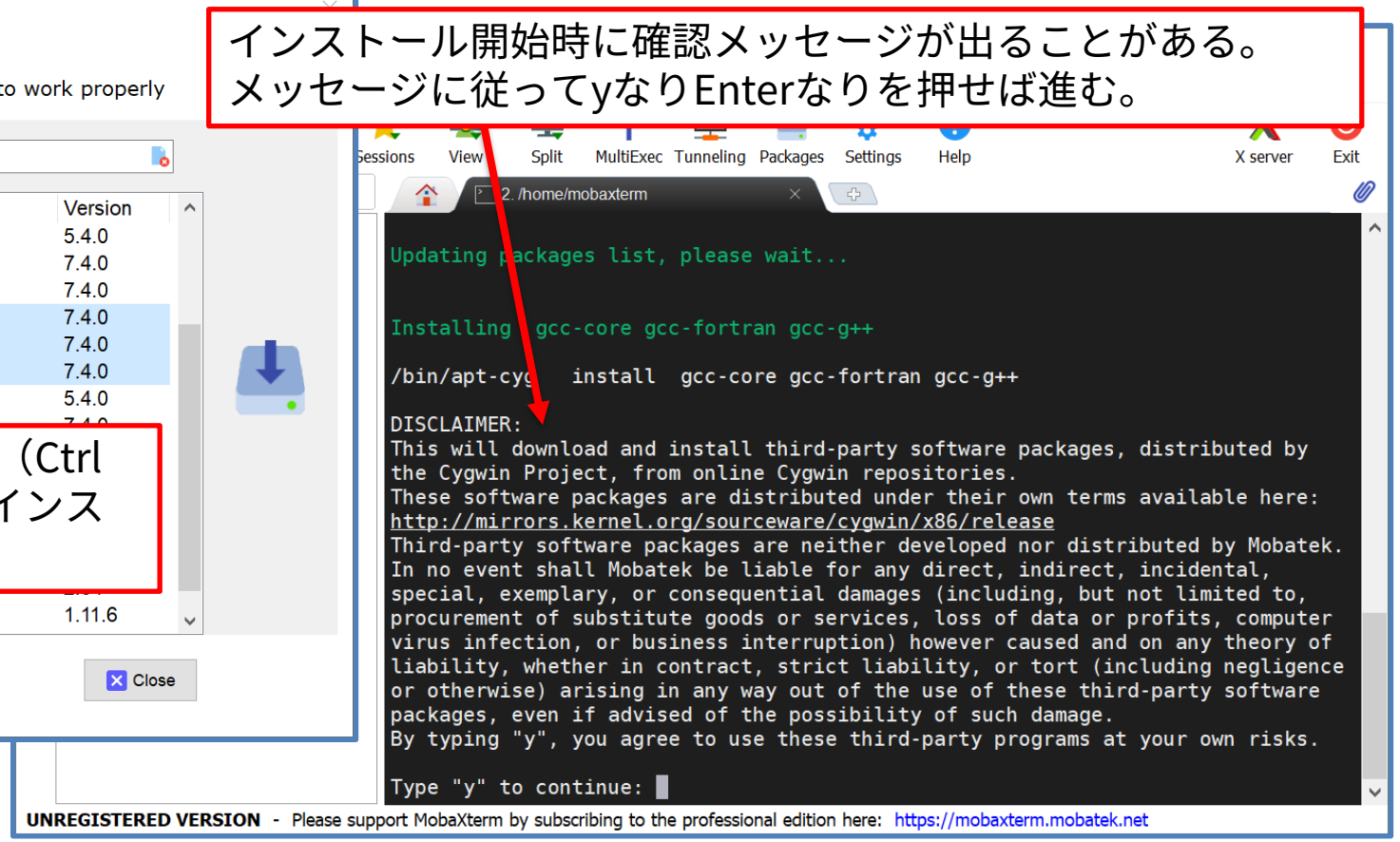


MobaXtermをローカル開発環境として使う：コンパイラの導入

- MobAptコマンドを実行（Toolsメニューの中のMobApt～からでも起動できる）してgcc等をインストールすれば、プログラムの開発環境としても利用可能。

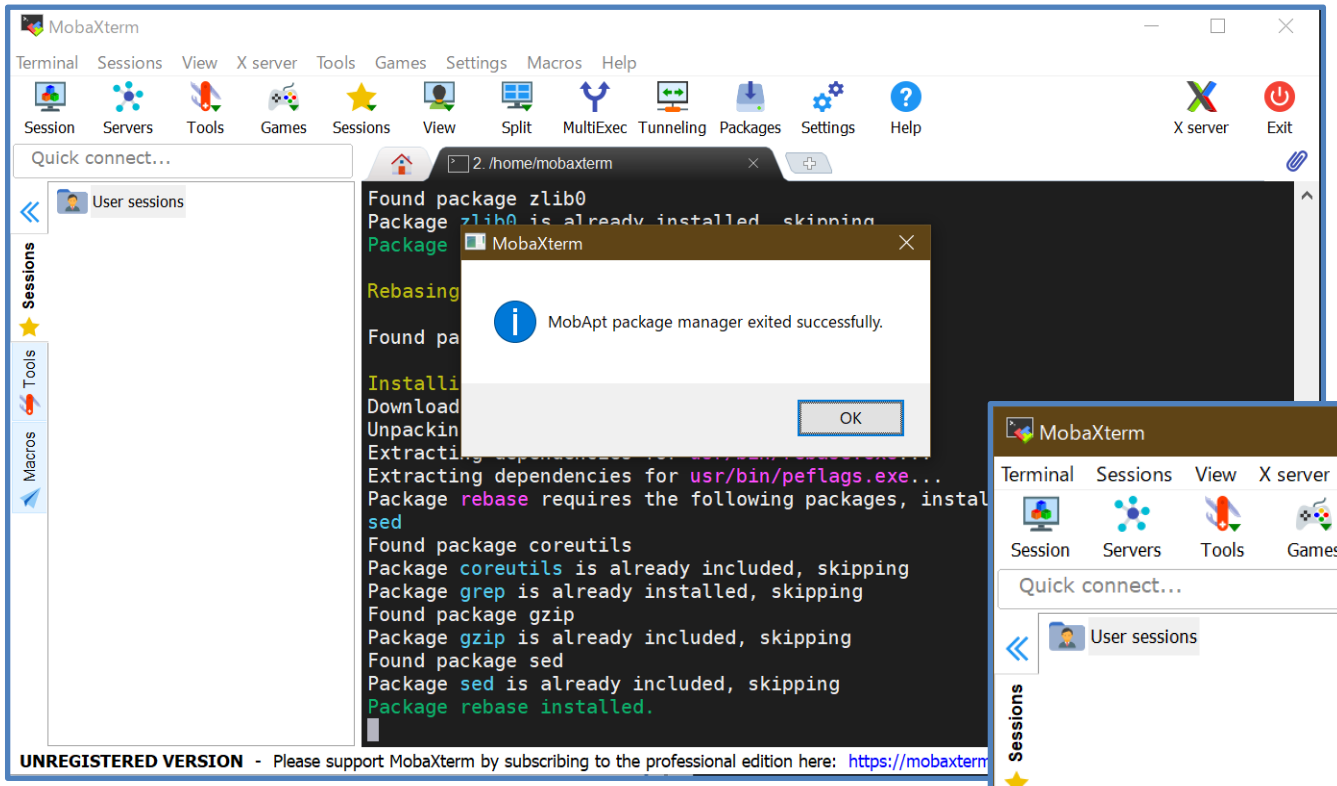


gcc-core、gcc-fortran、gcc-g++を選択したところ。（Ctrlを押しながら選べば複数選べる。）Install/updateでインストールされる。

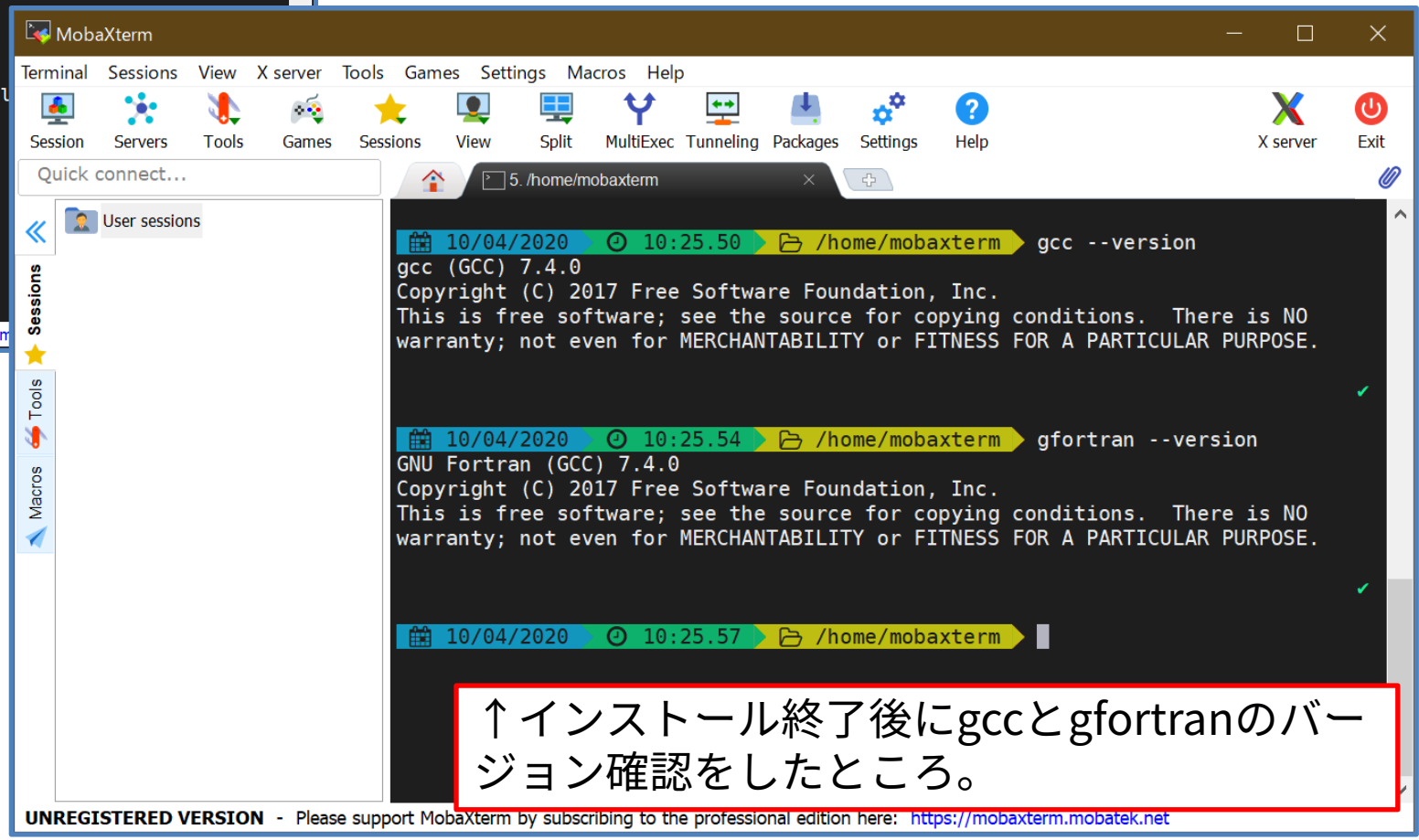


インストール開始時に確認メッセージが出ることがある。
メッセージに従ってyなりEnterなりを押せば進む。

↓インストール終了



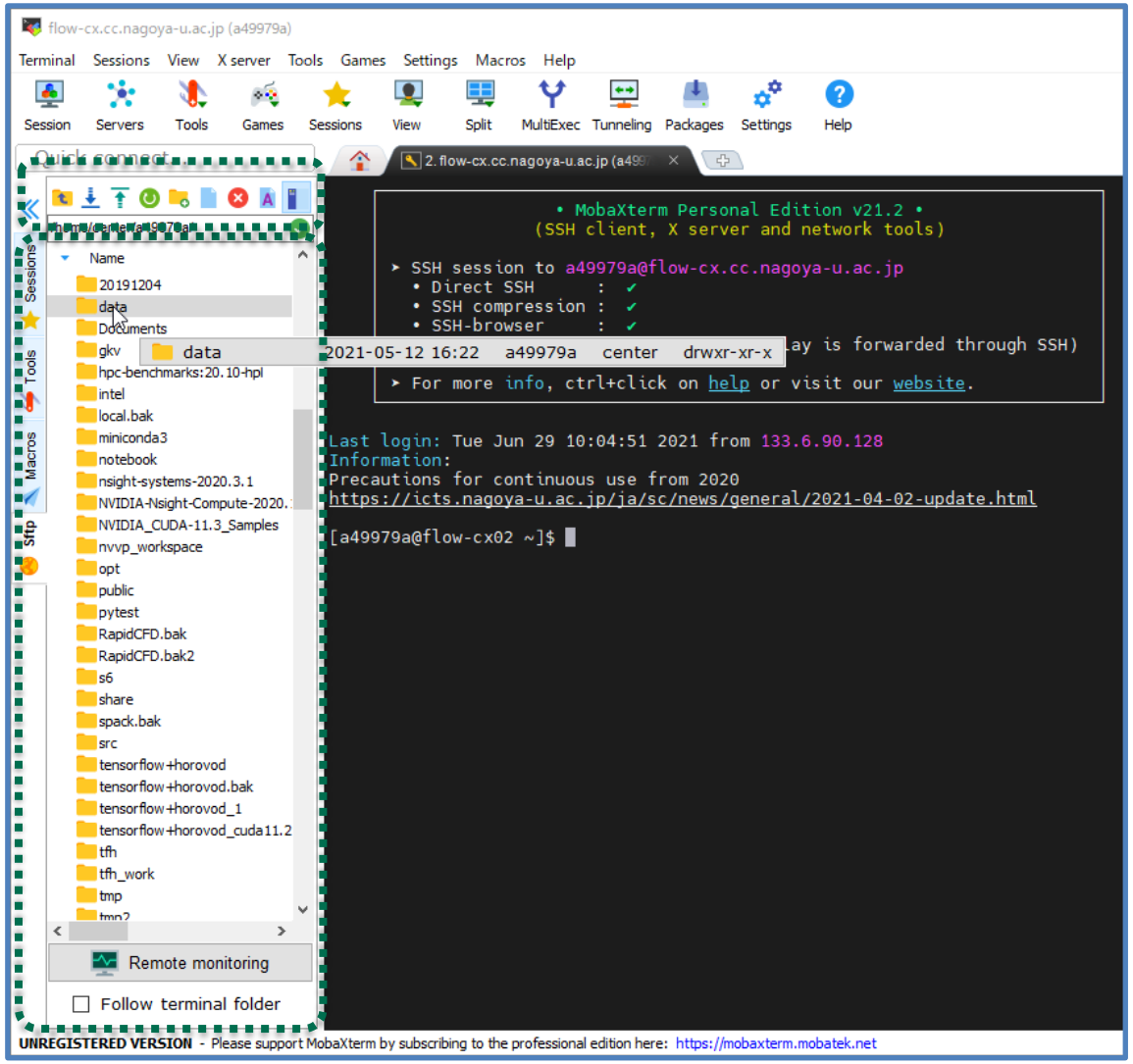
MobaXtermにはテキストエディタ（MobaTextEditor）も用意されており、editコマンドで起動可能。viもインストールされている。emacsなどを使いたい場合はgcc同様にMobAptでインストールすれば良い。
もちろん、Windowsのエディタで編集したものをコンパイルしても良い。



↑インストール終了後にgccとgfortranのバージョン確認をしたところ。

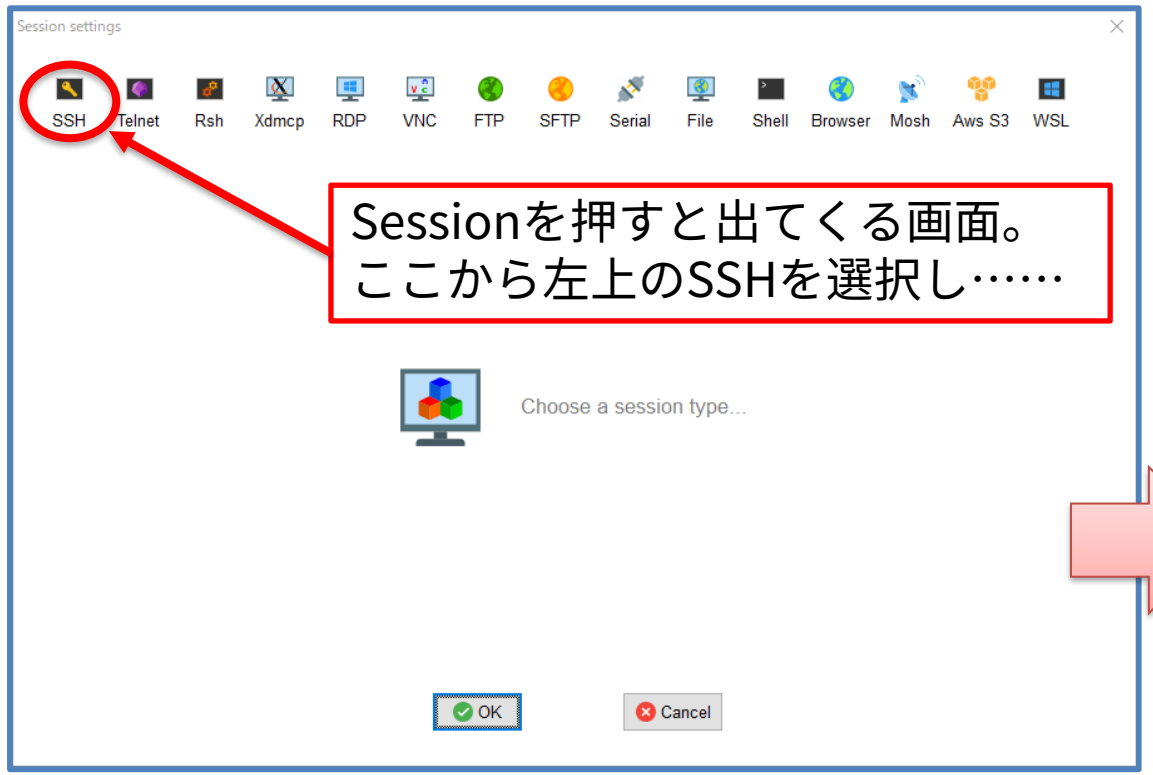
MobaXtermによるファイル操作

- 左側に接続先のディレクトリ・ファイル一覧が表示される
- 右クリックメニューや上部のアイコンからアップロードやダウンロードが可能



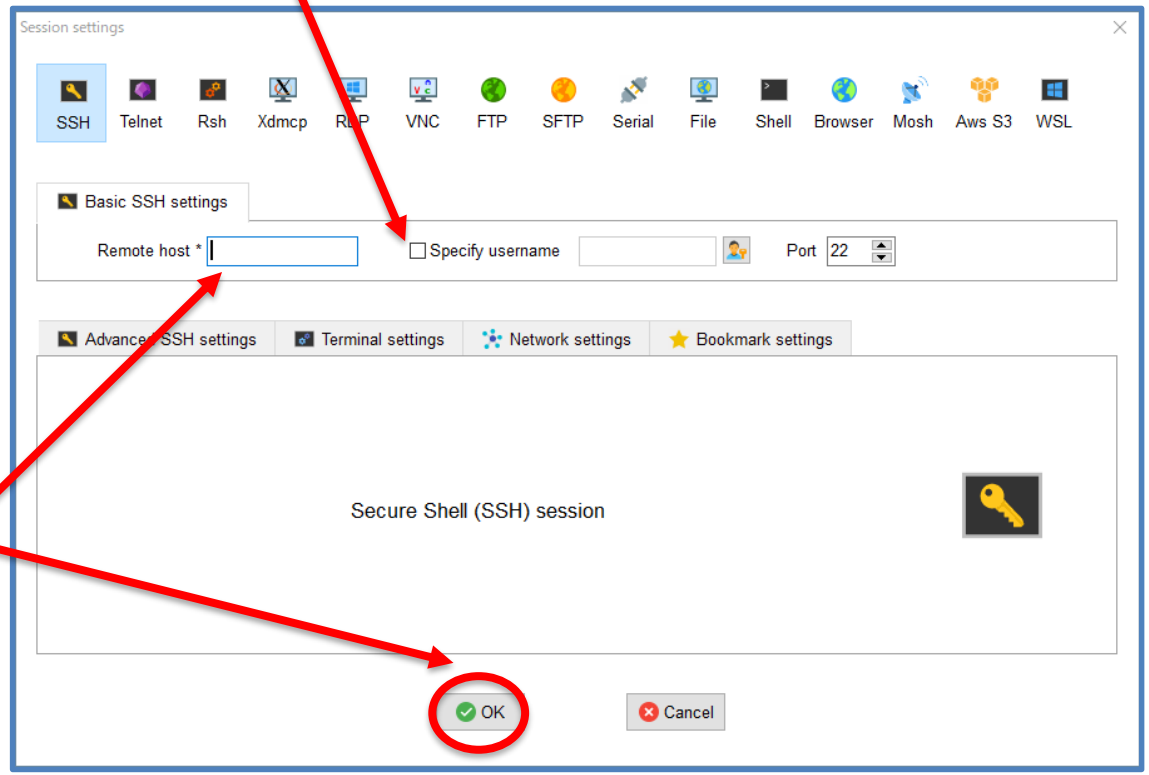
※「不老」への接続は公開鍵認証方式でのみ可能です

MobaXtermを使ったSSH接続の手順：パスワード認証を使う場合



Sessionを押すと出てくる画面。
ここから左上のSSHを選択し……

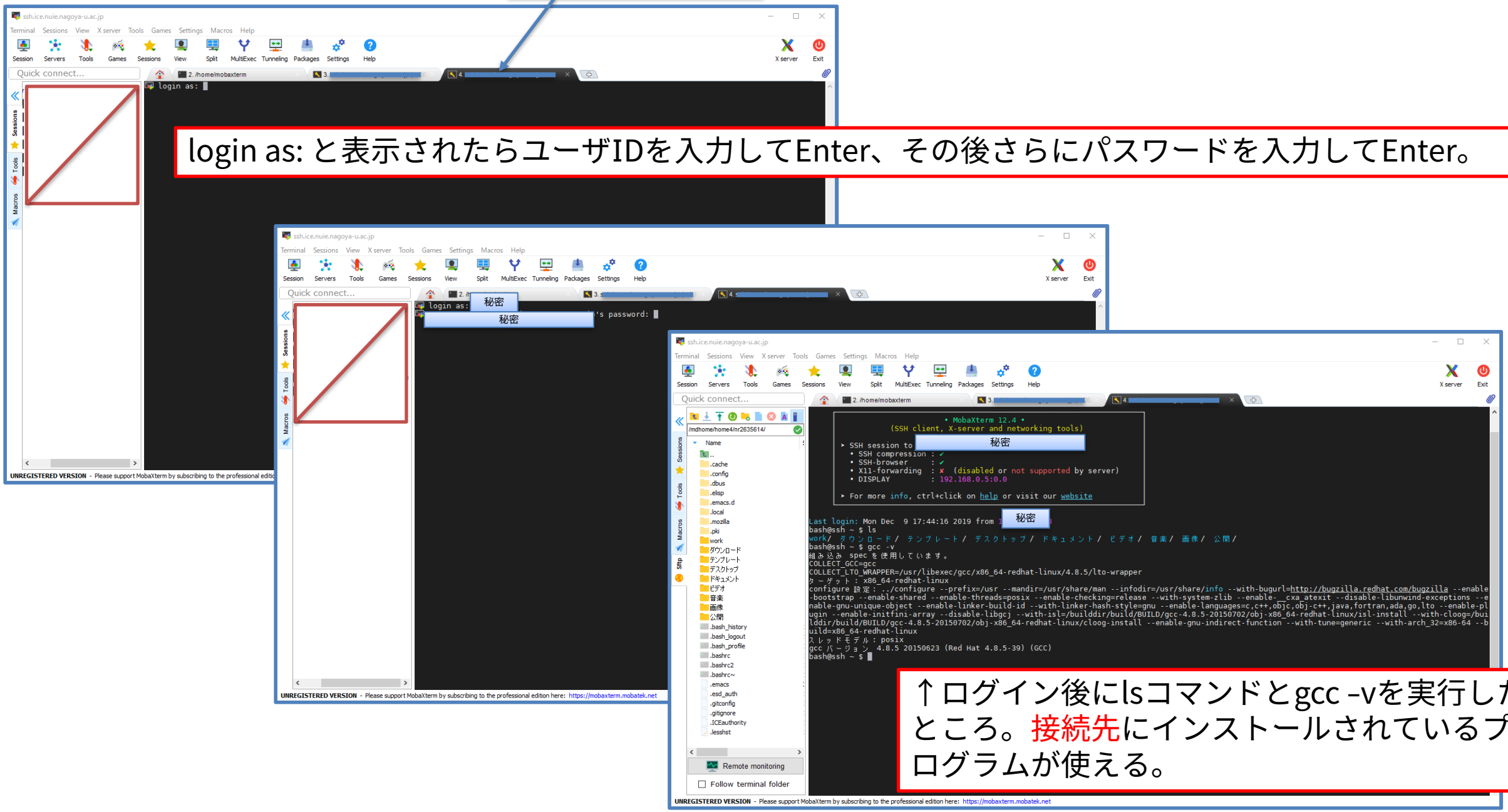
Specify usernameのチェックボックスをチェックしてテキストボックスにユーザIDを入力しておくと、次の画面でのID入力が省略される。



Remote hostに接続先ホスト名を入力して「OK」

タブには接続先の情報が表示される

login as: と表示されたらユーザIDを入力してEnter、その後さらにパスワードを入力してEnter。



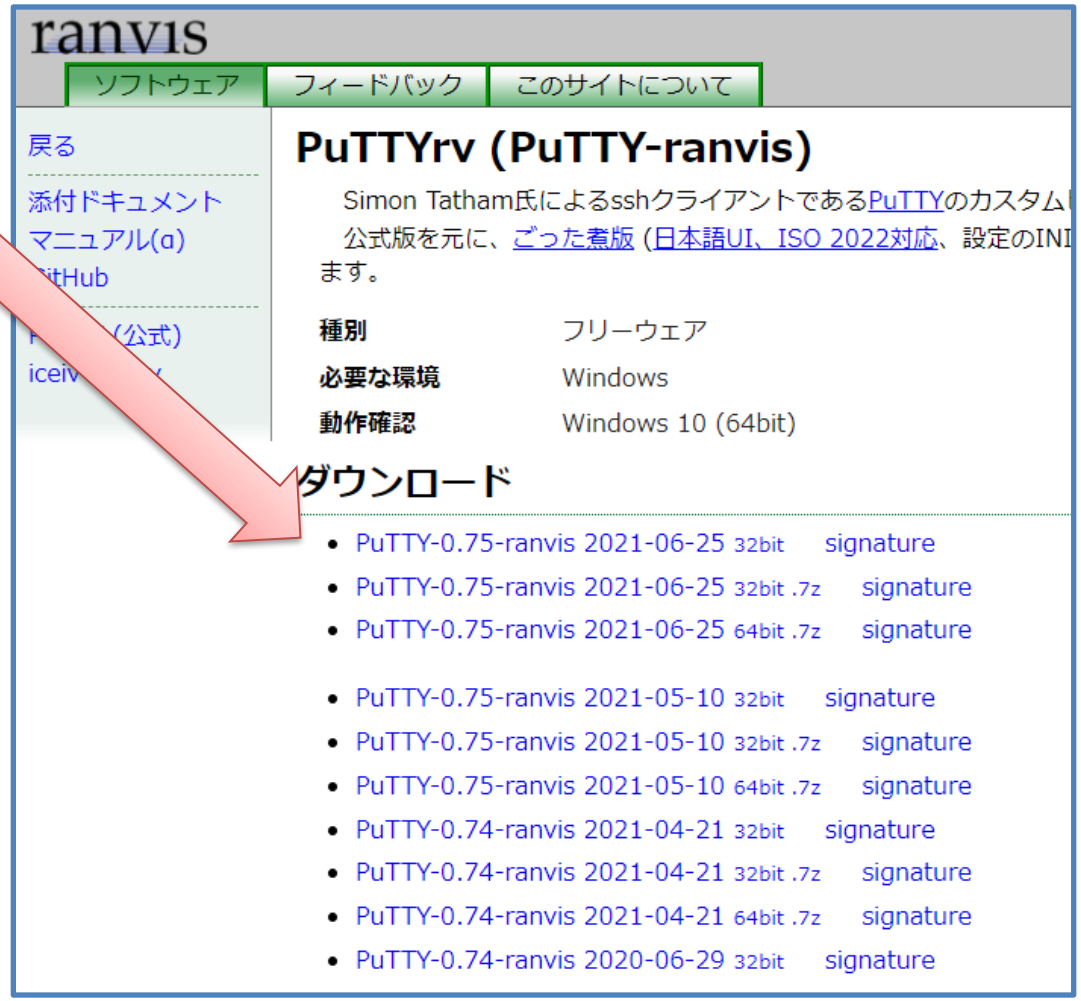
↑ログイン後にlsコマンドとgcc -vを実行したところ。接続先にインストールされているプログラムが使える。

PuTTYの使い方

PuTTYを使ってSSH接続する

- https://www.ranvis.com/putty　から最新版をダウンロードする
  - 本資料を作成した時点では　PuTTY-0.75-ranvis 2021-06-25 32bit

一番上のものを使う。

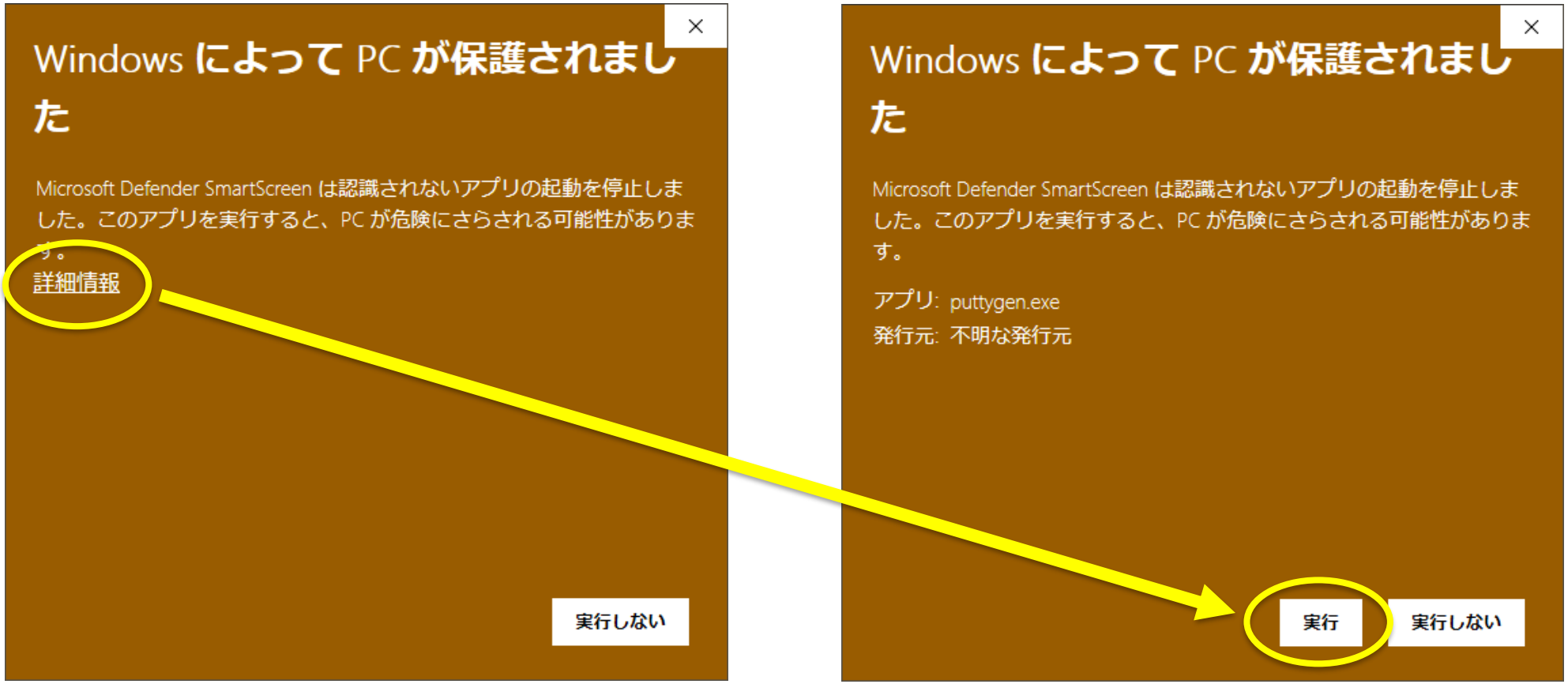


  - PuTTY-0.75-ranvis-20210625.win32.zip というzip圧縮ファイルが入手できる
  - zipファイルを展開する
    - ダウンロードしたファイルを右クリック →「すべて展開」
    - 使うのはその中にある putty.exeというファイル

PuTTYを使ってSSH接続する：SSH公開鍵認証を使う場合

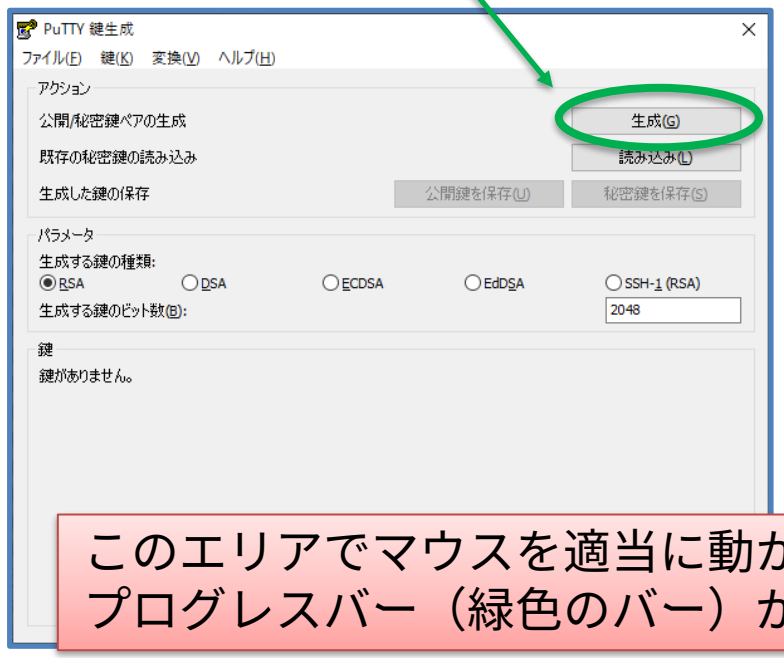
- まず鍵ファイルを作成する
  - puttygen.exeで公開鍵を準備

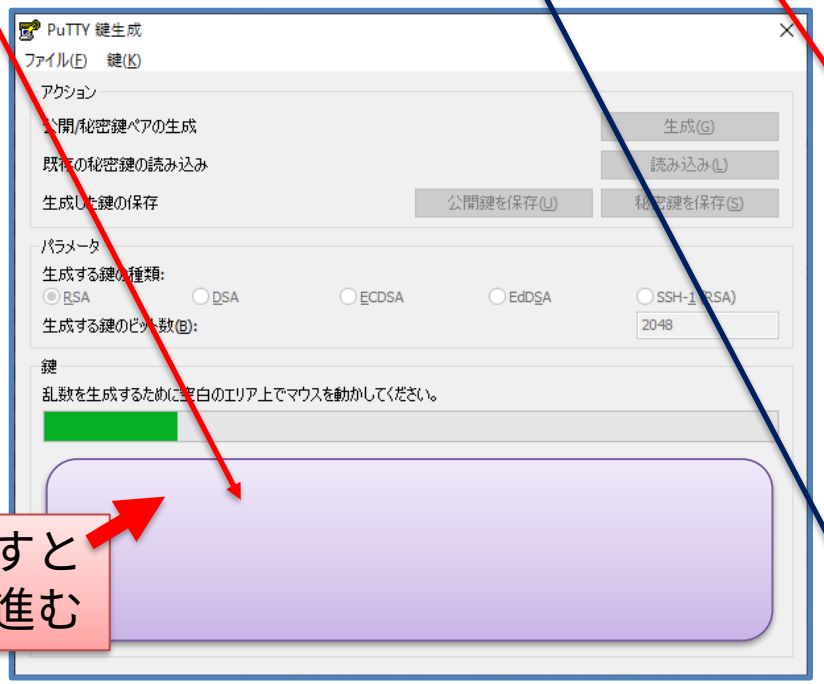
※起動時に警告が出る可能性がある（putty.exeでも同様）



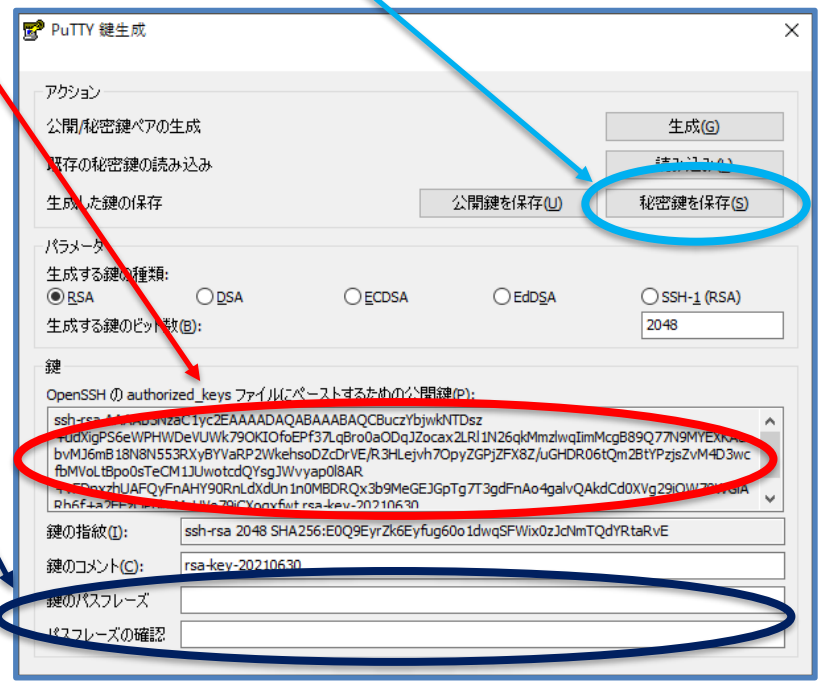
PuTTYを使ってSSH接続する：公開鍵認証を使う場合

- 鍵ファイルの作成
  - puttygen.exeを起動
  - 「生成」ボタン→マウスを動かす
  - 「パスフレーズ」（鍵を使うためのパスワード）を入力して「秘密鍵を保存」ボタン（拡張子ppkのファイルを保存）
  - 「OpenSSHのauthorized_keysファイルにペーストするための公開鍵」をどこかに保存しておく



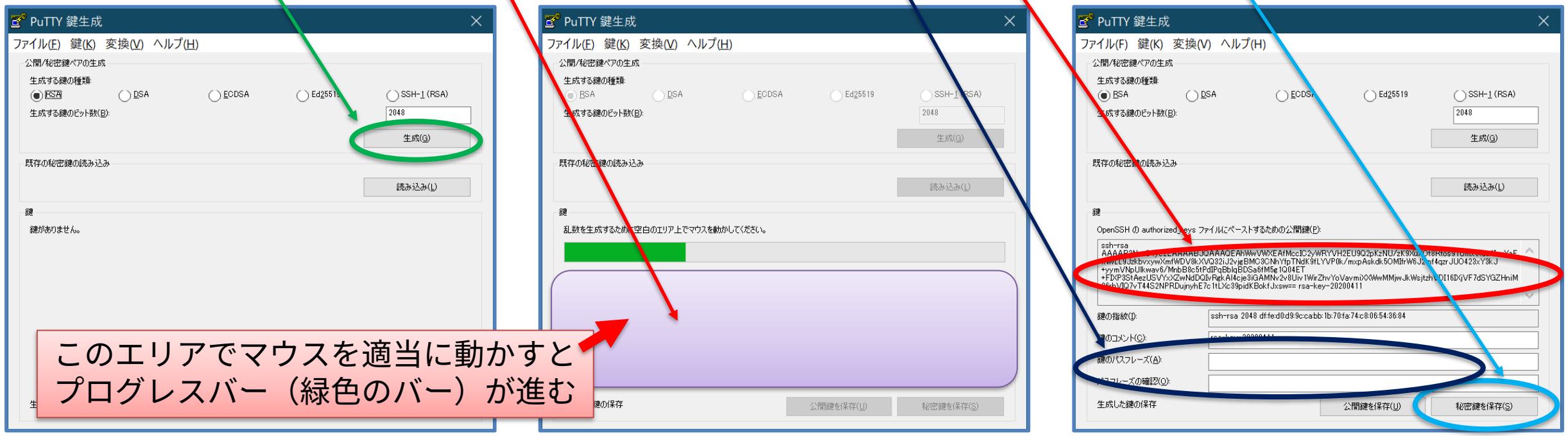


このエリアでマウスを適当に動かすと
プログレスバー（緑色のバー）が進む



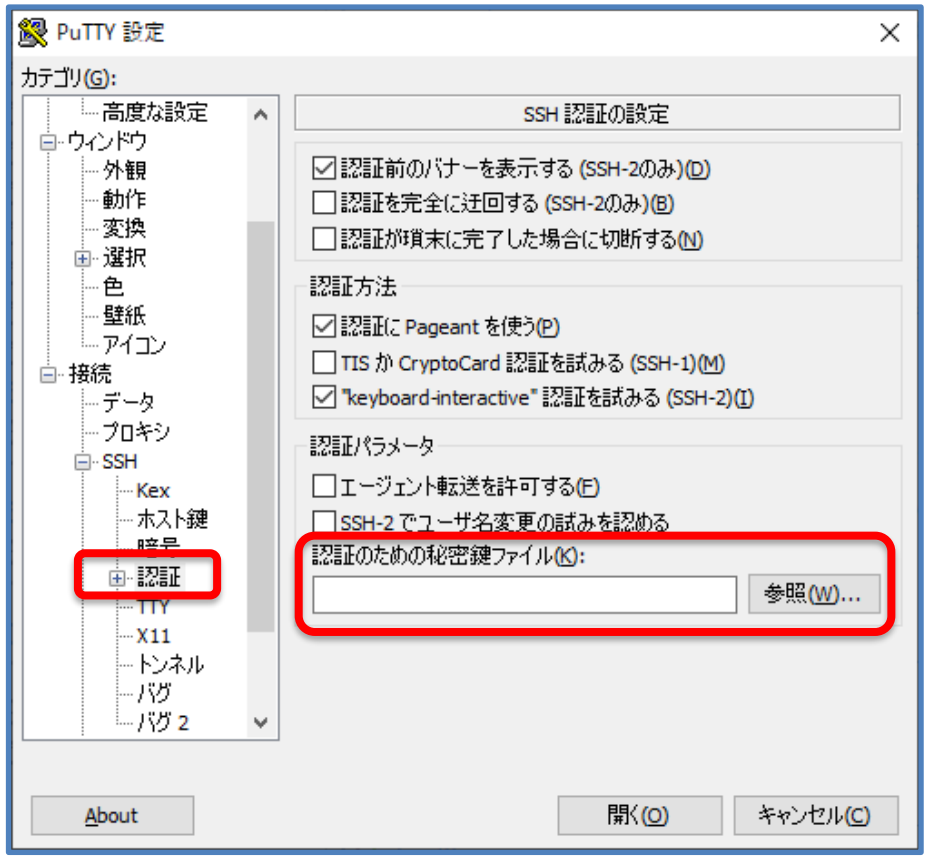
※PuTTYのバージョンが古い場合は画面の構成が違うので注意

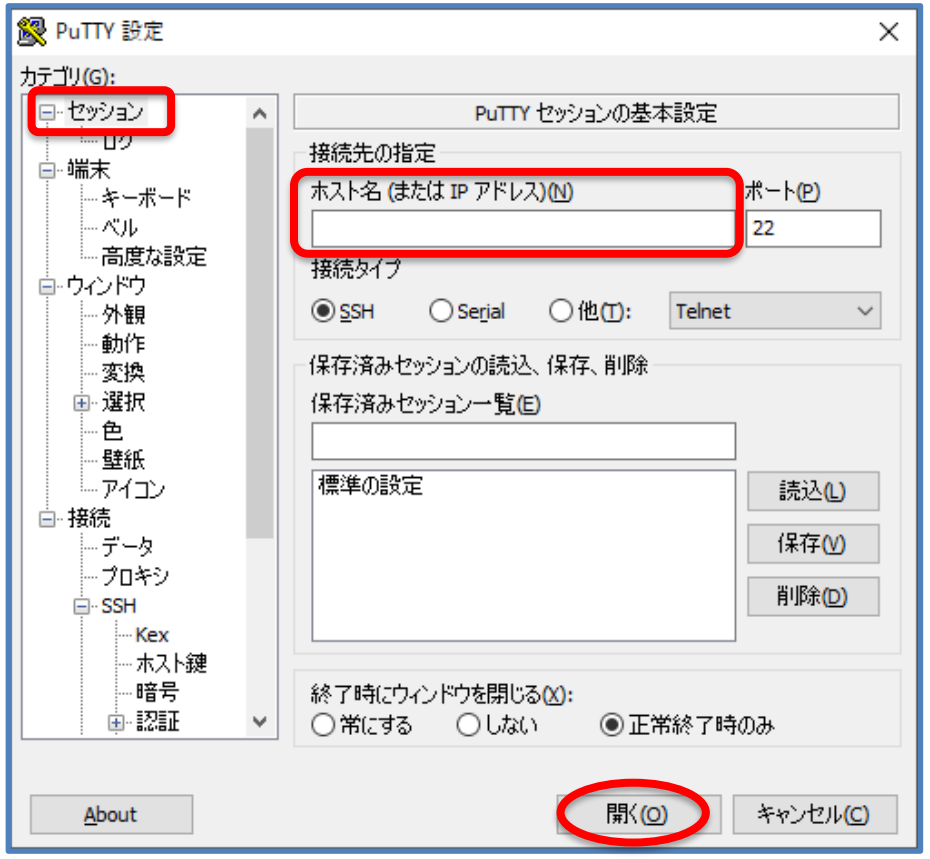
- 鍵ファイルの作成
  - puttygen.exeを起動
  - 「生成」ボタン→マウスを動かす
  - 「パスフレーズ」（鍵を使うためのパスワード）を入力して「秘密鍵を保存」ボタン（拡張子ppkのファイルを保存）
  - 「OpenSSHのauthorized_keysファイルにペーストするための公開鍵」をどこかに保存しておく



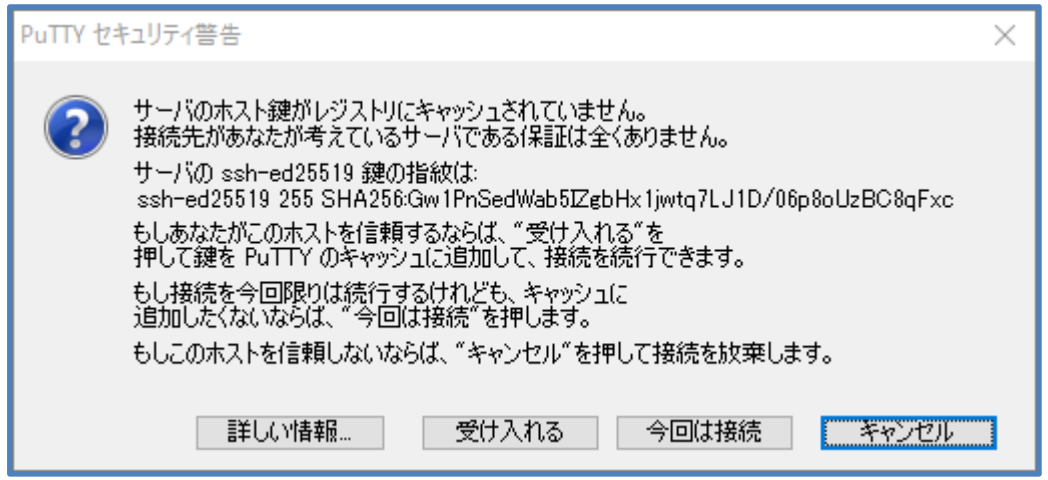
PuTTYを使ってSSH接続する：公開鍵認証を使う場合

- 作成した鍵ファイルを用いて接続する
  - 起動時の画面で「接続」-「SSH」-「認証」の画面を開き、「認証のための秘密鍵ファイル」の「参照」ボタンから保存した秘密鍵（拡張子ppkのファイル）を選択
  - 「セッション」に戻り、ホスト名を入力して「開く」ボタン

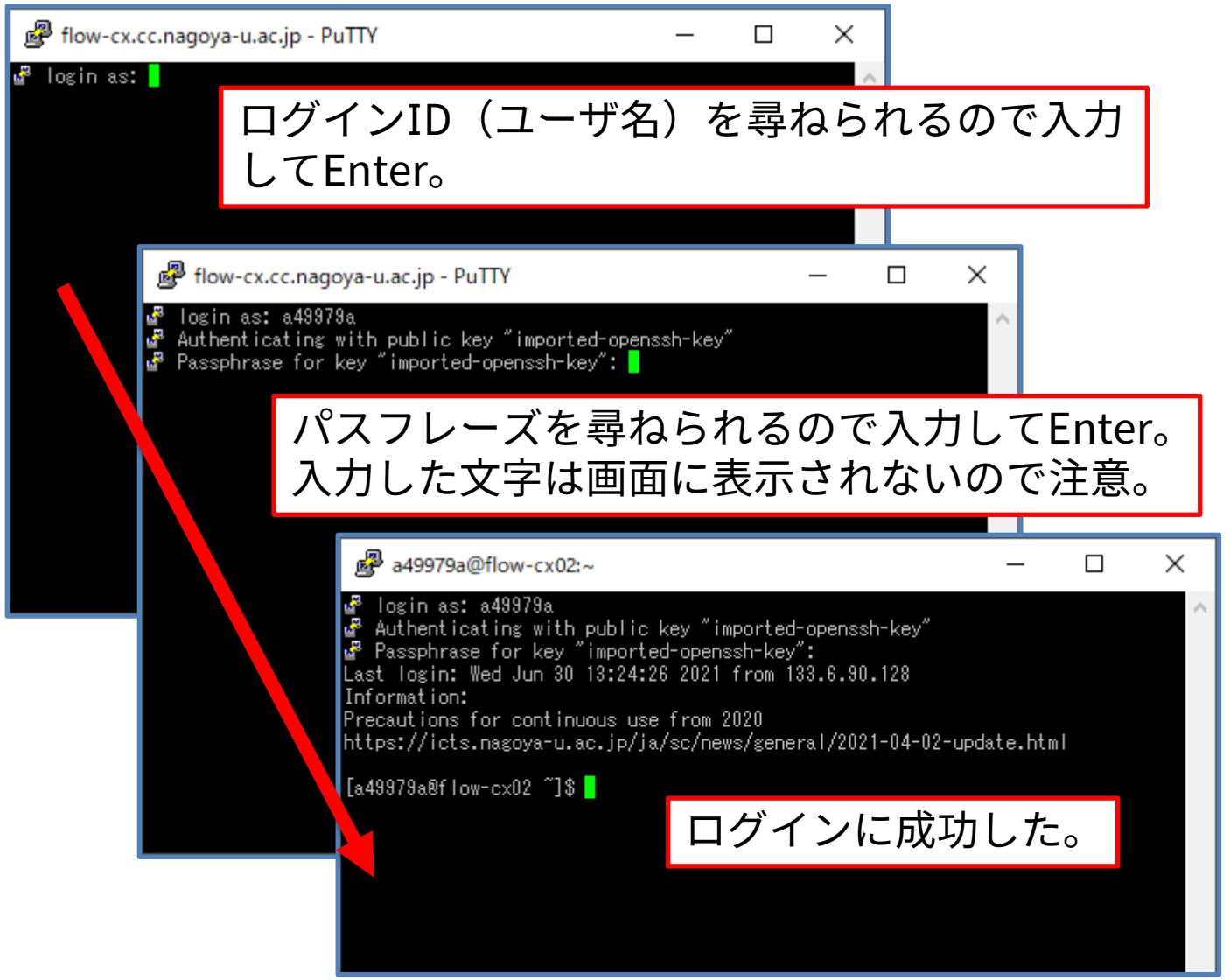




PuTTYを使ってSSH接続する：公開鍵認証を使う場合（つづき）

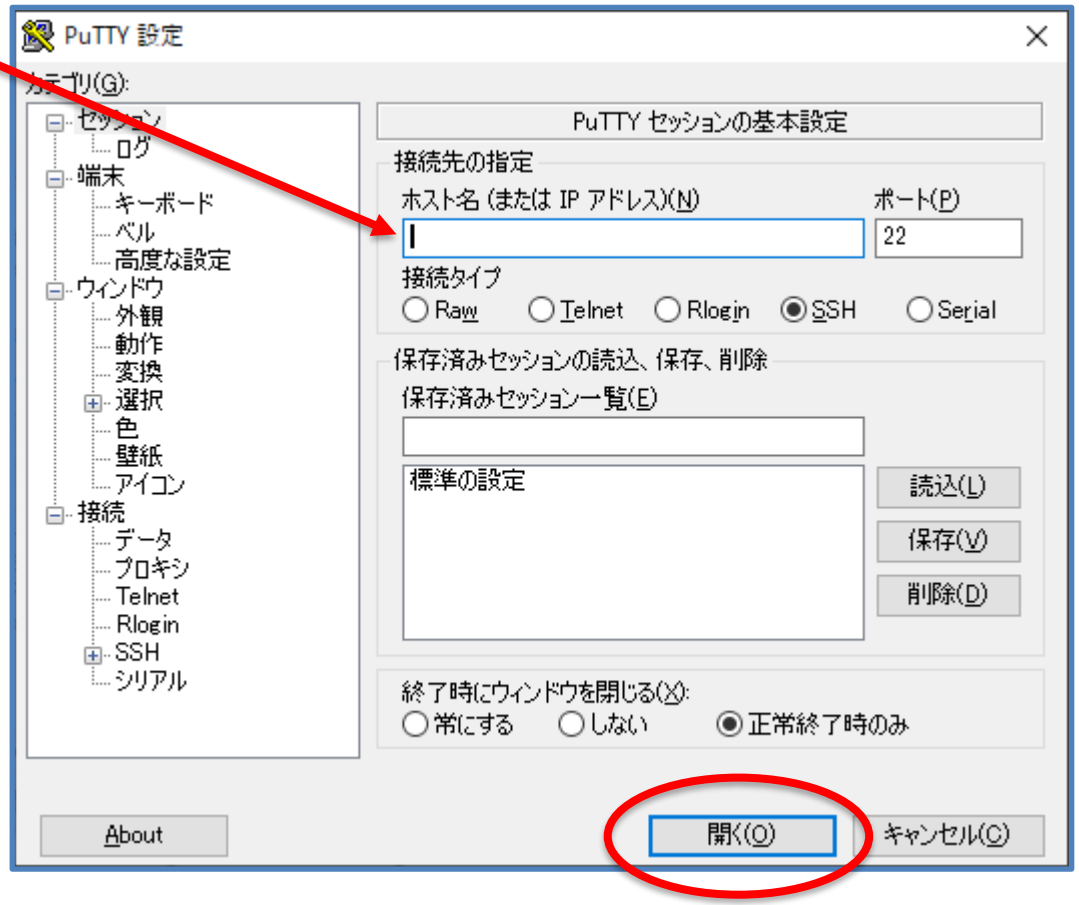


↑初回接続時には接続先が正しいか確認メッセージが表示される。問題なければ「受け入れる」。

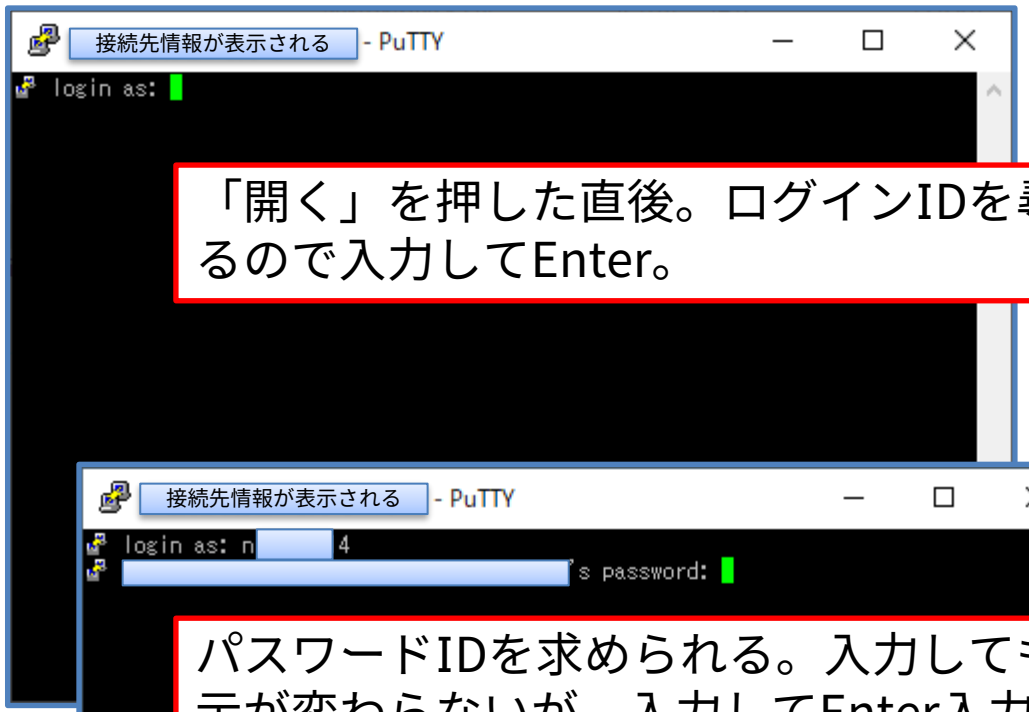


PuTTYを使ってSSH接続する：パスワード認証を使う場合

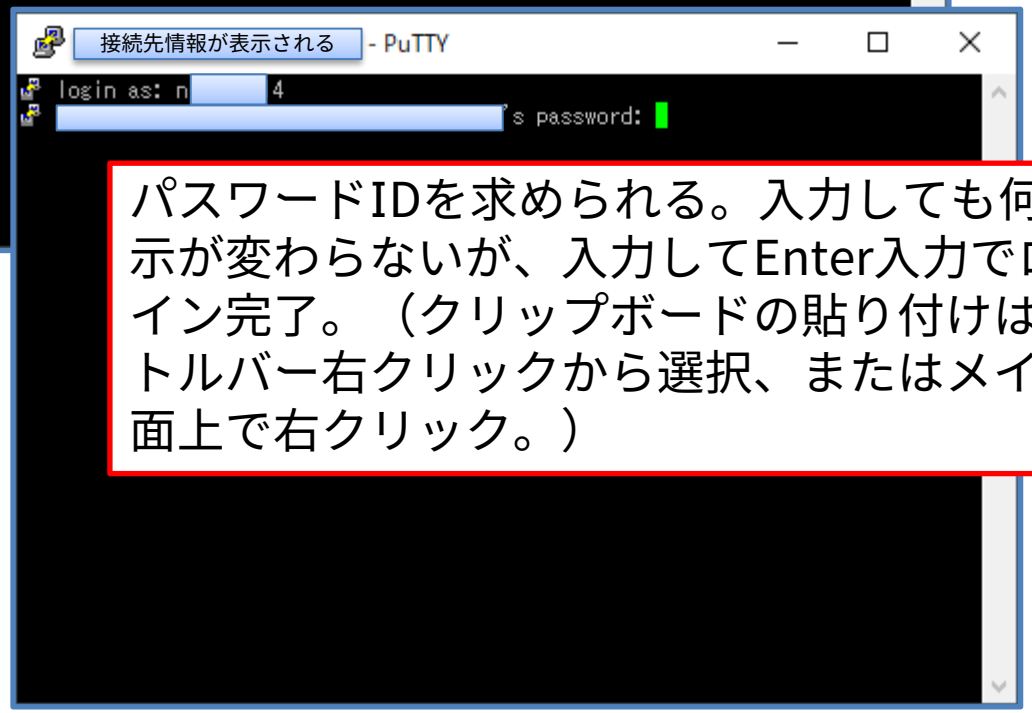
- 接続方法
  - ログインIDとパスワードで接続する場合は、
    ホスト名を入力して「開く」を選ぶだけでよい



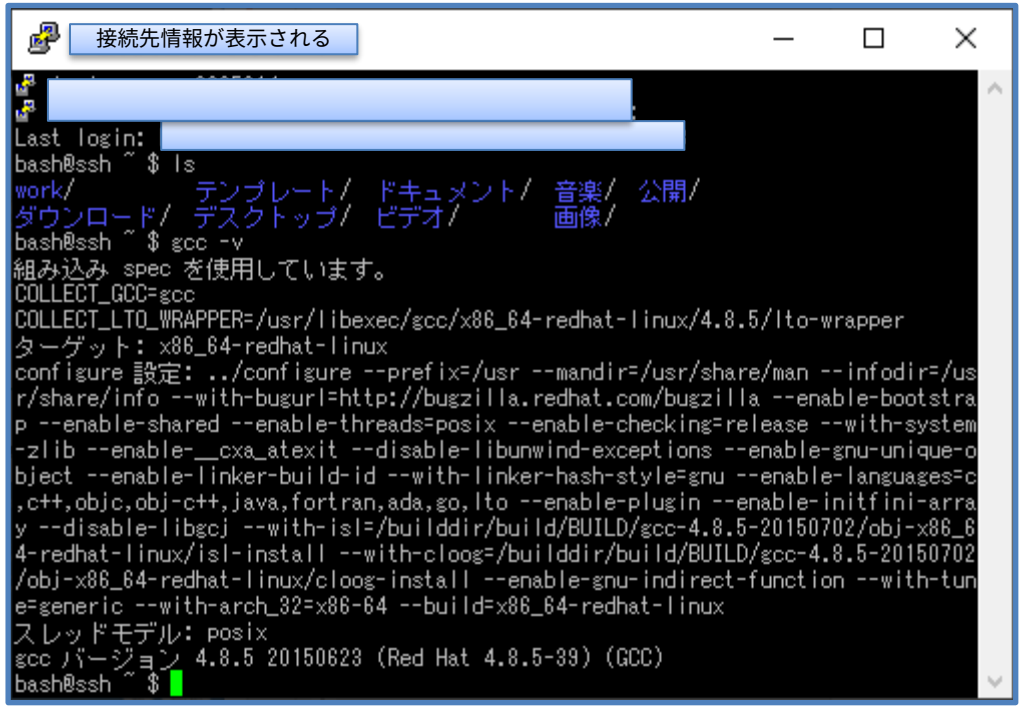
PuTTYを使ってSSH接続する：パスワード認証を使う場合（つづき）



「開く」を押した直後。ログインIDを尋ねられるので入力してEnter。



パスワードIDを求められる。入力しても何も表示が変わらないが、入力してEnter入力でログイン完了。（クリップボードの貼り付けはタイトルバー右クリックから選択、またはメイン画面上で右クリック。）



ログイン後にlsコマンドとgcc -vを実行したところ。接続先にインストールされているプログラムが使えている。

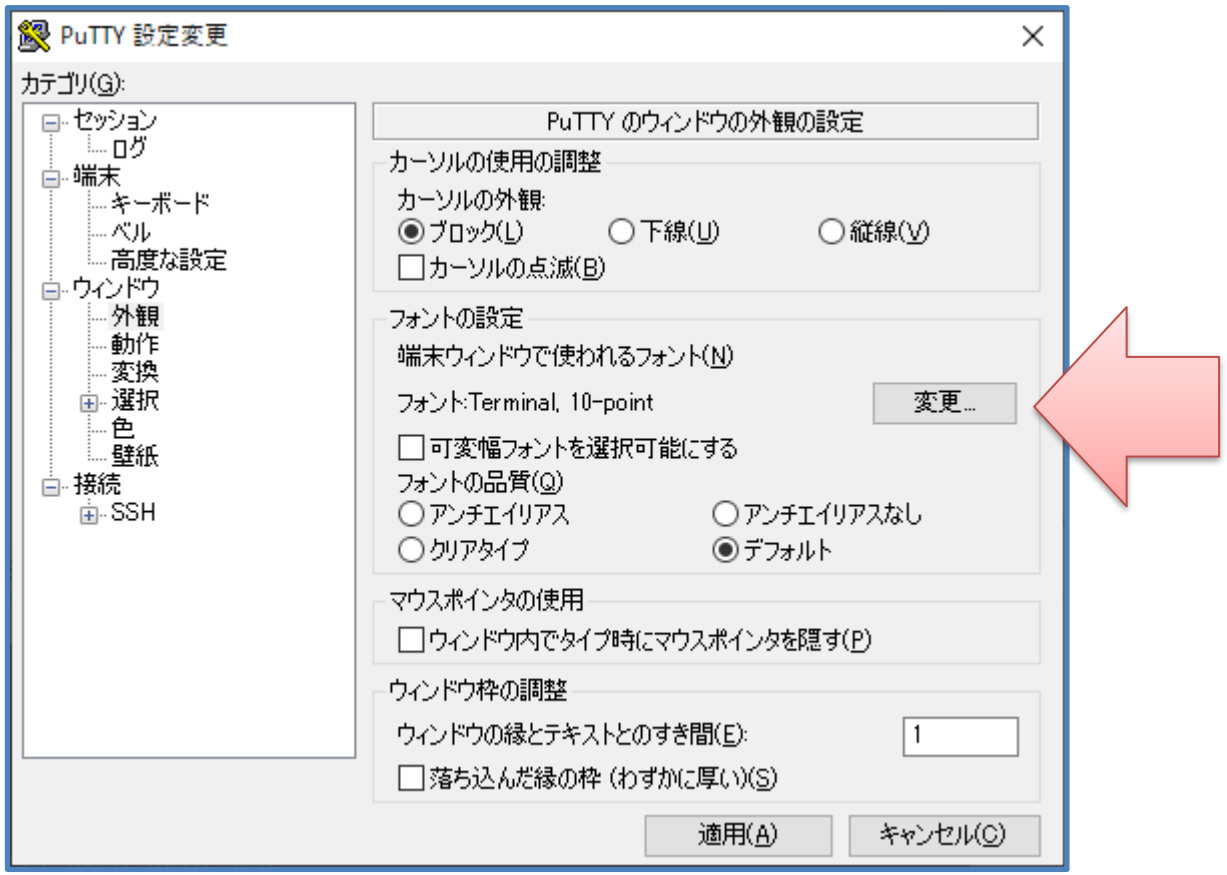
PuTTYの設定

見た目の設定（フォントなど）は
- 接続前のウィンドウ

または
- 接続後にタイトルバーを右クリック→「設定の変更」で表示されるウィンドウ

から変更する

フォント変更の例



Cygwinのインストール方法

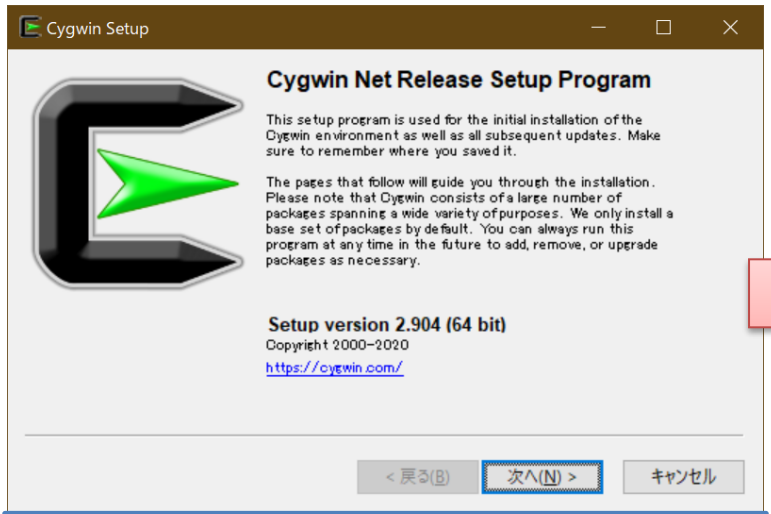
インストール

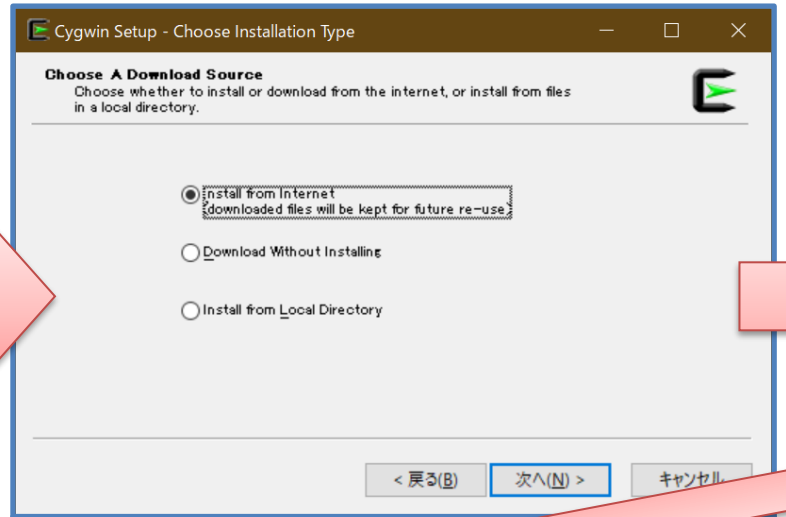
- Webページからダウンロードして実行
  - https://cygwin.com/

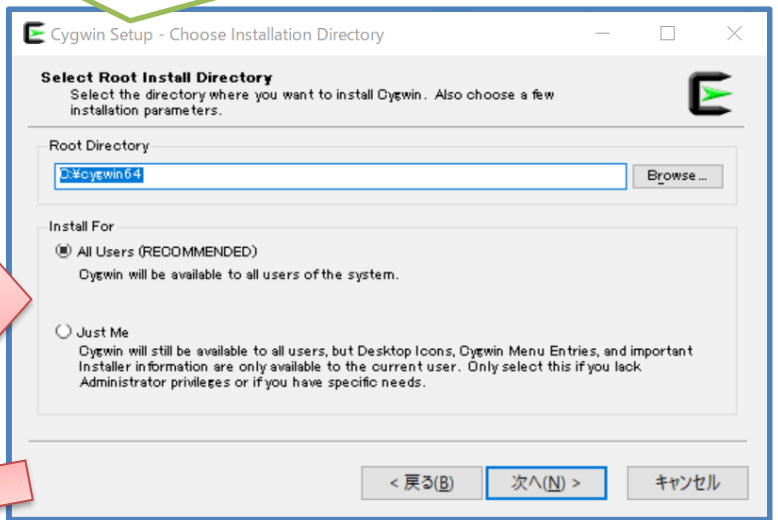
セットアップ手順

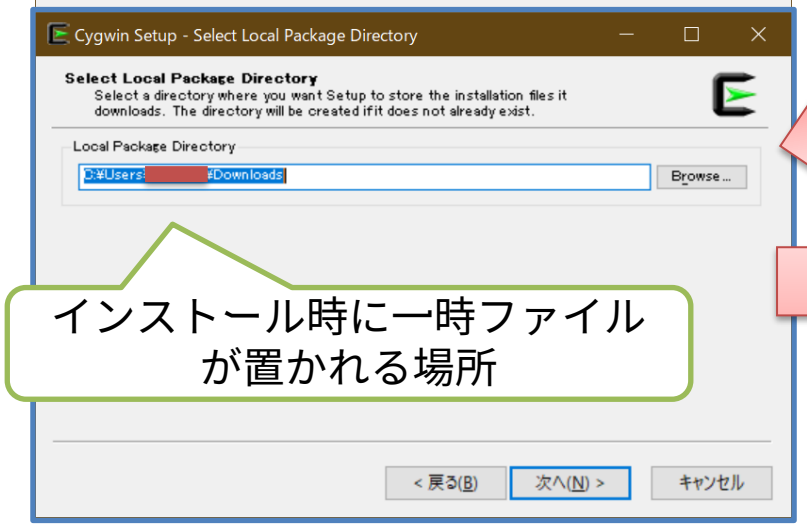
- 基本的には次へ次へ進めるだけ

インストール先、デフォルトはC:¥cygwin64

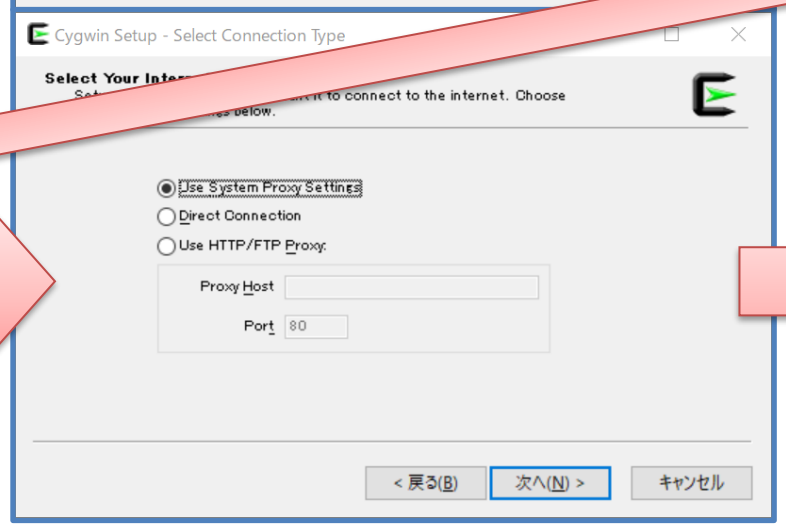


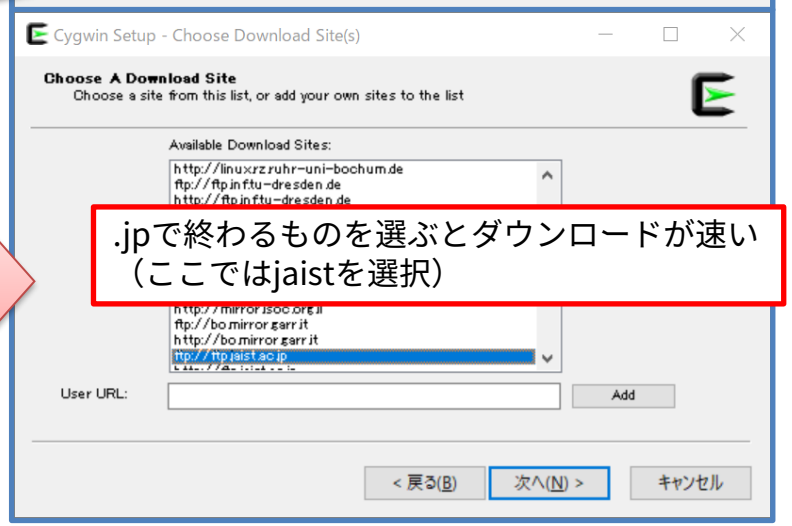






インストール時に一時ファイルが置かれる場所



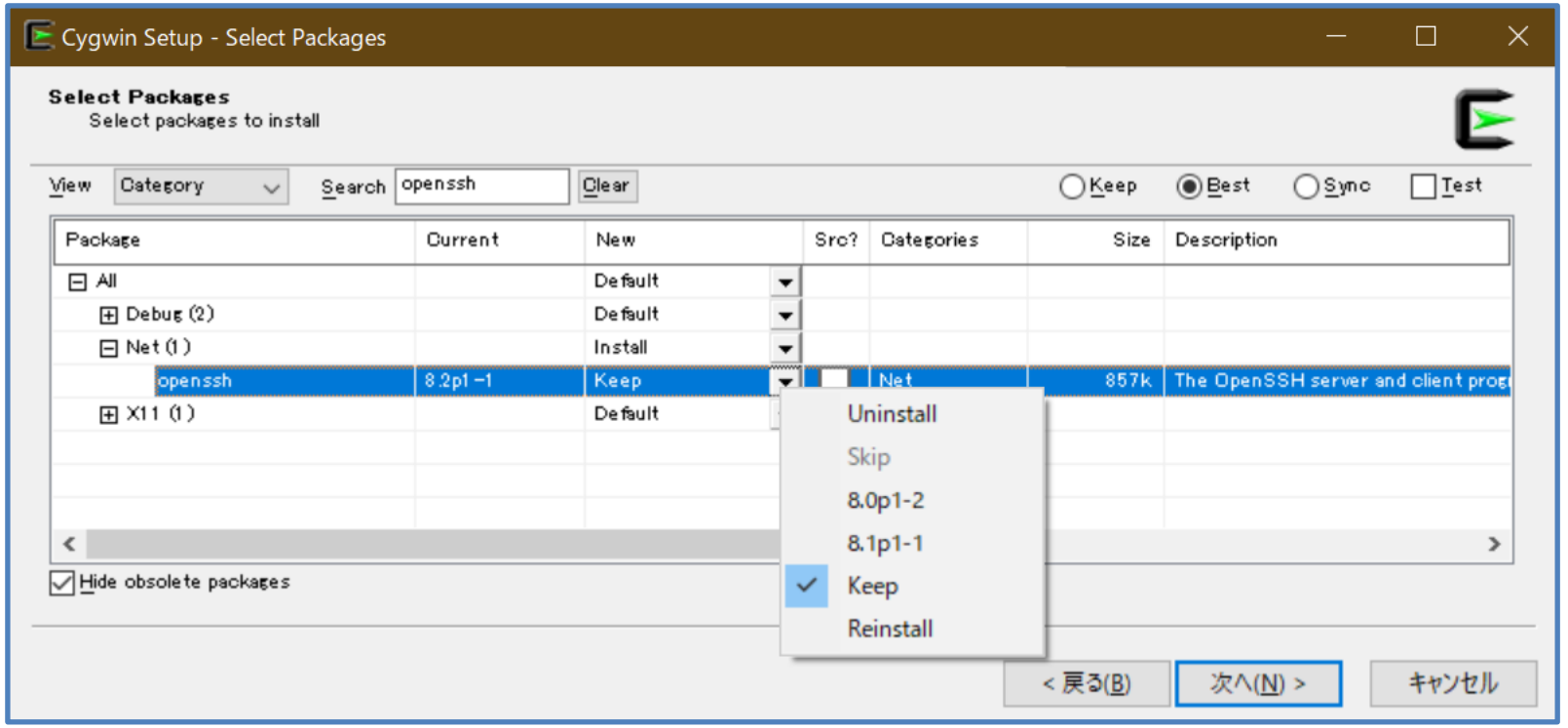


.jpで終わるものを選ぶとダウンロードが速い（ここではjaistを選択）

セットアップ手順：インストールするプログラムの選択

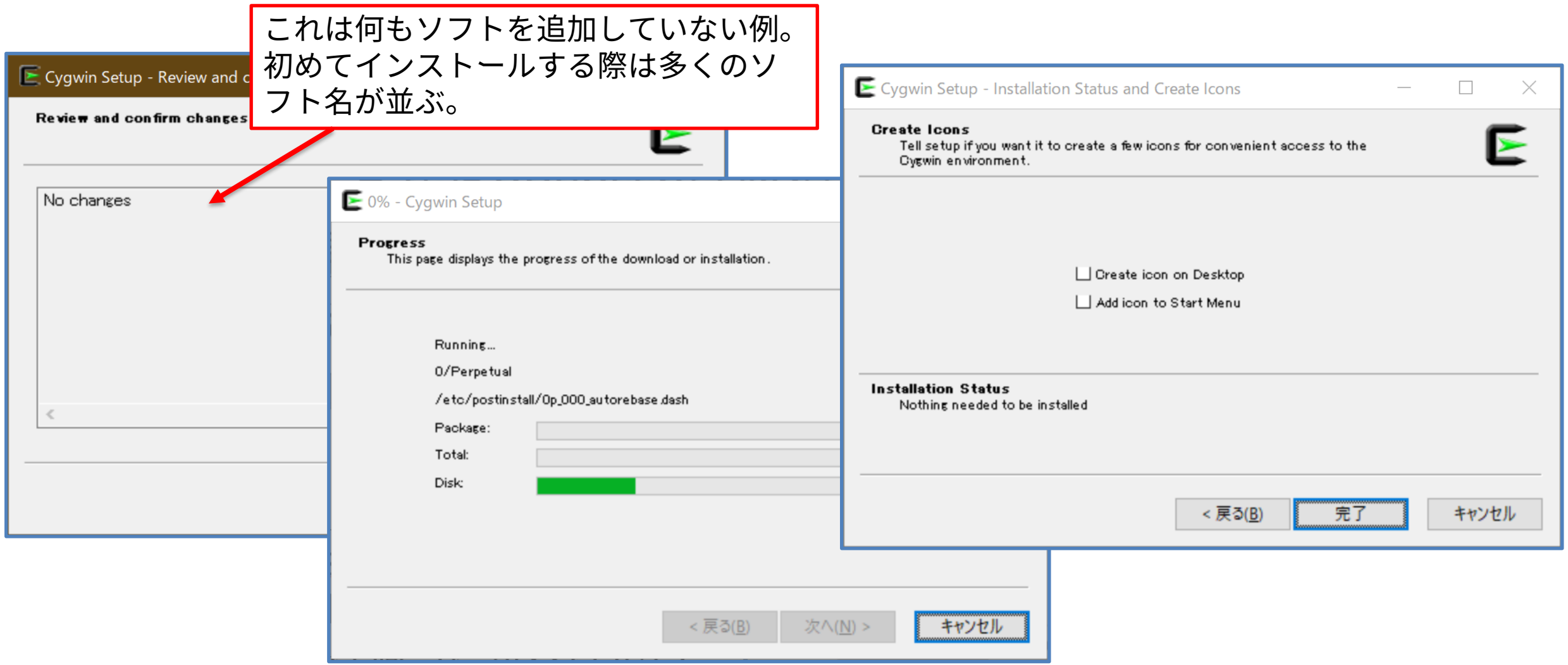
- SSH接続をするためにopensshは必須
- プログラム開発（コンパイル）をするならgcc-core, gcc-fortran, gcc-c++などを選択
- その他、gitやらemacsやら使うものを自由に選ぶ
  - 全部インストールしても構わないが、かなり長い時間とディスク容量が必要

右図は既にopensshがインストールしてある状態のためKeepとなっている。インストール時には適当に選べば良い。（大きな数字のものを選択するとか、画像では「Default」となっているところが「New」になっているはずなので、ここをダブルクリックして最初に出てくるものなど。）



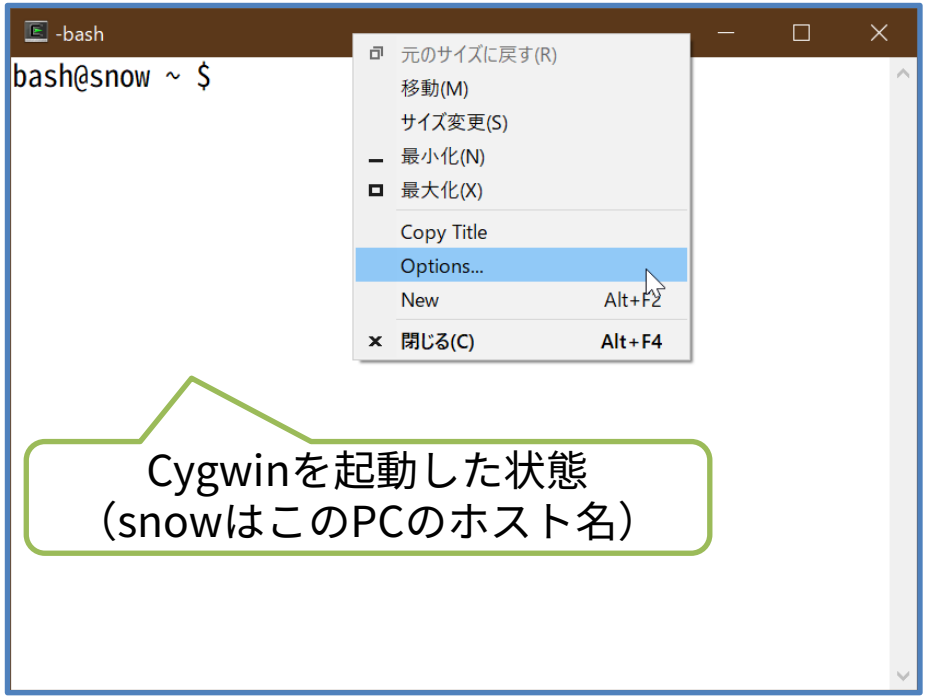
セットアップ手順：あとは待つだけ

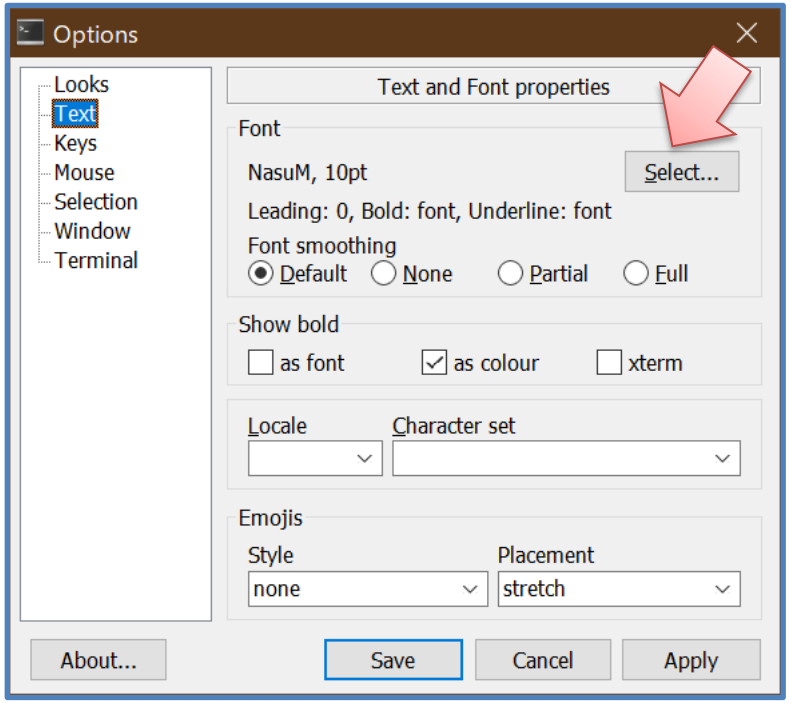
- たくさんのソフトをインストールすると時間がかかるが、待てば良いだけ

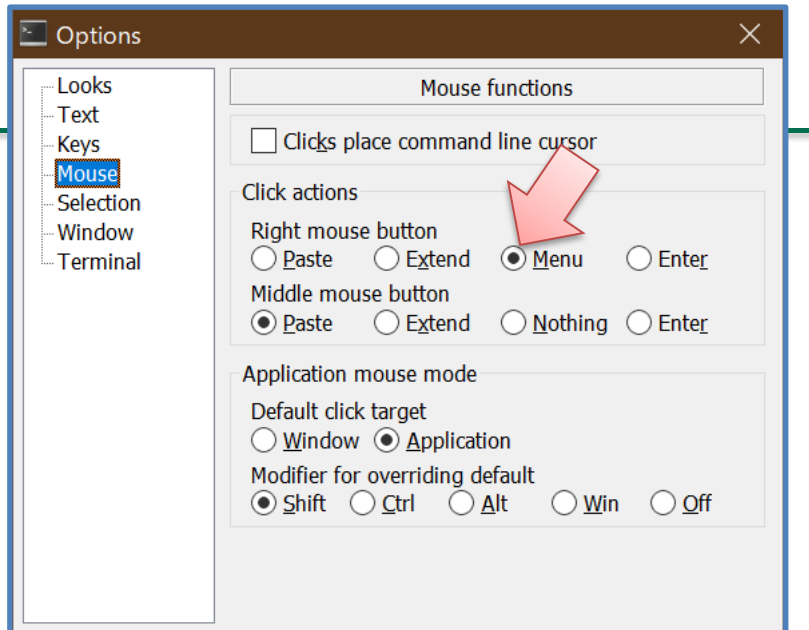


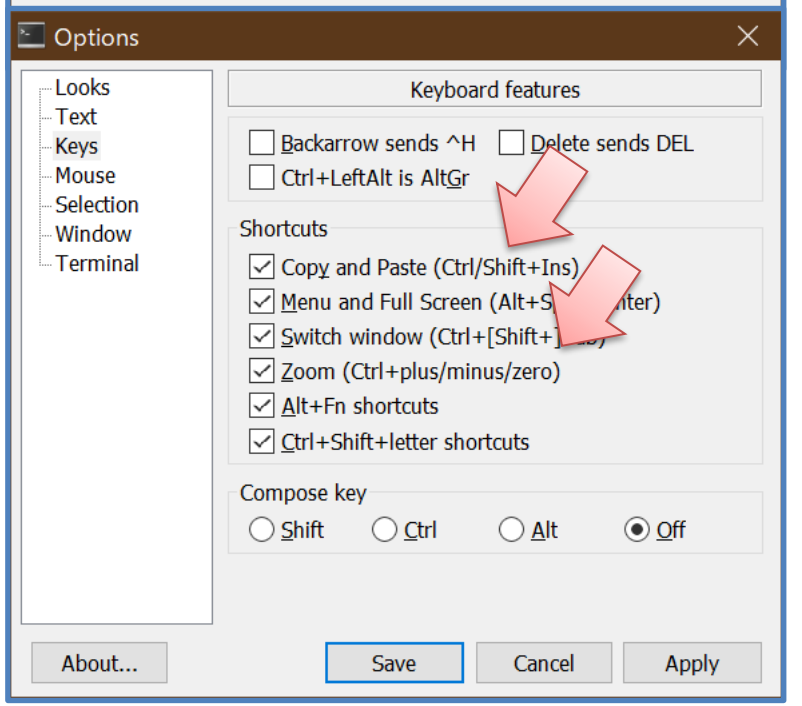
Cygwin起動後

- 基本的にLinuxと一緒
- 見た目（色、フォントなど）の変更などはタイトルバーを右クリックして「Options」から
  - フォントの他、コピーアンドペースト、ズーム、右クリックメニューなどは好みに合わせて調整しておくと使いやすくなる









Cygwin、Linux、MacでSSH接続を行う方法

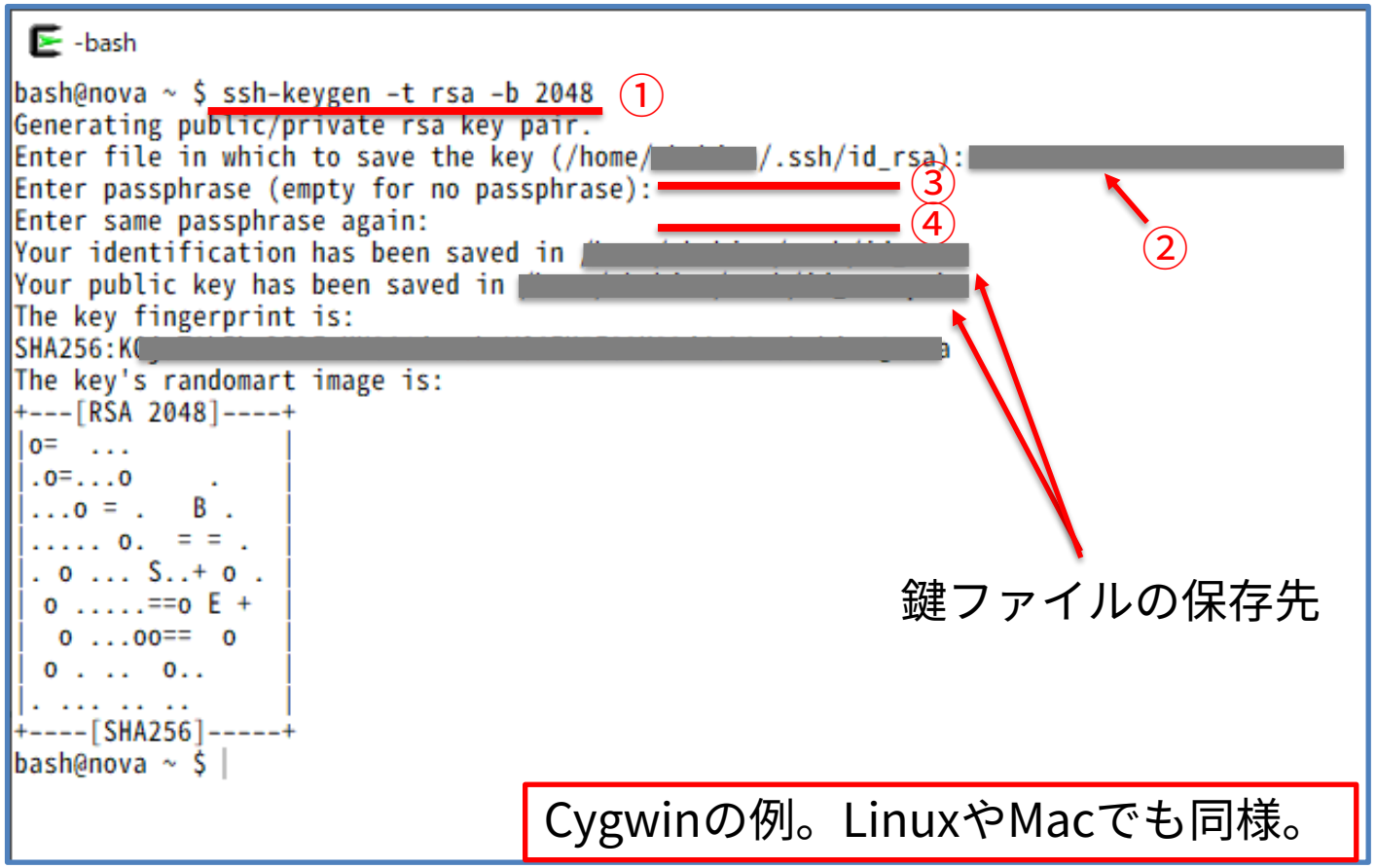
公開鍵ファイルの作成

- ssh-keygenプログラムで作成する
  - ① ssh-keygen -t rsa -b 2048　を実行（エラーが出てしまう場合は-t rsa -b 2048を消す）
  - ② ファイル保存先を指定（初めて作る時はそのままEnterで良い、公開鍵には.pubが追加される）
  - ③ パスフレーズを入力してEnter
  - ④ パスフレーズを再度入力してEnter

※③と④は入力しても表示されない

- 公開鍵ファイルは
  ssh-rsaで始まる一行の文字列
  - ランダムな文字列の後、空白、
    ユーザ名@作成したホスト名で終わる

```
$ cat ~/.ssh/id_rsa.pub
ssh‐rsa
AAAA… … …　… … … … … … … … …
(中略、英数字がしばらく続く)
… … … … … nmsbl6jA2N userX@host
```



SSH接続

- 基本
  - ssh ホスト名
  - 例：ssh example.com
  - 接続時にユーザ名とパスワードまたは公開鍵パスフレーズの入力を求められる

- ユーザ名や秘密鍵を明示する場合
  - ssh -i 秘密鍵ファイル ユーザ名@ホスト名
  - 例：ssh -i .ssh/id_rsa name@example.com

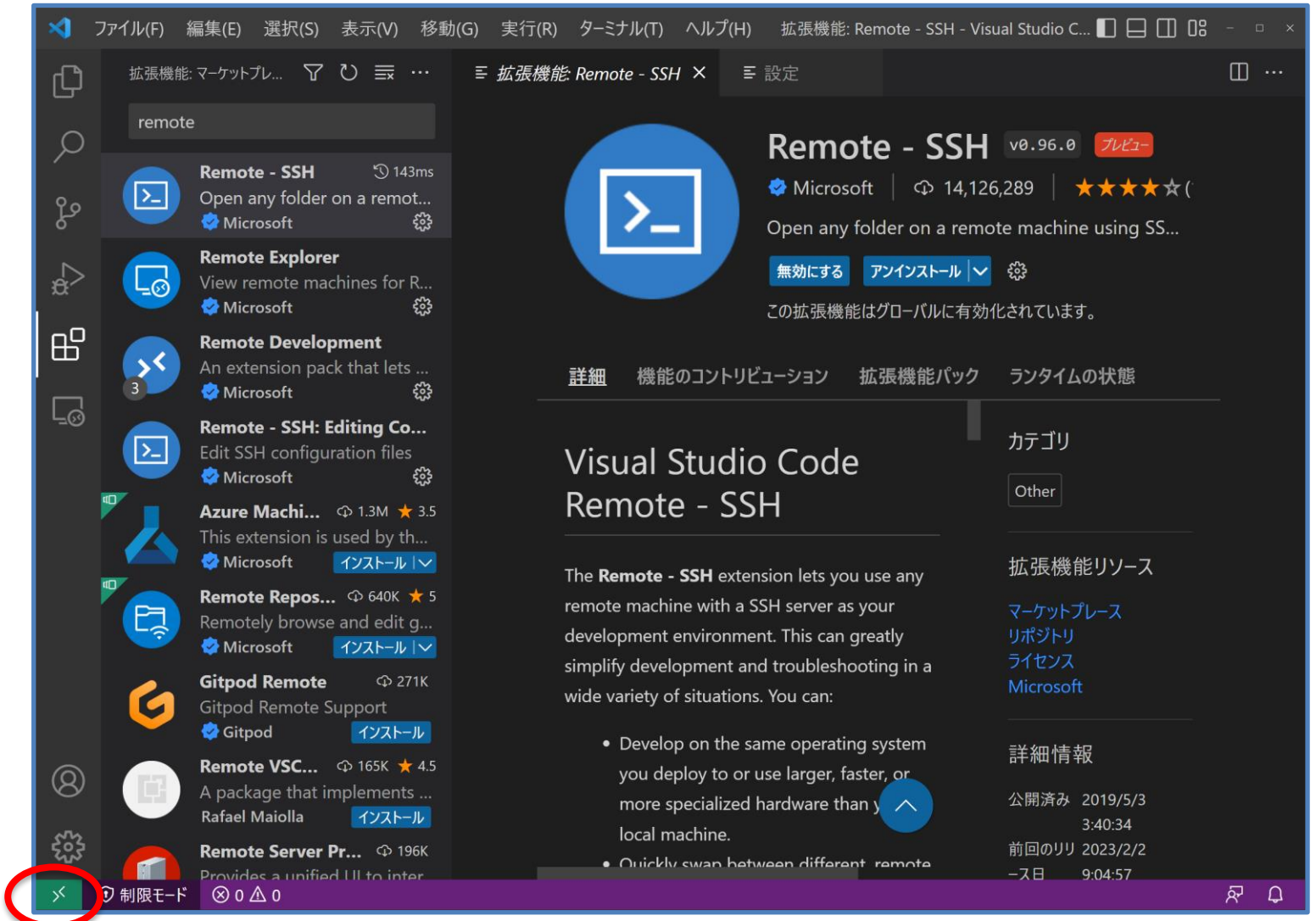
- 入力の省略などをしたい場合
  - .ssh/config
  - ssh-agent
  - あたりについて調べてみましょう

- ここでいうユーザ名（name）とは接続先のログインIDのこと
- 「不老」にssh接続する場合は、例えば
  ssh w49999a@flow-fx.cc.nagoya-u.ac.jp
  や
  ssh w49999a@flow-cx.cc.nagoya-u.ac.jp
  のようになる

VSCodeによるSSH接続

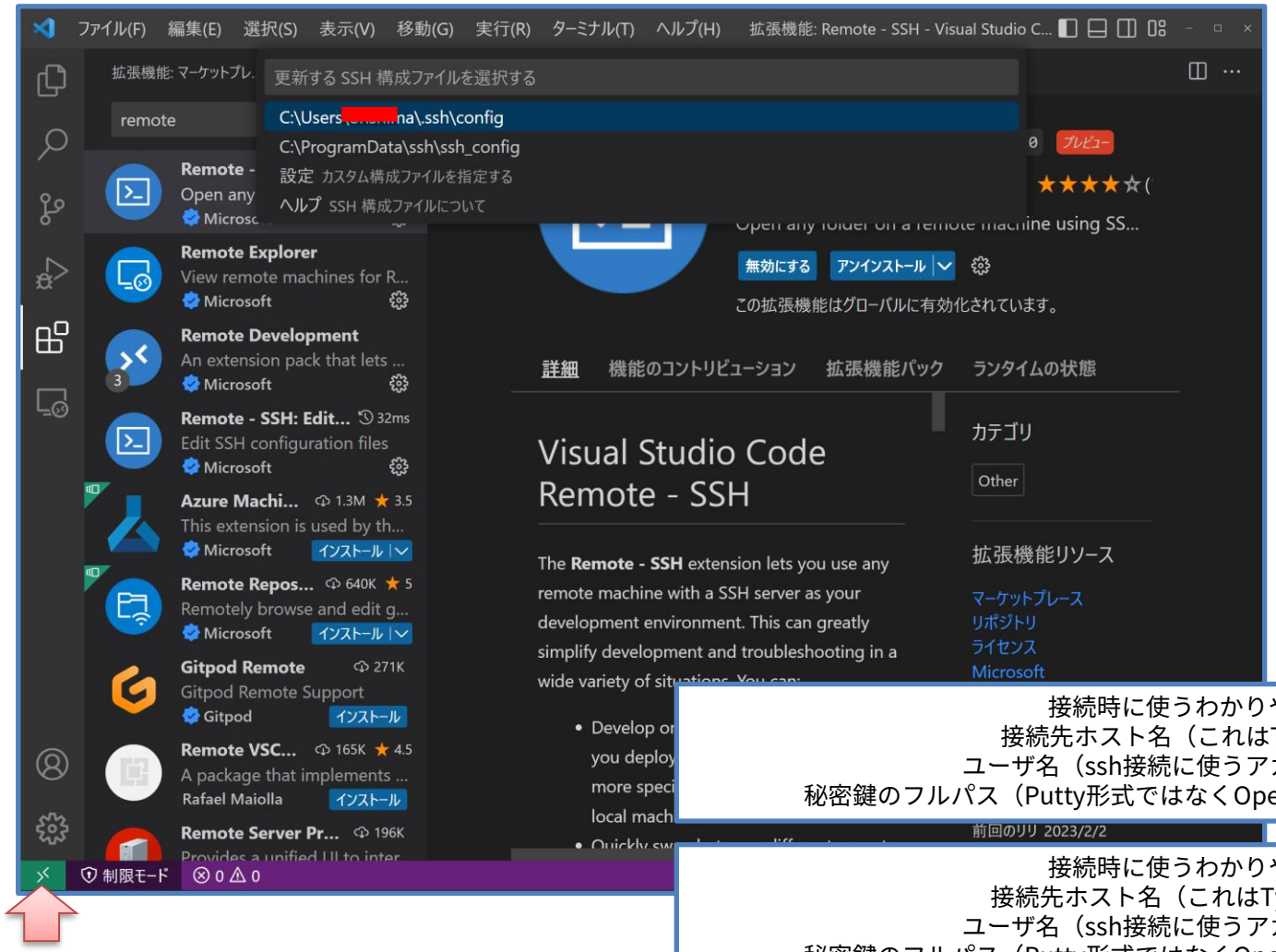
準備

- 拡張機能「Remote – SSH」をインストールしておく



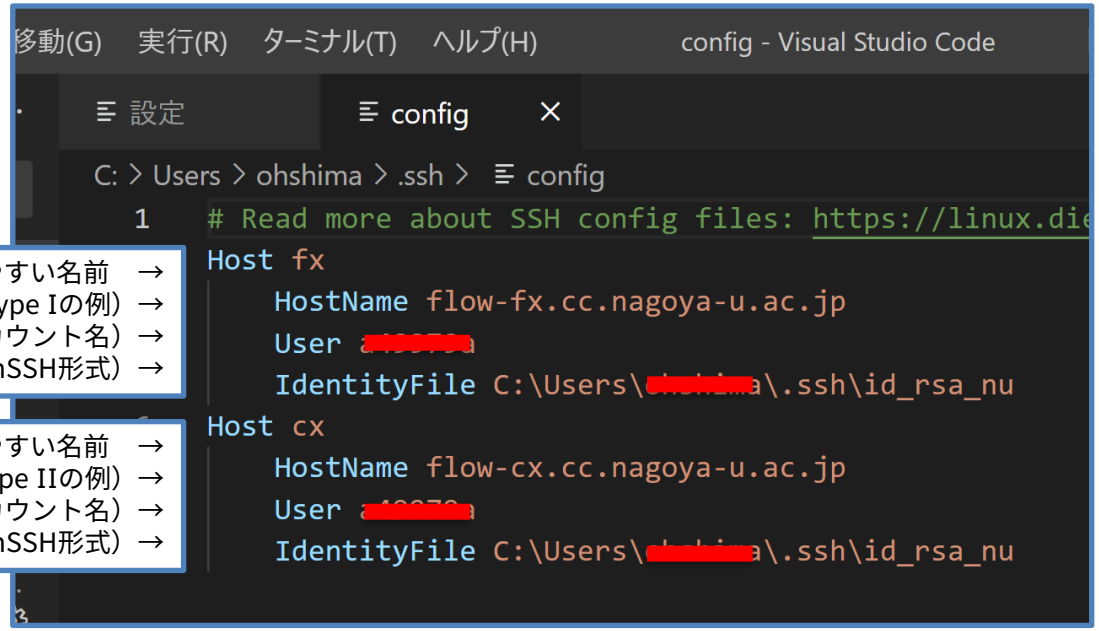
歯車アイコンまたは左下の><の部分から
設定画面を開くと接続設定を行える

ssh接続の設定



左下の><の部分を選び、上部から
「SSH 構成ファイルを開く...」　を選び、さらに
C:¥Users¥ユーザ名¥.ssh¥config　を選ぶ。

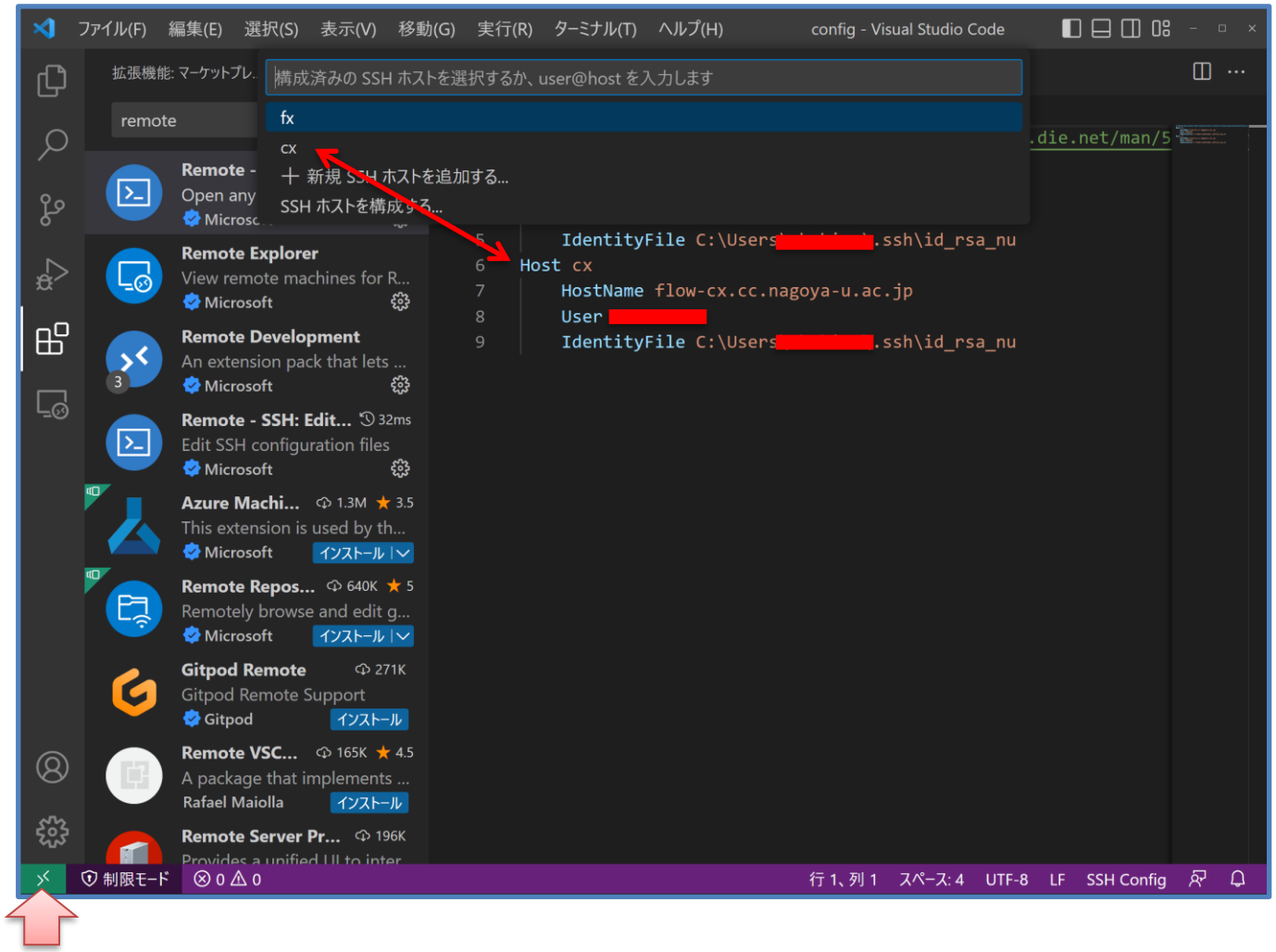
ssh設定ファイルが開くので、下図のような情報を書き込み保存する。（これはType IとType IIのログインノードへの接続設定を書いた例。）



接続時に使うわかりやすい名前　→
接続先ホスト名（これはType Iの例）→
ユーザ名（ssh接続に使うアカウント名）→
秘密鍵のフルパス（Putty形式ではなくOpenSSH形式）→

接続時に使うわかりやすい名前　→
接続先ホスト名（これはType IIの例）→
ユーザ名（ssh接続に使うアカウント名）→
秘密鍵のフルパス（Putty形式ではなくOpenSSH形式）→

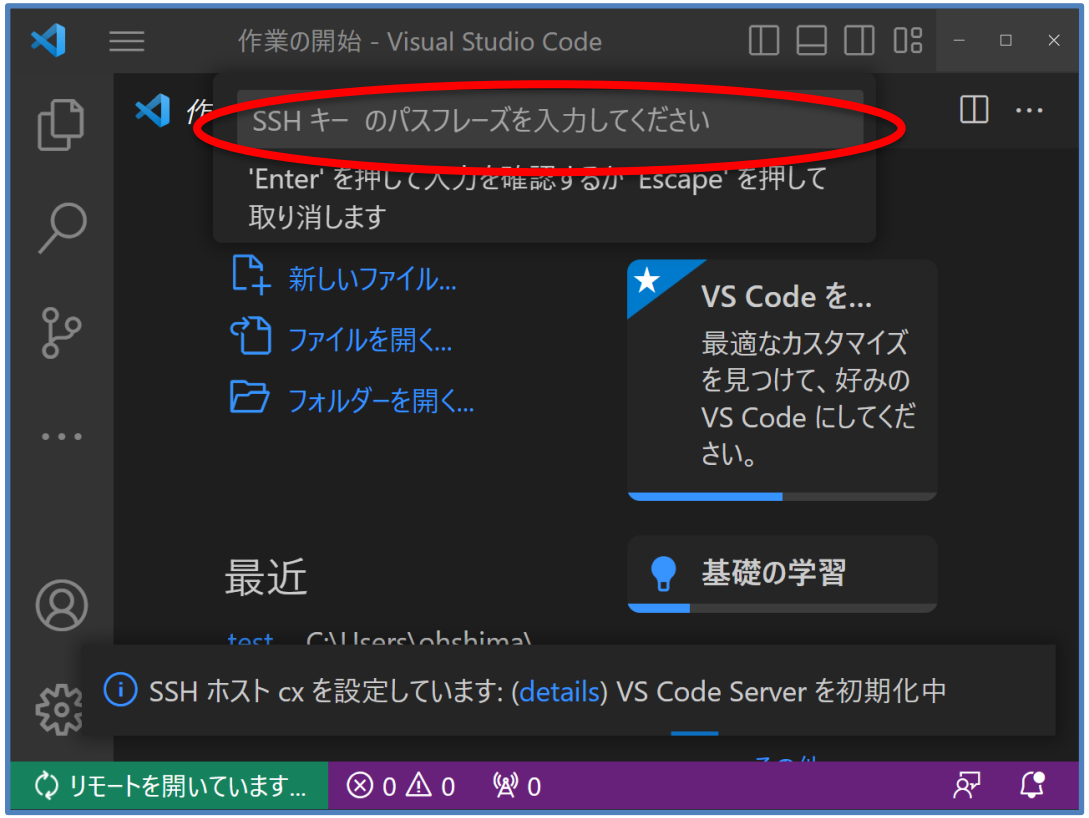
接続



左下の><の部分を選び、上部から
「ホストに接続する...」　を選ぶと、
先ほど設定した接続名が選べるようになっているはずなので、これを選ぶ。

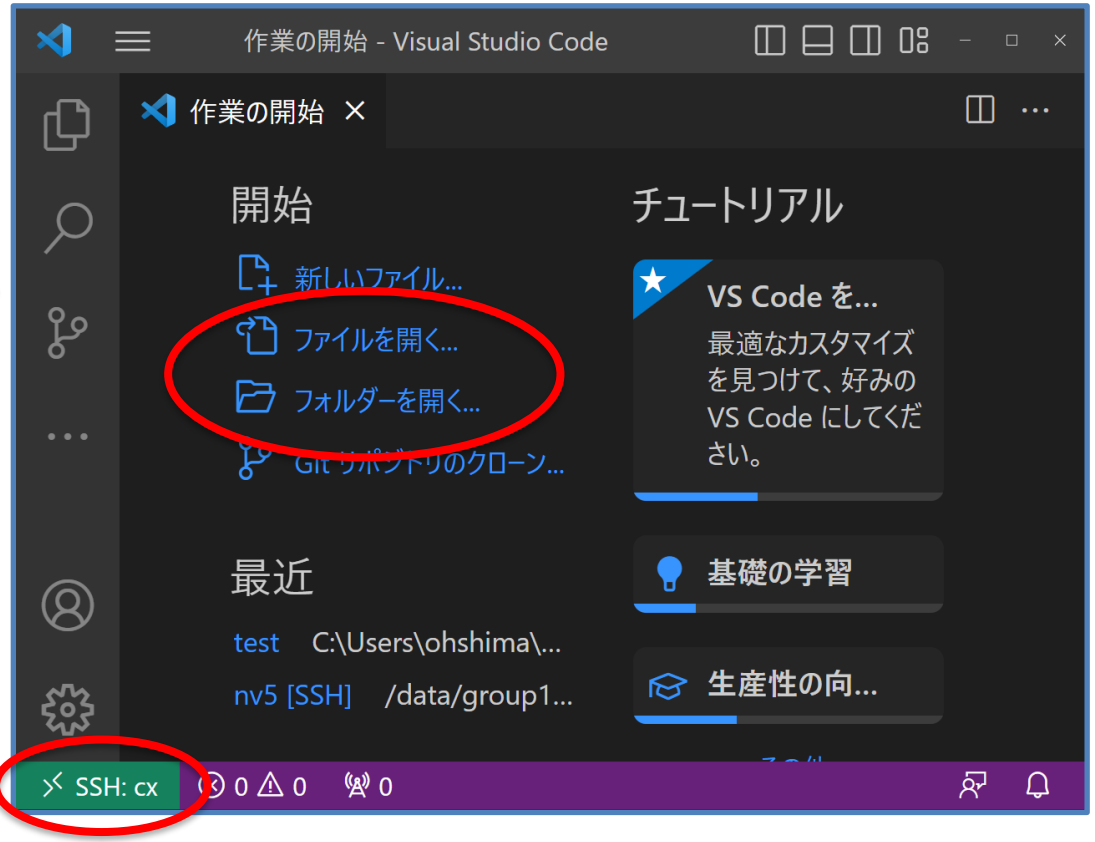
接続（続き）

- 新しいウィンドウが開く
- パスフレーズの入力が求められるので、入力してEnter



接続された状態

- 左下に接続先名が表示される
- ファイルやフォルダを開こうとするとリモートの情報が見える



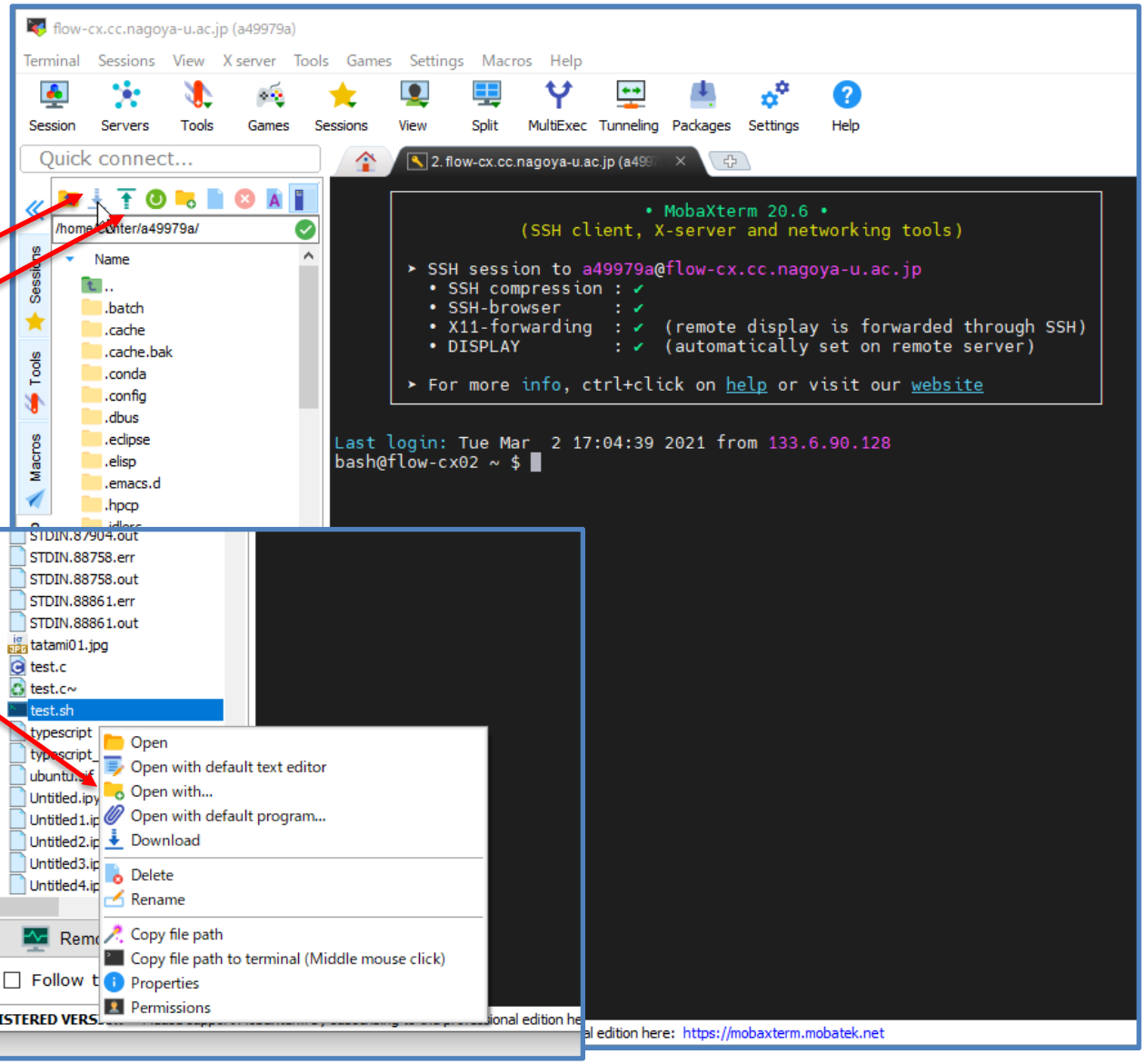
補足

SSH接続先と手元のPCとの間でファイルの送受信をする方法

- 方法1：scpコマンドやsftpコマンドを使う（MacやLinuxではすぐに使えるはず）
- 方法2：MobaXtermの左側ツリーウィンドウを使う（右クリックメニューで転送できる）
- 方法3：WinSCPやFileZillaなどのツールを使う
  - http://winscp.net/jp/
  - https://filezilla-project.org/
- いずれもSSHログインと同様にパスワードや公開鍵でログインして使う

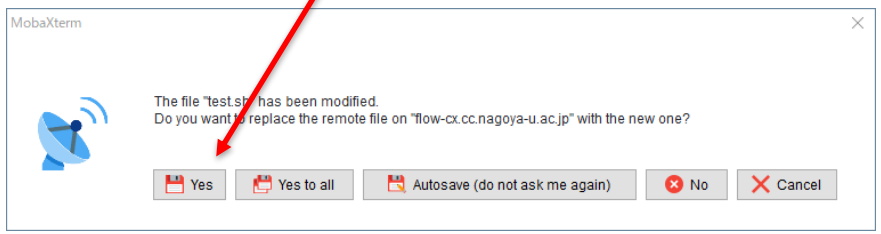
MobaXtermのファイル転送機能など

- 左側のツリービューのアイコンや
  右クリックからファイル転送などができる



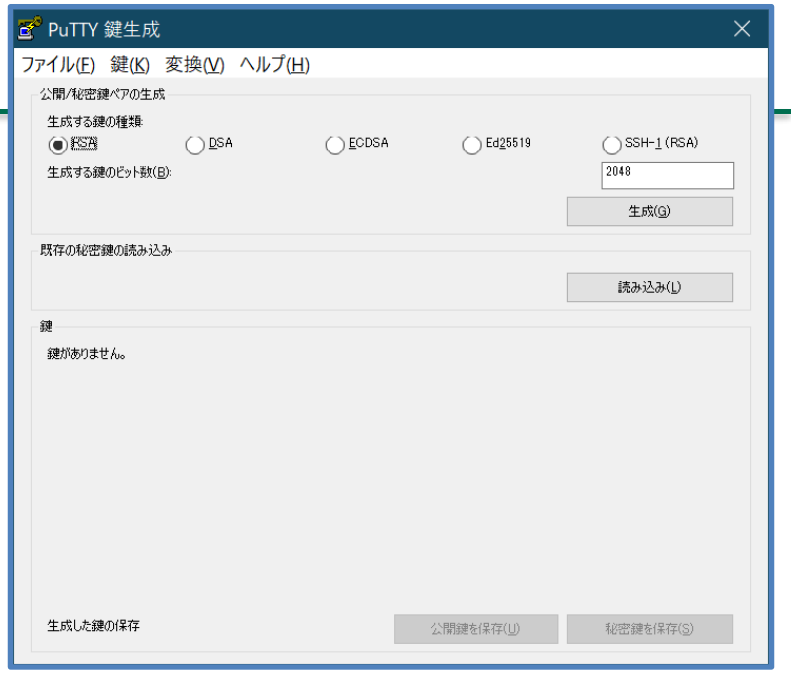
- 選択したファイル・フォルダを手元のPCに
  ダウンロード
- 手元のPCのファイルをアップロード

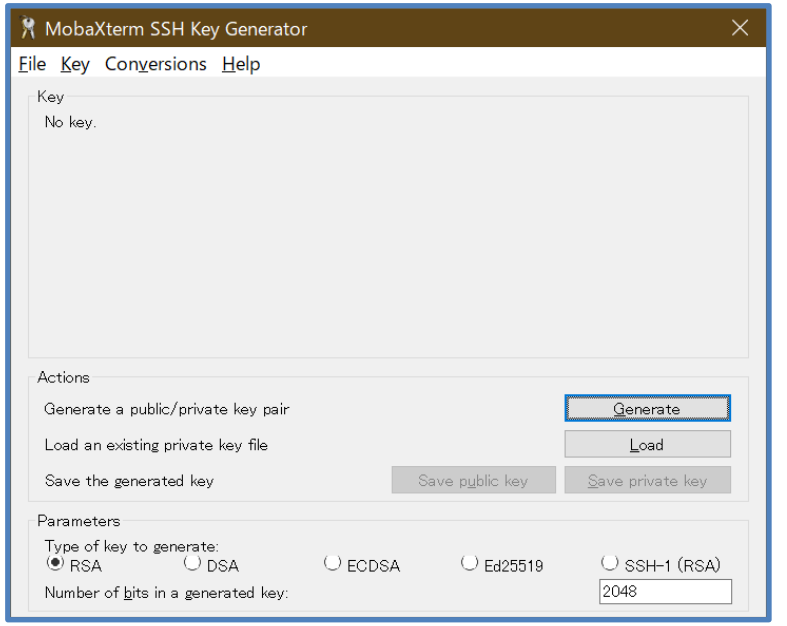
- 右クリックメニューからはファイルの編集
  も可能
  - Open with...を選ぶと手元のPCのエディ
    タで編集できる、保存して終了してYes
    を選ぶとアップロードしてくれる



秘密鍵ファイルの形式変換について

- OpenSSH（CygwinやLinux、Macで使う）が使う秘密鍵ファイル（一般的にはid_rsaファイル）とPuTTYやMobaXtermが使う鍵ファイル（ppkファイル）は形式が異なる
- puttygen/MobaKeygenを使えば形式の相互変換が可能
  - OpenSSH形式の秘密鍵ファイルを読み込む
    - 「変換」メニューの「鍵のインポート」
    - 「Conversions」 メニューの「Import key」
  - ppkファイルの秘密鍵ファイルを読み込む
    - 「読み込み」ボタン、「Load」ボタン
  - OpenSSH形式の秘密鍵ファイルを書き出す
    - 「変換」メニューの「OpenSSH形式へエクスポート」
    - 「Conversions」メニューの「Export OpenSSH key」
  - ppkファイルの秘密鍵ファイルを書き出す
    - 「秘密鍵を保存」ボタン、「Save private key」ボタン





画面転送（X転送）、その他

- 画面転送（X転送）
  - 環境によってはSSH接続先で表示される画面（ウィンドウ）を手元で見ることができる
  - MobaXtermではSSH接続時にデフォルトで「X11-Forwarding」が有効なので手元の設定は不要
  - Linux, Macではsshコマンドに-Yオプションを付ける
  - Cygwinでも-Yオプションで良いが、画面を表示するプログラムが別途必要
    - Cygwinインストーラでxorg-serverとxinitを追加し、スタートメニューからXWin Serverを起動しておく

- パスフレーズ入力の省略について
  - puttyのpageant.exeやOpenSSHのssh-agentを使うことで接続の度にパスフレーズを入力する手間を省くことができる。
  - 詳細は各自で調べてください。

FAQ（よくある問題と解決方法）

- Permission denied (publickey)　というエラーが出る（主にLinux環境）
  - .sshディレクトリと秘密鍵のパーミッションの問題の可能性があります。
  - chmod 700 ~/.ssh と chmod 600 ~/.ssh/id_rsa を実行すると改善しないでしょうか？
  - それでも改善しない場合は、（CygwinではなくWindows自体の）ファイルやディレクトリのプロパティの「セキュリティ」設定が必要かもしれません

- 公開鍵の追加や変更をしたい
  - 登録した公開鍵はスパコン側の~/.ssh/authorized_keysに書き込まれています。このファイルを編集すれば追加や変更が可能です。1行あたり1公開鍵です。編集に失敗すると二度とログインできなくなりますので、ssh接続を維持した状態で新たに接続できるかを確認するようにしてください。
  - ログインできなくなってしまった場合は公開鍵が未登録の状態に戻しますのでQ&Aにお問い合わせください。