

情報工学演習1

第2回

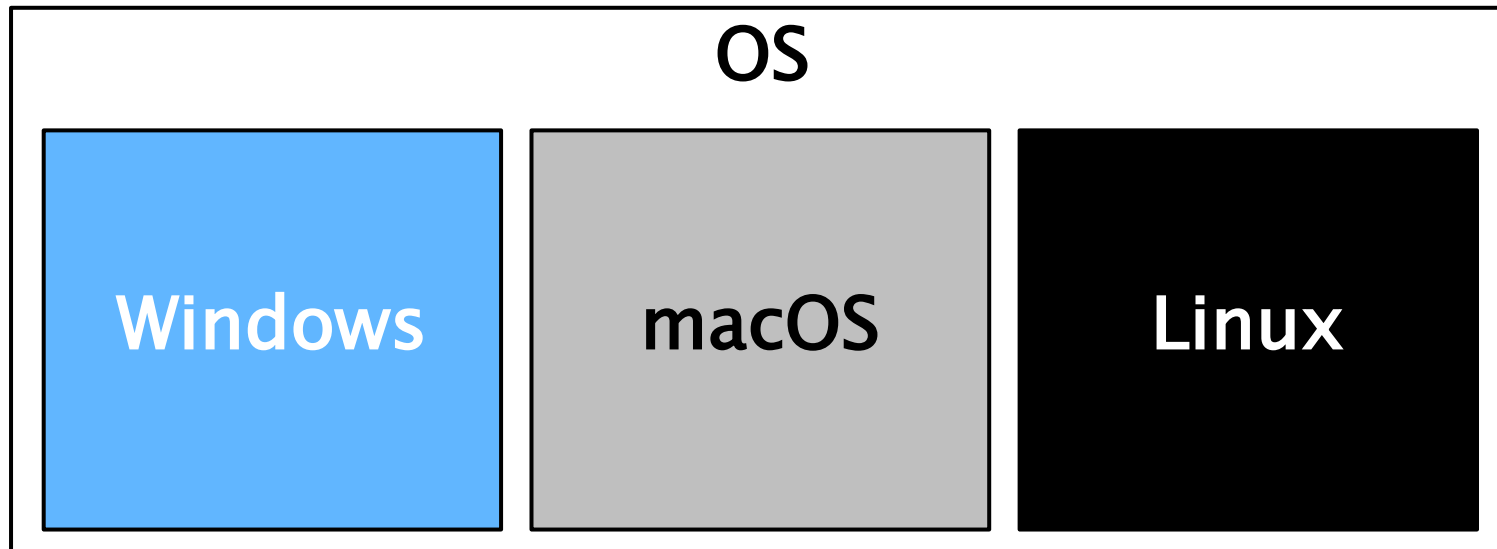
Linuxの使用方法1 (基本コマンド, 絶対パス・相対パス)

今回の目標

- ▶ Linuxのコマンドの使い方に慣れる.

Linux

- ▶ OS (Operating System)の一種.
- ▶ オープンソースのOS.
- ▶ 無料で使用できる.
- ▶ サーバーでよく使われる.
- ▶ プログラミングの世界でよく使われる.



Linuxコマンド

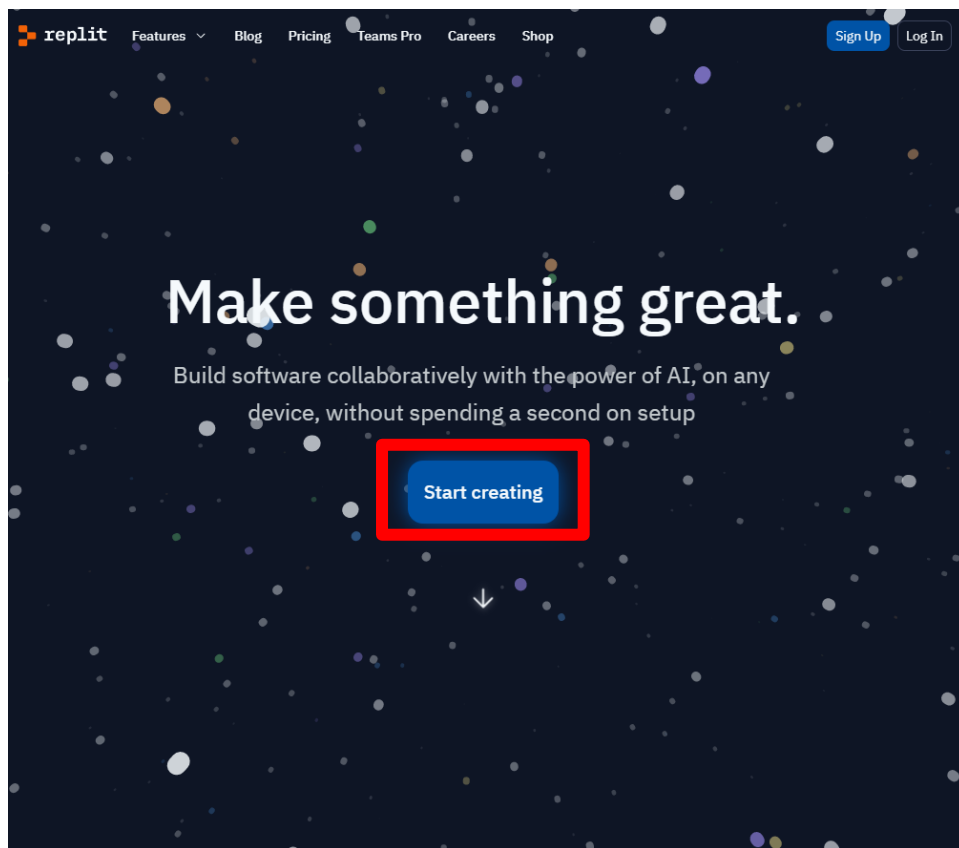
- ▶ PCを操作する方法は以下に大別される：
 - ▶ **GUI** (Graphical User Interface)
 - ▶ ウィンドウ, アイコン, ボタンなどをマウス等で選択して操作する.
 - ▶ 直感的に操作が可能.
 - ▶ マウスで多くの操作を行うのは大変.
 - ▶ **CUI** (Character User Interface)
 - ▶ キーボードでコマンドを入力して操作する.
 - ▶ 一連の処理を自動化できるため, 慣れるととても速い.
 - ▶ 操作の記録が残るため, 同様の操作をすぐに再実行できる.
- ▶ **Linuxコマンドによる操作はプログラミングでは必須のスキル.**

Linuxコマンドを試してみる

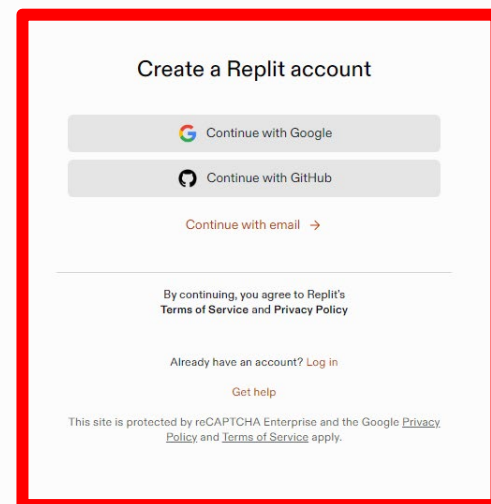
- ▶ 本演習では、**Web上(ブラウザ上)でプログラミングやLinuxの操作が可能なサービス**を使用する.
- ▶ **Replit**
<https://replit.com/>



Linuxコマンドを試してみる@Replit



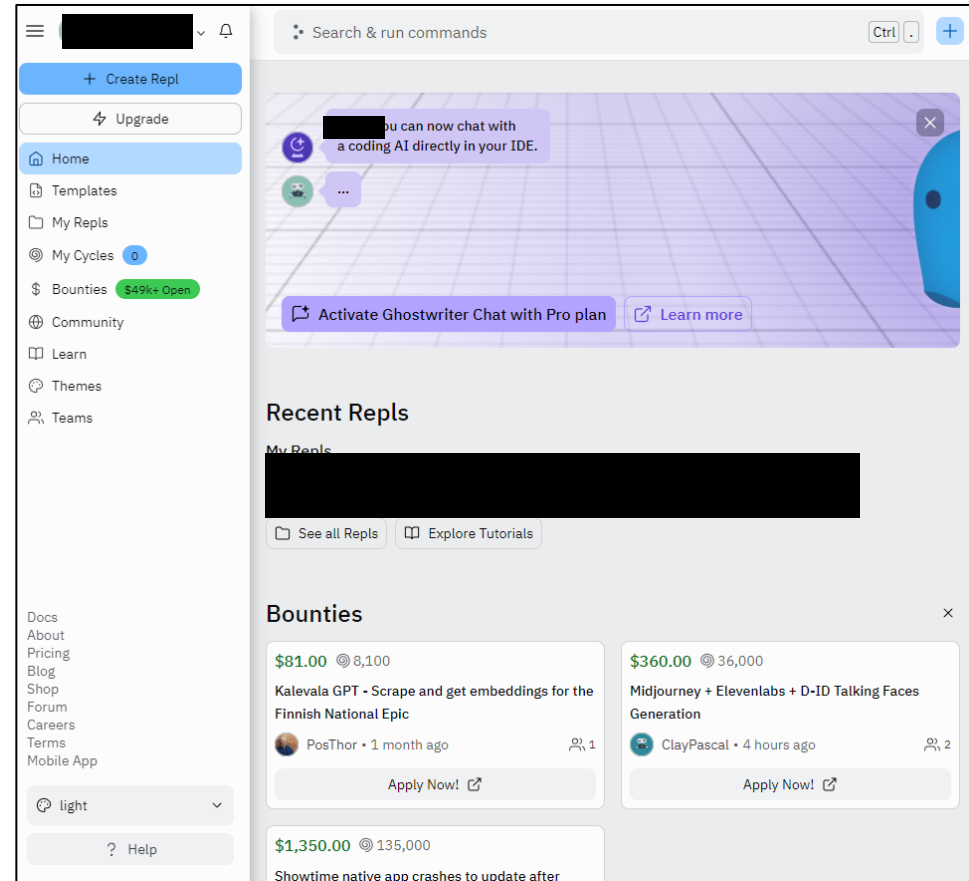
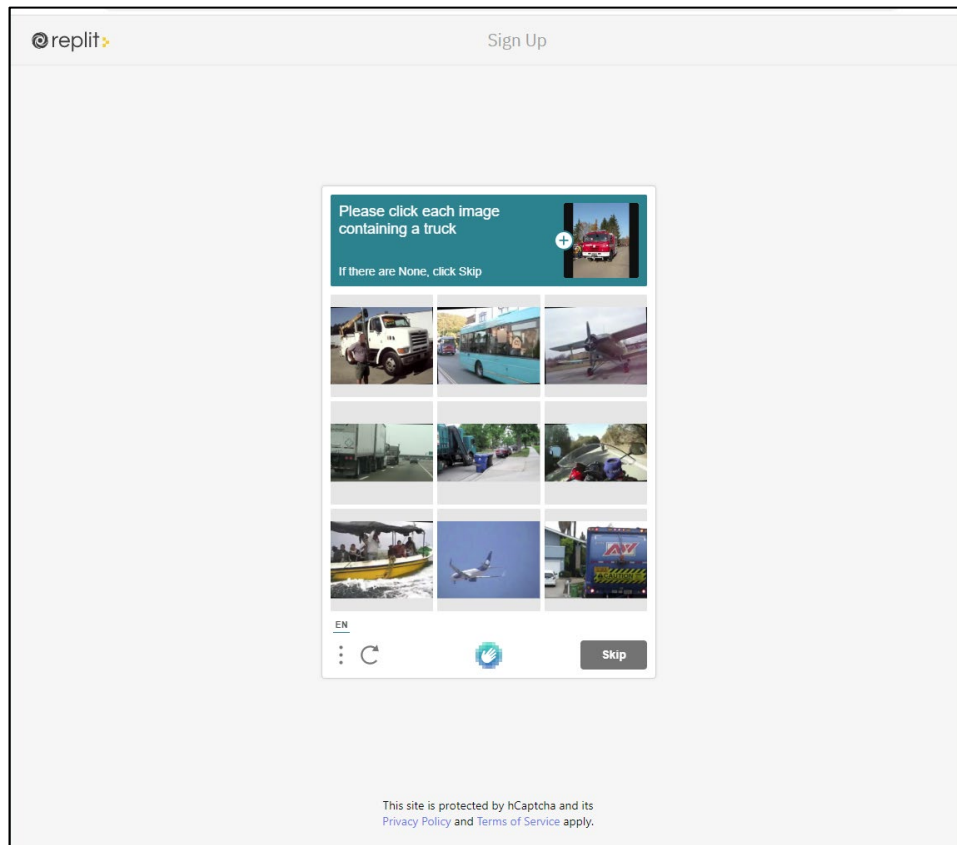
ブラウザでRepl.it (<https://replit.com/>) にアクセスし, 「Start building」をクリックする.



アカウントを持っていない場合,
作成する必要がある。
Google や GitHub アカウントでログイン
するか, emailでアカウントを作成する。

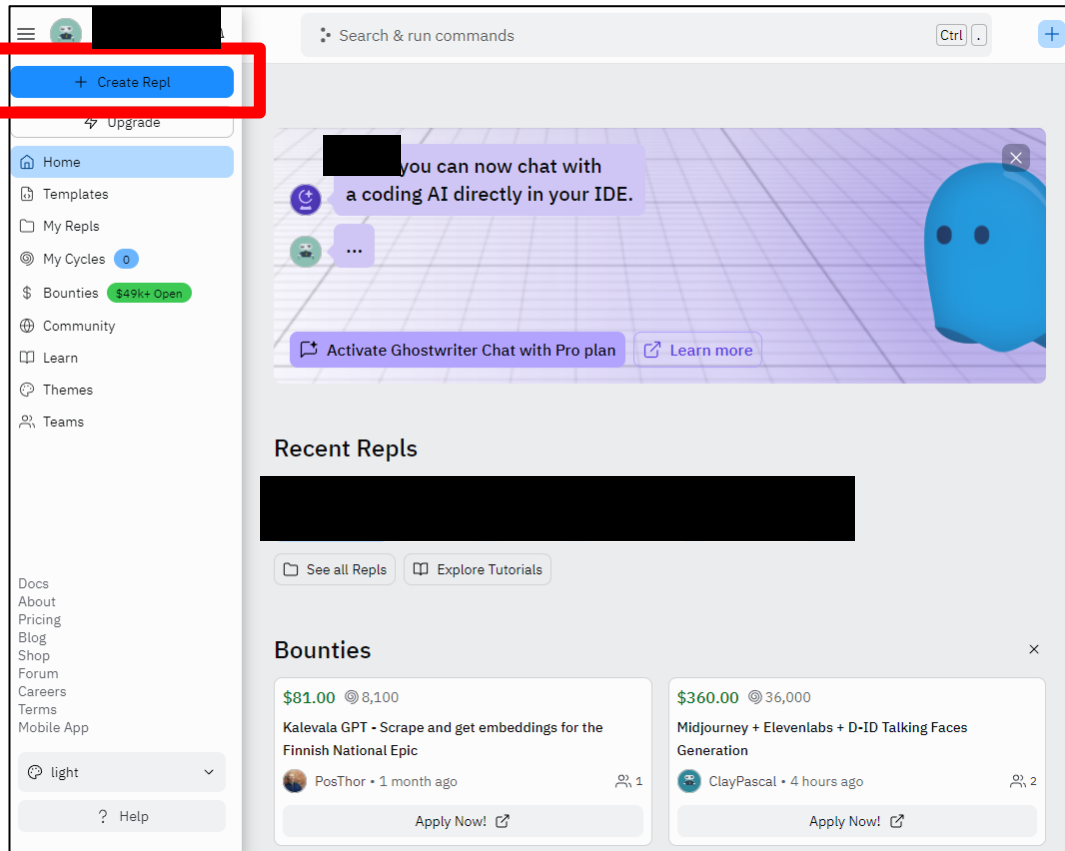
説明が古い場合, 適宜最新の
情報に読み替えてください

Linuxコマンドを試してみる@Replit

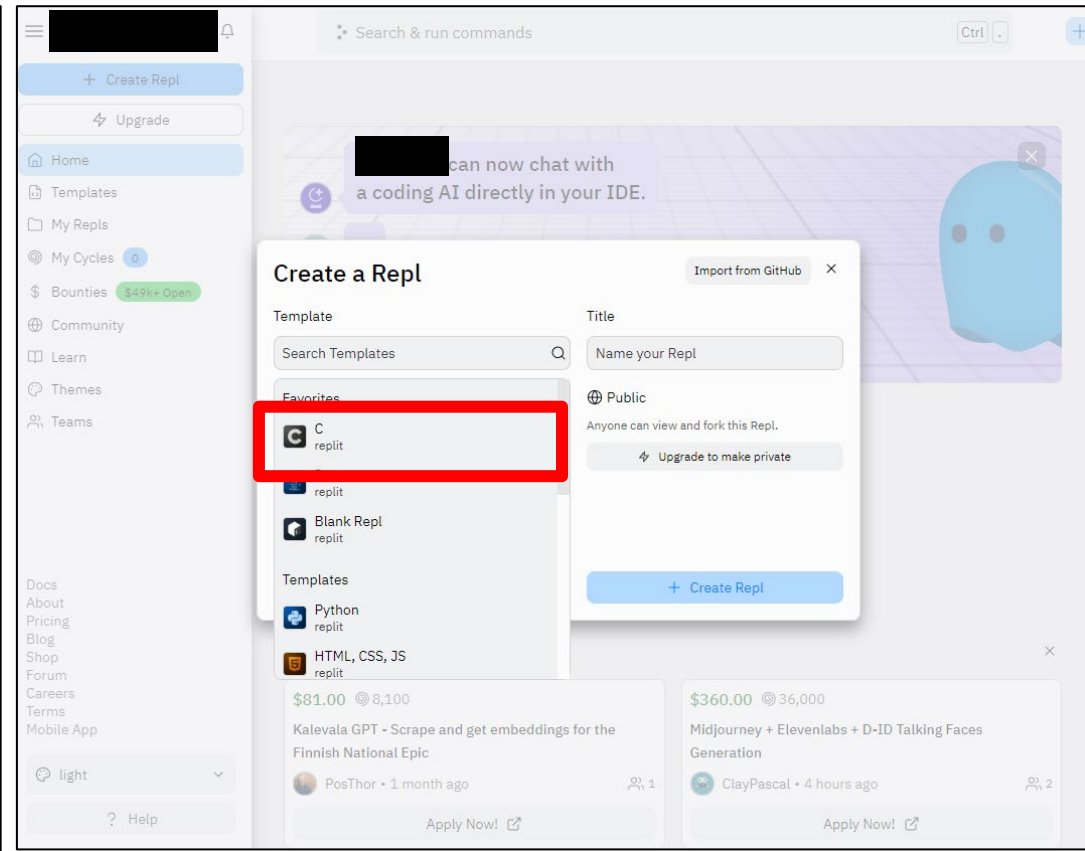


必要に応じて、画像認証や、メール認証を行い、アカウントを有効にし、ログインする。

Linuxコマンドを試してみる@Replit

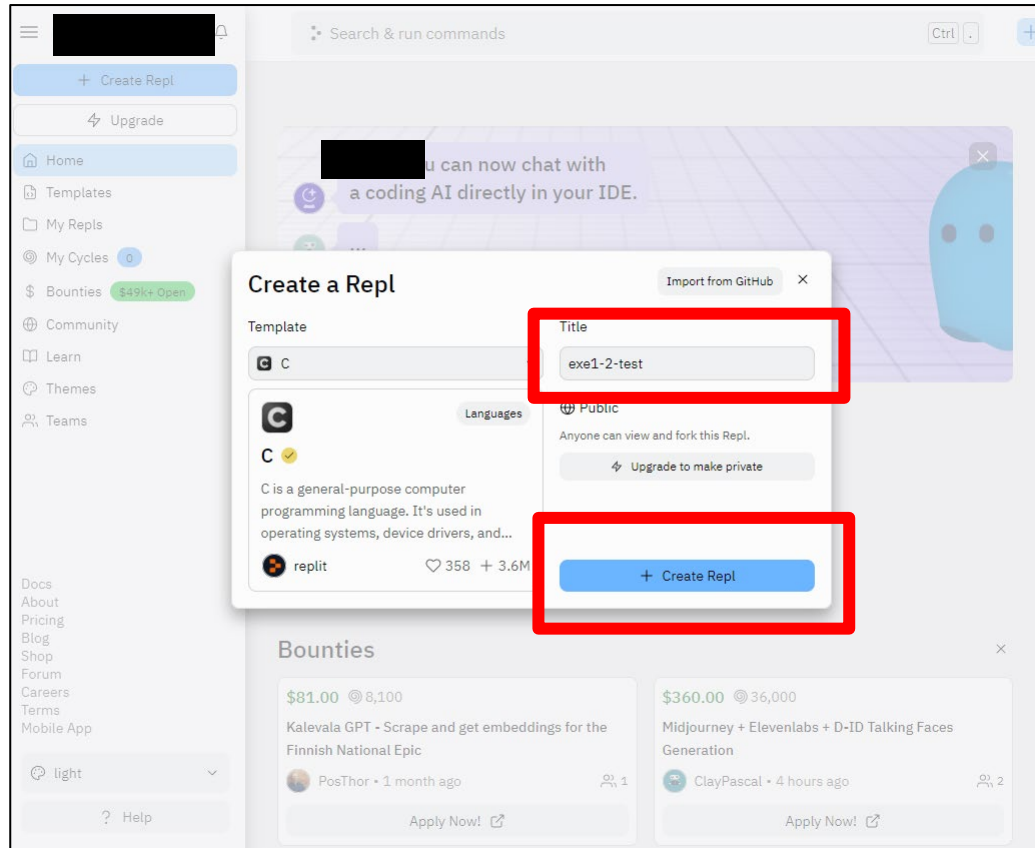


Homeの画面で,
「Create Repl」をクリック。

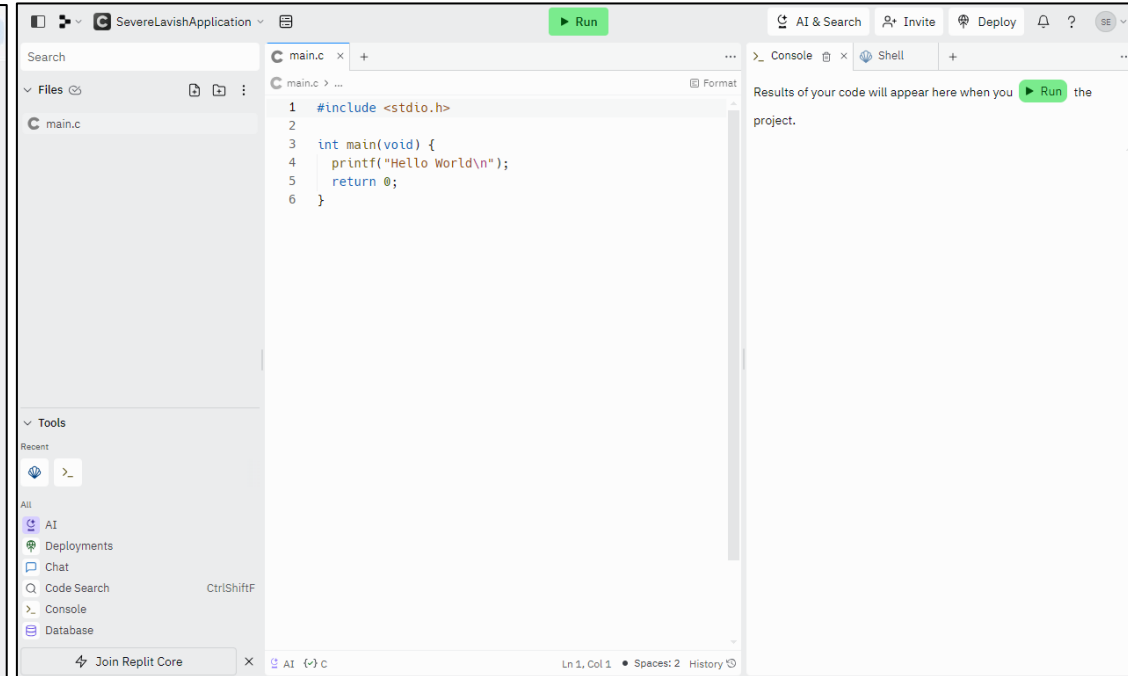


Template で「C」を選択する。

Linuxコマンドを試してみる@Replit

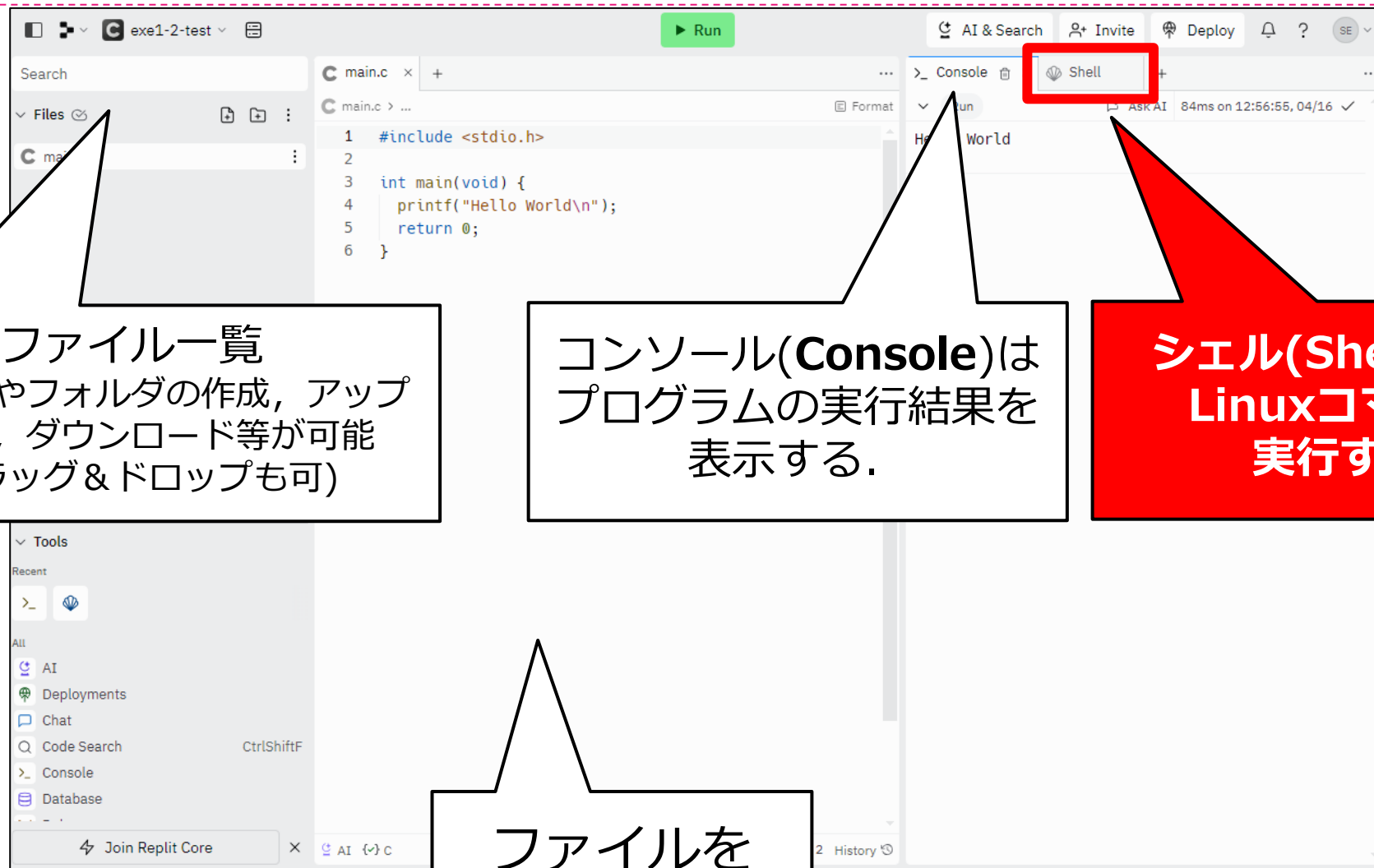


Titleを適当に設定する。
今回は「exe1-2-test」としてみた。
「Create Repl」をクリックする。



開発画面に移る。
デフォルトでC言語の Hello World の
コード(main.c)が用意されている。

Linuxコマンドを試してみる@Replit



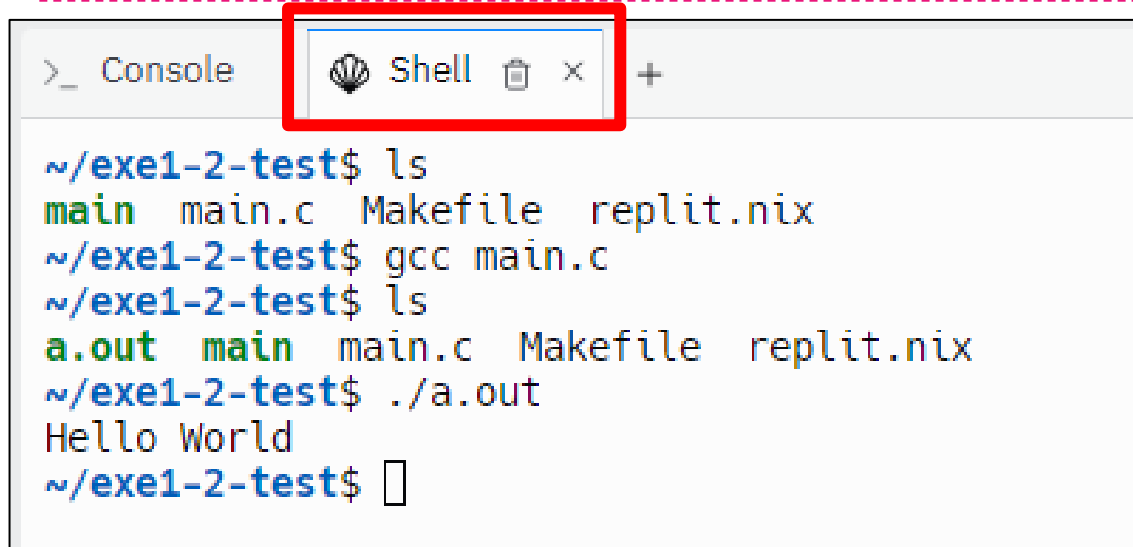
ファイル一覧
ファイルやフォルダの作成, アップ
ロード, ダウンロード等が可能
(ドラッグ&ドロップも可)

コンソール(**Console**)は
プログラムの実行結果を
表示する.

シェル(**Shell**)タブで
Linuxコマンドを
実行する.

ファイルを
編集する画面

Linuxコマンドを試してみる@Replit



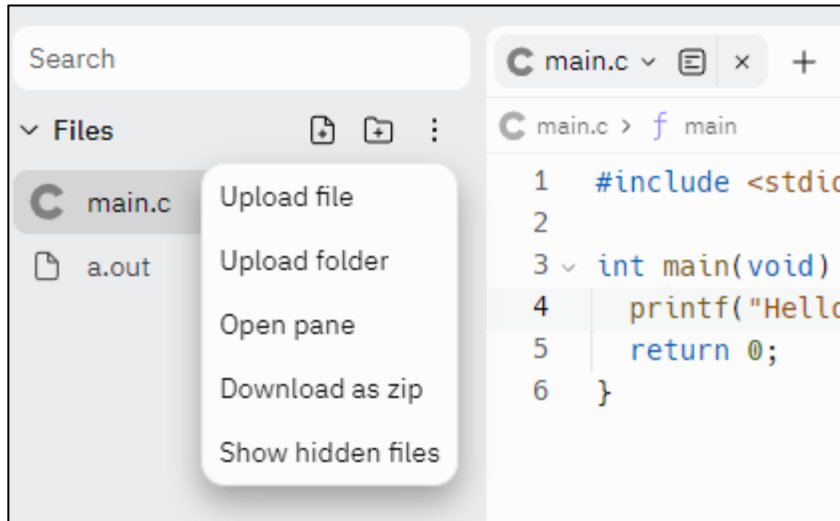
The screenshot shows a Replit terminal window with a tab labeled 'Shell' highlighted by a red rectangle. The terminal output is as follows:

```
>_ Console
~/exe1-2-test$ ls
main main.c Makefile replit.nix
~/exe1-2-test$ gcc main.c
~/exe1-2-test$ ls
a.out main main.c Makefile replit.nix
~/exe1-2-test$ ./a.out
Hello World
~/exe1-2-test$
```

Shellでのコマンドの実行例

1. `ls` で現在の作業ディレクトリ(フォルダ)の内容を表示
`main.c` が存在することを確認.
2. `gcc` で `main.c` をコンパイル
3. `ls` でディレクトリの内容確認
実行ファイル `a.out` が生成されていることを確認.
4. `./a.out` で実行ファイルを実行
`Hello World` が正常に動作することを確認.

Replit の注意点



- ▶ 左のカラム「Files」で新しいファイル・フォルダを作成したり、ソースコード等をアップロード・ダウンロードできる。
- ▶ **注意**
- ▶ 無料版の Replit では開発したファイル等が **オンライン上で公開された状態になってしまう**。
- ▶ 氏名や学籍番号が公開されることは望ましくないため、**終了する際はファイルをダウンロード後に削除することを推奨**。

基本コマンド ディレクトリ操作

コマンドは見聞きするだけでは
覚えにくいので手を動かして習得しよう

コマンド名	説明
pwd	p rint w orking d irectory 現在作業しているディレクトリを表示する
ls	l ist ファイルやディレクトリの情報を表示する
mkdir	m ake d irectory ディレクトリを作成する
rmdir	r emove d irectory ディレクトリを削除する
cd	c hange d irectory 作業ディレクトリを移動する



ディレクトリ操作の例

実際に自分で色々入力して試してみよう

```
> pwd
/home/runner/exe1-2-test-1
> ls
a.out main main.c Makefile replit.nix
> mkdir test
> ls
a.out main main.c Makefile replit.nix test
> cd test
> pwd
/home/runner/exe1-2-test-1/test
> cd ..
> pwd
/home/runner/exe1-2-test-1
> ls
a.out main main.c Makefile replit.nix test
> rmdir test
> ls
a.out main main.c Makefile replit.nix
> 
```

pwd で現在の作業ディレクトリの場所(パス)を表示

ls でファイル・ディレクトリを表示

mkdir test で test というディレクトリを作成

ls で test ディレクトリができていることを確認

cd test で test 内に移動する

pwd で test 内に移動したことを確認

cd .. で一つ上のディレクトリに移動できる

pwd で元のディレクトリに戻ったことを確認

ls で元のディレクトリの内容が表示されていることを確認

rmdir test で test を削除

ls で test が削除されていることを確認

ls の主なオプション

オプションによって、機能を変化させることができる。

オプション	説明	使用例
-F	ディレクトリや実行可能ファイルを記号付きで表示する。 ディレクトリ名の後に「/」、 実行可能なファイル名の後に「*」を付加する。	ls -F
-l	ファイルの詳細情報を表示する。 longなフォーマットなのでエル。 ファイルタイプ、パーミッション、ハードリンクの数、 オーナー名、グループ名、バイトサイズ、タイムスタンプ、 ファイル名が表示される。	ls -l
-a	隠しファイル（. で始まるファイル）も表示する。	ls -a
-R	サブディレクトリも再帰的に表示する。	ls -R

複数のオプションを, ls -la のように同時に指定できる



オプションの使用例

実際に自分で色々入力して試してみよう

```
> ls
a.out main main.c Makefile replit.nix test
> ls -F
a.out* main.c replit.nix
main* Makefile test/
> ls -l
total 84
-rwxr-xr-x 1 runner runner 15696 Apr 19 03:08 a.out
-rwxr-xr-x 1 runner runner 54536 Sep  8 2022 main
-rw-r--r-- 1 runner runner   77 Mar 31 16:28 main.c
-rw-r--r-- 1 runner runner  411 Oct 13 2022 Makefile
-rw-r--r-- 1 runner runner   81 Jan 18 2022 replit.nix
drwxr-xr-x 1 runner runner   0 Apr 19 03:08 test
> ls -a
.      .breakpoints main      .replit
..     .cache      main.c    replit.nix
a.out  .ccls-cache  Makefile test
> ls -R
.:
a.out main main.c Makefile replit.nix test

./test:
> ls -la
total 92
drwxr-xr-x 1 runner runner 146 Apr 19 03:08 .
drwxrwxrwx 1 runner runner  78 Apr 19 02:49 ..
-rwxr-xr-x 1 runner runner 15696 Apr 19 03:08 a.out
-rw-r--r-- 1 runner runner  17 Jun  1 2022 .breakpoints
drwxr-xr-x 1 runner runner  12 Oct 12 2021 .cache
drwxr-x--- 1 runner runner 164 Apr 19 02:49 .ccls-cache
-rwxr-xr-x 1 runner runner 54536 Sep  8 2022 main
-rw-r--r-- 1 runner runner  77 Mar 31 16:28 main.c
-rw-r--r-- 1 runner runner  411 Oct 13 2022 Makefile
-rw-r--r-- 1 runner runner 1437 Feb 16 14:41 .replit
-rw-r--r-- 1 runner runner  81 Jan 18 2022 replit.nix
drwxr-xr-x 1 runner runner   0 Apr 19 03:08 test
> □
```

ファイル a.out, コード main.c,
ディレクトリ test がある状態とする.

-F オプションで, 実行ファイルに *, ディ
レクトリに / が付加される.

-l オプションで詳細情報を表示.
パーミッション等の情報の意味は
各自で調べてみよう.

-a オプションで . から始まる
隠しファイルを表示.
「.」は作業ディレクトリ,
「..」は一つ上のディレクトリを表す.

-R オプションでディレクトリの再帰的表示.
今回は多階層でないため再起的かは確認で
きないが各フォルダの構成が見られる.

-l と -a オプションの組み合わせの例

基本コマンド

ファイル操作

コマンド名	説明
cat	catenate/conc atenate ファイルの内容を表示する
cp	copy ファイルやディレクトリをコピーする
mv	move ファイルやディレクトリを移動する ファイル名やディレクトリ名を変更する
rm	remove ファイルやディレクトリを削除する
touch	touch ファイルのタイムスタンプを更新する 空のファイルを作成する

cp コマンド

- ▶ 最後の引数がファイル名（あるいは存在しないパス）のときとディレクトリ名のときで挙動が異なる
- ▶ `cp file1 file2`
file1 を file2 という名前でコピーする
- ▶ `cp file1 dir1`
file1 を dir1 の中にコピーする
- ▶ `cp file1 file2 file3 dir1`
file1 と file2 と file3 を dir1 の中にコピーする
- ▶ `cp -r dir1 dir2`
dir1 の内容を dir2 という名前で再帰的にコピーする



mv コマンド

- ▶ cpと比べて, **元のファイルがなくなる**点が異なる
- ▶ コピー&ペーストとカット&ペーストの違いに似ている
- ▶ `mv file1 file2`
file1 を file2 にリネームする
- ▶ `mv file1 dir1`
file1 を dir1 の中に移動する



rm コマンド

- ▶ 通常、ディレクトリは削除できない
- ▶ -r オプションをつけるとサブディレクトリも含めて再帰的に削除する
→ ディレクトリも削除できる
- ▶ 上位のディレクトリで削除を実行すると下位のディレクトリが全部消えてしまうので、使用には注意が必要

重要なファイルを削除しないように注意！

[参考] 危険！決して実行してはいけないLinuxコマンド7個(+1個)

<https://orebibou.com/2014/08/%E5%8D%B1%E9%99%BA%EF%BC%81%E6%B1%BA%E3%81%97%E3%81%A6%E5%AE%9F%E8%A1%8C%E3%81%97%E3%81%A6%E3%81%AF%E3%81%84%E3%81%91%E3%81%AA%E3%81%84linux%E3%82%B3%E3%83%9E%E3%83%B3%E3%83%89/>

touch コマンド

- ▶ 存在するファイルを引数にとった場合,
タイムスタンプを現在時刻に更新する.
- ▶ 存在しないファイル名を引数にとった場合,
その名前の空のファイルを作成する.
⇒ ファイルを新規作成したいときに使える.



隠しファイル

- ▶ ドット"."で始まるファイルを隠しファイルという

```
~/exe1-2-test$ ls
a.out main main.c Makefile replit.nix
~/exe1-2-test$ touch .kakushi.txt
~/exe1-2-test$ ls
a.out main main.c Makefile replit.nix
~/exe1-2-test$ ls -a
. a.out .cache .gitignore .kakushi.txt main.c .replit
. .breakpoints .ccls-cache .kakushi main Makefile replit.nix
~/exe1-2-test$
```

- ▶ touch .kakushi.txt で隠しファイルを作成してみた
- ▶ ls ではファイルリストに表示されない
- ▶ ls -a だと隠しファイルが表示される
- ▶ システムファイルなど書き換えられたくないファイルに用いられる

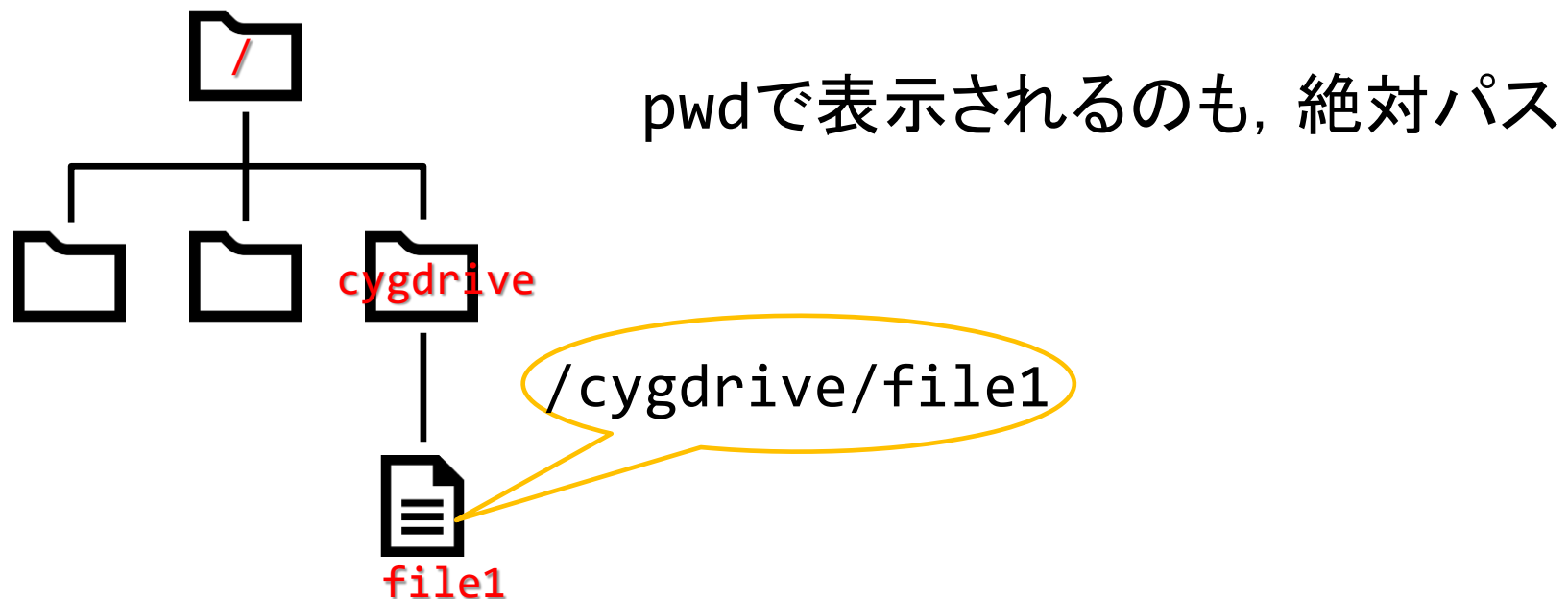
絶対パス

▶ パス

- ▶ ファイルやディレクトリの場所を示す文字列

▶ 絶対パス

- ▶ “/” (ルートディレクトリ) からディレクトリを辿っていった、ファイルやディレクトリの**絶対的な位置**を表現するパス

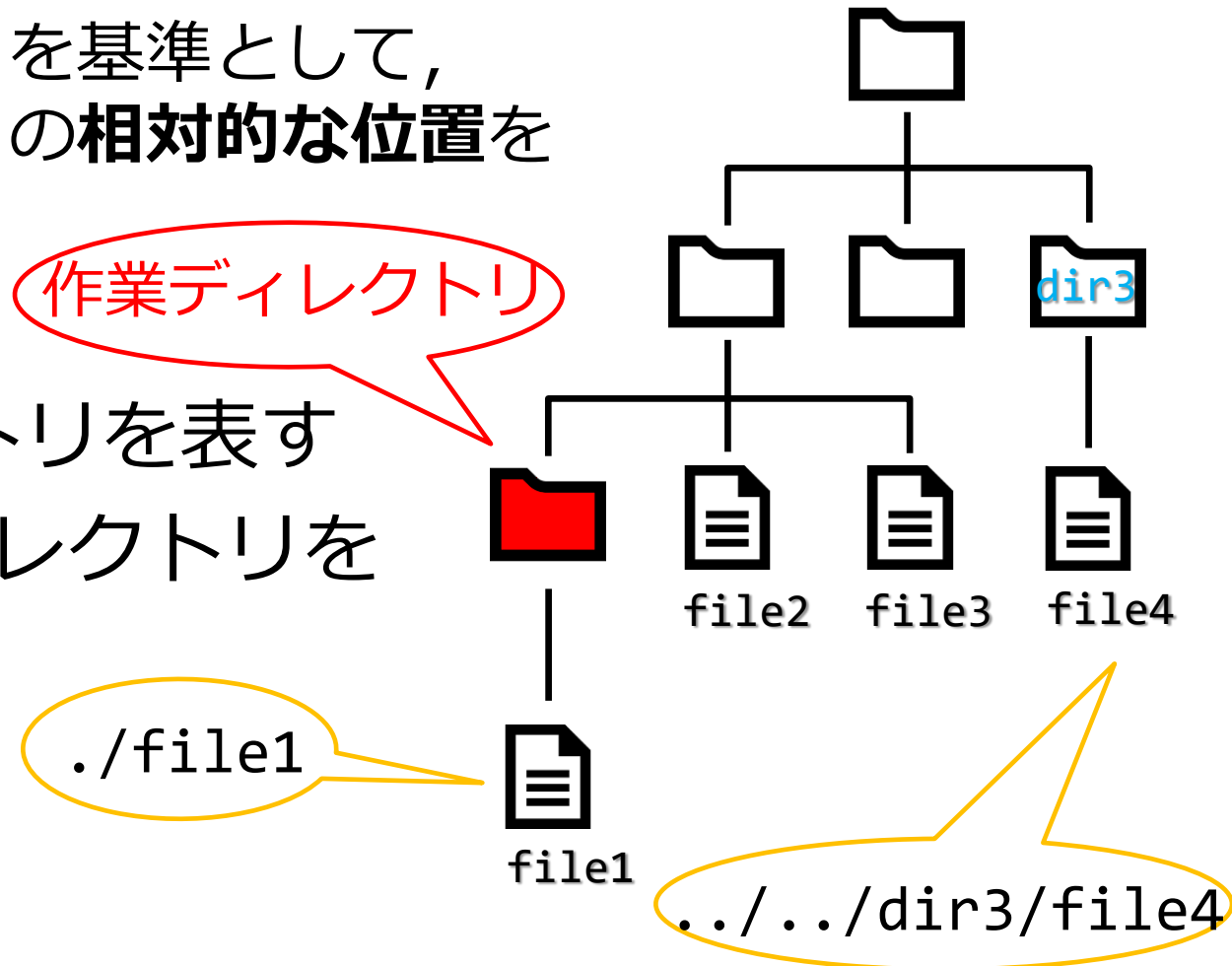


相対パス

▶ 相対パス

- ▶ 現在の作業ディレクトリを基準として、ファイルやディレクトリの**相対的な位置**を表現するパス

- ▶ 「`.`」は作業ディレクトリを表す
- ▶ 「`..`」は一つ上のディレクトリを表す



実行ファイルを `./a.out` と指定していたのは相対パス

ワイルドカード

複数のファイル・ディレクトリをパスに含めることができる。
条件に当てはまるファイル・ディレクトリを一括で処理する際に便利。

記号	意味
*	0文字以上の文字列 ただし、先頭の "." (ドット) は除く
?	任意の1文字 ただし、先頭の "." (ドット) は除く
[]	[]内の任意の1文字 " - " (ハイフン) を使って文字範囲を指定できる
{ }	{ }内の任意の文字列 文字列は " , " (カンマ) を使って複数列挙できる



ワイルドカードの使用例

▶ `rm ???`

作業ディレクトリ内で、ファイル名が 3 文字のファイルをすべて削除する（.で始まる隠しファイルは削除しない）
ディレクトリも該当するが `-r` オプションがない場合は削除されない。

▶ `ls *.txt`

作業ディレクトリ内で、ファイル名・ディレクトリ名が「.txt」で終わるものを全て表示する。

▶ `ls {[0-9]*,*[0-9]}`

作業ディレクトリ内で、ファイル名・ディレクトリ名が数字で始まるか、数字で終わるものを全て表示する。



第2回の課題 その1

▶ 指定ファイル名：**report-2-1-学籍番号.txt**

▶ 以下の各処理を実行するコマンドを答えよ

1. 作業ディレクトリの中にある、拡張子が「.txt」の全てのファイル（**隠しファイルも含む**）の詳細情報を表示する.

※「.test.txt」のようにドットから始まるファイルが隠しファイルです.

2. 作業ディレクトリの2つ上位のディレクトリの中にある、拡張子が「.c」または「.cpp」のファイルを全て、作業ディレクトリの中にある src という名前のディレクトリにコピーする.

**「1.」「2.」はそれぞれ
1度のコマンドで実行できるものとする.**

第2回の課題 その2

- ▶ 指定ファイル名：**report-2-2-学籍番号.c**
 - ▶ **[1, 100]** の一様乱数を**100個表示**するプログラム.
 - ▶ 数字は, **1行に1つ**, **整数**で出力されるものとする.
 - ▶ このコードは次回も使います.



提出に関して

- ▶ 全提出ファイルを1つのフォルダに格納し,
そのフォルダをzip圧縮したファイルを提出せよ.
- ▶ 指定フォルダ名 : **report-2-学籍番号**
 - ▶ 「report-2-0123456789」の形式で, 学籍番号を各自のものに置き換えること.
- ▶ 指定zipファイル名 : **report-2-学籍番号.zip**
↑このファイルを提出する.
- ▶ zipの中のフォルダ内のファイル構成 :
 - ▶ **report-2-1-学籍番号.txt**
 - ▶ **report-2-2-学籍番号.c**
- ▶ 提出期限は, 授業支援システム参照
- ▶ 提出方法 : 授業支援システムから提出
- ▶ 提出ファイルの中に, 学籍番号と氏名を明記のこと
 - ▶ プログラムではコメントの中に書くこと
- ▶ 注意点
 - ▶ 課題要件を満たさない場合や, 各種指示を守っていない場合(学籍番号と氏名が書かれていない, ファイル名が間違っている, 指定されていないファイル形式で提出している等)は減点する.
 - ▶ **今回指定した環境**でコンパイルできないプログラムは採点しない(0点)



zipファイル, 内部フォルダ,
提出ファイル2種の構成