```
# In order to accelerate the ops of creating files, we use the async file creation method.
import os
import asyncio
from aiofiles import open as aio_open
from typing import List


async def async_create_file(file_path: str, content: str) -> None:
    """
    Asynchronously creates a file at the specified path and writes the given content to it.

    Args:
        file_path (str): The path where the file will be created.
        content (str): The content to be written to the file.

    Returns:
        None
    """
    async with aio_open(file_path, "w") as file:
        await file.write(content)


async def create_multiple_files(
    file_paths: List[str], contents: List[str]
) -> None:
    """
```

Asynchronously creates multiple files at the specified paths and writes the corresponding content to each file.

Args:
file_paths (List[str]): A list of paths where the files will be created.
contents (List[str]): A list of content to be written to each file, corresponding to the file paths.

Returns:
None
"""
tasks = [
async_create_file(file_path, content)
for file_path, content in zip(file_paths, contents)
]
await asyncio.gather(*tasks)

async def create_file_with_directory(
file_path: str, content: str
) -> None:
"""
Creates a file with the specified directory path and content. If the directory does not exist, it is created.

Args:
file_path (str): The path of the file to be created, including the directory.

```
        content (str): The content to be written to the file.

    Returns:
        None
    """
    directory = os.path.dirname(file_path)
    if not os.path.exists(directory):
        os.makedirs(directory)

    await async_create_file(file_path, content)


def sync_create_file(file_path: str, content: str) -> None:
    """
    Synchronously creates a file at the specified path and writes the given content to it.

    Args:
        file_path (str): The path where the file will be created.
        content (str): The content to be written to the file.

    Returns:
        None
    """
    asyncio.run(async_create_file(file_path, content))
```

```
def sync_create_multiple_files(
    file_paths: List[str], contents: List[str]
) -> None:
    """
     Synchronously creates multiple files at the specified paths and writes the corresponding content to each file.

    Args:
        file_paths (List[str]): A list of paths where the files will be created.
        contents (List[str]): A list of content to be written to each file, corresponding to the file paths.

    Returns:
        None
    """
    asyncio.run(create_multiple_files(file_paths, contents))



def sync_create_file_with_directory(
    file_path: str, content: str
) -> None:
    """
     Synchronously creates a file with the specified directory path and content. If the directory does not exist, it is created.

    Args:
        file_path (str): The path of the file to be created, including the directory.
```

content (str): The content to be written to the file.

Returns:

None

"""

asyncio.run(create_file_with_directory(file_path, content))