3.2.1 Define Callback Function

The callback function is used to receive real-time output results from the RKLLM model. It is bound during the initialization of RKLLM and continuously outputs results to the callback function during the RKLLM model inference process, returning only one token each time.

Here is an example that callback function prints the output results in real-time to the terminal:

```
void callback(RKLLMResult* result, void* userdata, LLMCallState state)
{
    if(state == LLM_RUN_NORMAL){
        printf("%s", result->text);
        for (int i=0; i<reuslt->num; i++) {
            printf("token_id: %d logprob: %f", result->tokens[i].id,
            result->tokens[i].logprob);
        }
    }
    if (state == LLM_RUN_FINISH) {
        printf("finish\n");
    } else if (state == LLM_RUN_ERROR){
        printf("\run error\n");
    }
}
```

1) LLMCallState is a status flag, and its specific definition is as follows:

Table 3-8 Explanation of LLMCallState Status Flags

Definition	LLMCallState
Introduction	Used to indicate the current running state of RKLLM.
Enumeration Values	***LLM_RUN_NORMAL:*** indicates that the RKLLM model is currently inferencing; ***LLM_RUN_FINISH:*** indicates that the RKLLM model has completed inference; ***LLM_RUN_WAITING:*** indicates that the currently decoded character from RKLLM is not a complete UTF-8 encoding and needs to be concatenated with the next decoding result; ***LLM_RUN_ERROR:*** indicates that an error has occurred during inference;

During the design process of the callback function, users can set different post-processing behaviors based on the different states of LLMCallState.

2) RKLLMResult is a return value structure, and its specific definition is as follows:

Table 3-9 Explanation of RKLLMResult Structure