The transfer point is in arch/arm/lib/bootm.c

```
/* Subcommand: GO */
static void boot_jump_linux(bootm_headers_t *images, int flag)
{
#ifdef CONFIG_ARM64
        void (*kernel_entry)(void *fdt_addr);
        int fake = (flag & BOOTM_STATE_OS_FAKE_GO);

        kernel_entry = (void (*)(void *fdt_addr))images->ep;

        debug("## Transferring control to Linux (at address %lx)...\n",
                (ulong) kernel_entry);
        bootstage_mark(BOOTSTAGE_ID_RUN_OS);

        announce_and_cleanup(fake);

        if (!fake)
                kernel_entry(images->ft_addr);
#else
        unsigned long machid = gd->bd->bi_arch_number;
        char *s;
        void (*kernel_entry)(int zero, int arch, uint params);
        unsigned long r2;
        int fake = (flag & BOOTM_STATE_OS_FAKE_GO);

        kernel_entry = (void (*)(int, int, uint))images->ep;

        s = getenv("machid");
        if (s) {
                strict_strtoul(s, 16, &machid);
                printf("Using machid 0x%lx from environment\n", machid);
        }

        debug("## Transferring control to Linux (at address %08lx)" \
                "...\n", (ulong) kernel_entry);
        bootstage_mark(BOOTSTAGE_ID_RUN_OS);
        announce_and_cleanup(fake);

        if (IMAGE_ENABLE_OF_LIBFDT && images->ft_len)
                r2 = (unsigned long)images->ft_addr;
        else
                r2 = gd->bd->bi_boot_params;

        if (!fake)
                kernel_entry(0, machid, r2);                    ①
#endif
}
```

①

kernel_entry(0, machid, r2);

The transfer point!

The following is call stack when running on the transfer point.

_main() in crt0.S

--> board_init_r() in board.c

--> main_loop() in main.c

--> process_boot_delay() in main.c

--> run_command_list() in main.c

--> parse_string_outer() in hush.c

--> parse_stream_outer() in hush.c

--> run_list() in hush.c

--> run_list_real() in hush.c

--> run_pipe_real() in hush.c

--> cmd_process() in command.c

--> cmd_call() in command.c

--> do_bootm() in cmd_bootm.c

--> do_bootm_states() in cmd_bootm.c

--> boot_selected_os() in cmd_bootm.c

--> do_bootm_linux() in bootm.c

--> boot_jump_linux() in bootm.c