

Task: Startup

1. Start recording the screen of the VM using the menu option View -> Video Capture.
2. Log into the account you've been assigned with the same password as the username.

Task INCLINE: Warmup

Integration goal:

```
#include "Marlin.h"
#define VERSION_STRING "1.0.0"

float current_position[NUM_AXIS] = { 0.0, 0.0, 0.0, 0.0 };
#if defined(BARICUDA)
int ValvePressure=0;
int EtoPPressure=0;
#endif

static float destination[NUM_AXIS] = { 0.0, 0.0, 0.0, 0.0 };
#if defined(DELTA)
static float delta[3] = {0.0, 0.0, 0.0};
#endif

void process_commands() {
#if !defined(DELTA)
plan_set_position(current_position[X_AXIS], current_position[Y_AXIS],
current_position[Z_AXIS], current_position[E_AXIS]);
#else
calculate_delta(current_position);
plan_set_position(delta[X_AXIS], delta[Y_AXIS], delta[Z_AXIS], current_position[E_AXIS]);
#endif
}

#if defined(DELTA)
void calculate_delta(float cartesian[3]) {
    delta[X_AXIS] = sqrt(sq(DELTA_DIAGONAL_ROD)
        - sq(DELTA_TOWER1_X-cartesian[X_AXIS])
        - sq(DELTA_TOWER1_Y-cartesian[Y_AXIS])
    ) + cartesian[Z_AXIS];
    delta[Y_AXIS] = sqrt(sq(DELTA_DIAGONAL_ROD)
        - sq(DELTA_TOWER2_X-cartesian[X_AXIS])
        - sq(DELTA_TOWER2_Y-cartesian[Y_AXIS])
    ) + cartesian[Z_AXIS];
    delta[Z_AXIS] = sqrt(sq(DELTA_DIAGONAL_ROD)
        - sq(DELTA_TOWER3_X-cartesian[X_AXIS])
        - sq(DELTA_TOWER3_Y-cartesian[Y_AXIS])
    ) + cartesian[Z_AXIS];
}
#endif
```

Task: INCLINE

You will now perform an integration in the INCLINE tool. Your task is to transform the provided code into the integration goal, provided below, using intentions.

1. Launch MPS from the desktop.
2. From the project structure on the left, find the Examples solution, open the `integrated-*.c` file.
3. Arrange the views (**Ctrl+Alt+Shift+V**) and proceed with applying intentions to reach the integration goal (next page).
4. When you are satisfied, save the file in `~/results/incline.c` using the menu "Tools -> Export C++ File".

Task INCLINE: Vim

Integration goal:

```
static char *(p_bg_values[]) = {"light", "dark", NULL};
static char *(p_nf_values[]) = {"bin", "octal", "hex", "alpha", NULL};
#if defined(FEAT_CMDL_COMPL)
static char *(p_clcot_values[]) = {"menu", "menuone", "longest", "noinsert", "noselect",
NULL};
#endif /* defined(FEAT_CMDL_COMPL) */
static char *(p_ff_values[]) = {FF_UNIX, FF_DOS, FF_MAC, NULL};

options[] =
#if defined(OS2)
(char_u *)"/c",
#else
(char_u *)"-c",
#endif /* defined(OS2) */

#if defined(FEAT_CMDL_COMPL)
/* Don't allow recursive cmdline mode when busy with completion. */
if (clpum_compl_started || clpum_compl_busy || clpum_visible())
{
    EMSG(_(e_secure));
    return NULL;
}
clpum_compl_clear(); /* clear stuff for clpum */
#endif /* defined(FEAT_CMDL_COMPL) */

switch (c) {
case K_UP:
#if defined(FEAT_CMDL_COMPL)
if (clpum_visible())
showmode();
#endif /* defined(FEAT_CMDL_COMPL) */
#if defined(FEAT_CMDHIST)
i = hiscnt;
#endif /* defined(FEAT_CMDHIST) */
beep_flush();
}
```

Task: Eclipse

You will now perform an integration task in Eclipse CDT. Your task is to merge the two variant files you are provided with into the integration goal, provided below. We will use RCPTT to record editing actions.

1. Launch RCPTT from the desktop.
2. The file called recording should already be open.
3. On the bottom panel labeled Application, right click the item "org.eclipse..." and press Run.
4. In the RCPTT instance, press the "Record" button. The RCPTT instance will minimize.
5. Open the workspace folder task, select the mainline and fork files in it.
6. Right click and select "Compare With -> Each other".
7. A diff view is opened. The files can be swapped from left to right if they are on the opposite side of what you would prefer.
8. The diff view is editable, so you may make any manual edits you like, as well as using the tools available in the diff view. Proceed with applying edits until you reach the integration goal (next page).
9. When you are satisfied, save the resulting file in ~/results/eclipse.c. (It is not enough to just leave it in the workspace.)
10. Stop recording your actions in the RCPTT window by pressing the red button.

Task Eclipse: BusyBox

Integration goal:

```
#if defined(ANDROID) || defined(__ANDROID__)
#define endgrent() ((void)0)
struct timex;
pid_t getsid(pid_t pid);
int stime(const time_t *t);
int sethostname(const char *name, size_t len);
#ifdef READAHEAD
int adjtimex(struct timex *buf);
int pivot_root(const char *new_root, const char *put_old);
ssize_t readahead(int fd, off64_t offset, size_t count);
#endif
#endif

#ifdef ENABLE_SELINUX
#include <selinux/selinux.h>
#include <selinux/context.h>
#ifdef FLASK
#include <selinux/flask.h>
#include <selinux/av_permissions.h>
#endif
#endif

extern loff_t bb_copyfd_eof(int fd1, int fd2) FAST_FUNC;
extern loff_t bb_copyfd_size(int fd1, int fd2, loff_t size) FAST_FUNC;
extern void bb_copyfd_exact_size(int fd1, int fd2, loff_t size) FAST_FUNC;
extern void complain_copyfd_and_die(loff_t sz) NORETURN FAST_FUNC;

#define OFF_T_MAX ( ( off_t ) ~ ( ( off_t ) 1 << ( sizeof ( off_t ) * 8 - 1 ) ) )

struct BUG_off_t_size_is_misdetected {
char BUG_off_t_size_is_misdetected[sizeof(off_t) == sizeof(uoff_t) ? 1 : -1];
};

#ifdef BIN
#define LIBBB_DEFAULT_LOGIN_SHELL "-/bin/sh"
#else
#define LIBBB_DEFAULT_LOGIN_SHELL "-/sbin/sh"
#endif

#ifdef _LARGEFILE64_SOURCE
/* For lseek64 */
#define _LARGEFILE64_SOURCE
#endif

#ifdef BIN
putenv((char *) "SHELL=/bin/sh");
#else
putenv((char *) "SHELL=/sbin/sh");
#endif
```
