ASE: Séance 1

François Lepan
Benjamin Van Ryseghem

25 septembre 2012

1 Retour à un contexte

`#include <setjmp.h>`
`int setjmp(jmp_buf ctx)` save the current context into the `ctx` variable.
first time, returns 0 ;
second time returns something different from 0.
`void longjmp(jmp_buf ctx, int val)` where val is different from 0. Resume the cx context.

⚠ Beware of not keeping a context to a dead memory space (same principle that referring to a freed pointer)

2 Exercice 2

Ce programme est erroné car le `longjmp` change le contexte courant pour un contexte appartenant la partie de la pile qui a déjà été libérée.

3 Exercice 3

```
jmp_buf buffer;
static int mul(int depth) {
        int i;
        switch (scanf("%d", &i)) {
                case EOF :
                        return 1; /* neutral element */
                case 0:
                        return mul(depth+1); /* erroneous read */
                case 1 :
                        if (i)
                                return i * mul(depth+1);
                        else
                                return longjmp(buffer, ~0);
        }
}

int main() {
        int product;
        printf("A list of int, please\n");
        if (!setjmp(buffer)){
                product = mul(0);
        }
        else { product = 0; } */
        printf("product = %d\n", product);
}
```

François Lepan
Benjamin Van Ryseghem