

SOI - Laboratorium 1.

Zadaniem do wykonania było stworzenie nowego wywołania systemowego, polegającego na dodaniu zadanej liczby całkowitej do identyfikatora procesu wołającego i wyświetlenie rezultatu w terminalu.

Kolejne etapy pracy podczas laboratorium:

1. W pliku `usr/include/minix/callnr.h` utworzony został identyfikator nowego wywołania o nazwie `ADDTOINT` I numerze 79 oraz zwiększona została liczba `NCALLS`.

```
#define NCALLS      80  /* number of system calls allowed */
#define ADDTOINT    79
```

2. W pliku `/usr/src/mm/proto.h` utworzony został prototyp funkcji obsługującej:

```
_PROTOTYPE( int add_to_int, (void) );
```

3. W pliku `/usr/src/mm/main.c` zosrała napisana publiczna procedura obsługi o nazwie `add_to_int`. Procedura ta zwraca szukaną w zadaniu wartość dodając do identyfikatora procesu wołającego liczbę zadaną jako argument programu testowego.

```
PUBLIC int add_to_int()
{
    return mproc[mm_in.m_source].mp_pid + mm_in.m1_i1;
}
```

Identyfikator procesu wołającego otrzymujemy dzięki sprawdzeniu indeksu w tablicy `mproc` procesu wysyłającego wiadomość do procedury (`mm_in.m_source`) oraz odwołując się do odpowiedniej wartości w strukturze.

4. W pliku `/usr/src/mm/table.c` w tablicy `call_vec` w odpowiednim miejscu wstawiony został adres (nazwa) funkcji `add_to_int`, zaś w pliku `/usr/src/fs/table.c` w tym samym miejscu umieszczony został adres pusty funkcji, `no_sys`.

```
add_to_int    /* 79 = addtoint */
no_sys       /* 79 = ADDTOINT */
```

5. Po powyższych czynnościach dokonano rekompilacji I przeładowania systemu MINIX z nowym jądrem. Kompilację wykonano w katalogu `/usr/src/tools` wykorzystując polecenie `# make hdbboot`.
6. Po ponownym uruchomieniu systemu napisany został program testujący nowe wywołanie o nazwie `testowy.c`, którego kod źródłowy wysłany został jako załącznik. Dzięki dodaniu katalogu z programem do zmiennej `PATH` program można wywołać z dowolnego miejsca w systemie wpisując:

```
# testowy int
```

gdzie "int" jest liczbą całkowitą dodawaną do identyfikatora procesu podczas stworzonego wywołania systemowego `ADDTOINT`. Przykładowe wywołanie:

testowy 5

dało następujący wynik:

PID: 39,
Wynik wywołania: 44

faktyczny identyfikator procesu jest otrzymywany w programie testowym z funkcji getpid() I jest wyświetlany w celu sprawdzenia czy program działa poprawnie.