Semaphore

TEST SET

Key

Recap

CSE 430: Operating Systems

Class: 12  Date: 10/15
\[ f(z) = z \]  

\[
\begin{align*}
\text{pigeonhole principle} & \quad \forall n \in \mathbb{N}, \quad \exists j \in \{1, 2, \ldots, n\} \\
\text{such that} \quad & \quad a_j = z \quad \text{for some } z \in \mathbb{C} \\
\text{Thus,} \quad & \quad \text{if } \deg f = 0; \quad \text{then } \deg g = 0.
\end{align*}
\]
main()

\[ \exists \ f_1() \rightarrow \text{only } f_1 \text{ runs} \]

\[ f_2() \]

\[ \]

\[ \rightarrow \text{want both to run} \]
Fe 0 1
Fe 0 0
Fe 1 2
Fe 1 3
Fe 0 0
Fe 0 0
Fe 1 3
Fe 1 2
Fe 0 1

Top — not signed
is the instructor

freshman - concept supervene

does examined - tab

low level tools
C library

```c
#include "context.h"

typedef struct {
    /* context */
} context_t;

swap_context();
gcontext();
```
Each & every

context - t context?
context - t prev?
context - t next?

Run, a queue of tests
3. Proceed to the next step, then proceed.

Address (LMask, F, s)

Call interface (t, F, s)

Allocate a stack (s

Allocate a target (t

Stack frame (f)
def search_next_context(cur, next):
    if next:
        yield next
    else:
        next = cur
        while True:
            yield cur
            cur = search_next_context(cur, next)
3. Function defines int p

Some people

deserve vengeance (not necessarily of

decide to take it).
For now suppose busy logic input

\[ \text{while (flag = busy)} \]

\[
\begin{cases}
\text{exit and } x & : \text{if } 0 < s \leq -2 \\
\text{else and } a & \\
\end{cases}
\]

\[
P(\tau) \leftarrow x : \text{if } s > 0
\]
Init \( p \), \( v \)

- Increment

\( u(s) : (s) \)

Atomic

\( s++ \)
\( V(5) \)
\( V(5) \)
\( V(5) \)
\( V(5) \)

\( c \)

\( c \)

\( c \)

\( p(5) \)

\( p(5) \)

\( p(5) \)

\( p(5) \)

\( \overline{\text{Thus: } \frac{1}{2} \leq i \leq 3} \) \( \text{Semiprime } S; \text{ intersecting } T \) \( \text{and } \Phi_2 \text{ meet} \) \( T \) \( \text{which such are } C \) \( \text{numbers} \)
\[ \mathcal{L} = \mu(\mathcal{L}) \]